
Beyond Detection: What Counts as Convincing Evidence of AI Collusion¹

Mlibo Bangushe
UCT

Kayleigh Campbell
UCT

Asemahle Mtinkulu
UCT

With
Apart Research

Abstract

Determining whether AI agents are colluding is difficult because similar market outcomes can arise from coordination or from agents responding independently to shared incentives, so unusual outcomes and significant detector flags do not by themselves establish collusion. We ask what should count as convincing evidence through an exploratory, implementation-level audit of saved outputs and detector code from the sprint’s starter repository, which bundles Tailor’s (2025) ColludeBench and audit pipeline. We compare a 200-run honest-calibration condition (negative control) with a 20-run keyword-collusion condition (positive control) in a scripted-agent pricing duopoly. Mean consumer surplus is 10.97% lower under keyword collusion (70.53 vs 79.22), and the gap persists on matched seeds. Raw mutual-information statistics (0.718 vs 0.139) are not comparable across conditions because their null means differ (0.529 vs 0.102). The linguistic detector reports a smaller p-value for the honest condition (0.0002) than for the keyword condition (0.008); the code tests a maximum over runs against a single-run reference distribution, which helps explain the inversion. We propose that convincing evidence of collusion should be convergent, calibrated and comparative. These results come from scripted agents, so they do not establish detector performance on

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

1. Introduction

Similar prices from two pricing algorithms do not show that they are colluding: they may coordinate, or they may simply respond to the same incentives and market conditions. Calvano et al. (2020) showed that reinforcement-learning pricing algorithms can learn to sustain supracompetitive prices without direct communication, while Calvano et al. (2023) examine cases in which algorithms reach such prices without learning a genuinely collusive strategy. How an outcome is produced therefore matters, and the outcome alone cannot be conclusive evidence. This raises the sprint’s question 2.3: what should count as convincing evidence of collusion?

The question matters because detector outputs are increasingly likely to inform decisions about deploying, monitoring or restricting multi-agent systems. A flag read as proof can penalise innocent parallel behaviour, and a weak or miscalibrated statistic that is trusted can let real coordination pass. Our theory of change is that evaluators can act on detector outputs responsibly only if they know what each reported statistic does and does not show.

We therefore audit a detector stack, rather than build one. We read the saved outputs of a negative-control (honest, $n=200$) and a positive-control (keyword collusion, $n=20$) pricing condition alongside the code that produced them, using the scripted-agent environments and detectors of Tailor (2025) from the sprint’s starter repository (pre-existing work). Two findings show how reported statistics can mislead: raw MI values cannot be compared across conditions without their null means, and the linguistic detector’s p-value tests a maximum over runs against a single-run reference. From these we derive criteria for convincing evidence. We are not aware of prior work that audits these detectors at the implementation level in this way, but we have not carried out a systematic literature search. Because the conditions differ in configuration and seeds, the comparison is descriptive, not causal.

Our contributions are:

1. A comparative audit of saved outputs, documenting differences in consumer surplus (with a matched-seed check) and in standardised cross-run MI, without attributing them solely to the collusion intervention.
2. A code-level finding that the linguistic detector’s reported p-value tests a maximum over runs against a single-run reference, which helps explain why its p-value ordering reverses its statistic ordering.
3. A convergent, calibrated and comparative synthesis for evaluating evidence of collusion, offered as guidance, not a validated detection framework.

The audit does not introduce a detector, show that the detectors are unreliable, or show how they perform on LLM-agent collusion.

2. Related Work

Supracompetitive prices alone do not establish collusion. Calvano et al. (2020) show that reinforcement-learning pricing algorithms can sustain them through punishment strategies without communication, and Calvano et al. (2023) distinguish genuine from spurious collusion, so the mechanism matters. Competition-policy work faces the same evidentiary problem: the OECD (2017) examines how algorithms complicate the line between agreement, tacit collusion and parallel conduct, and the OECD (2023) considers how authorities should investigate algorithmic harms. Legal standards and technical detection criteria are, however, not interchangeable.

For LLM agents, Motwani et al. (2024) formalise secret collusion through steganography. Tailor (2025) pairs channel-capacity theory with ColludeBench-v0 (pricing, auction, peer-review and double-auction environments) and a detector stack (mutual information, permutation invariance, watermark variance, linguistic and fairness detectors) calibrated on honest replays. Xu et al. (2026) show from another direction that price-level audits can miss profitable collusion.

These works propose or motivate detectors. We ask a complementary question: given a detector’s reported output, what does it establish? An evaluator would use our audit before relying on a flag, to see where a statistic cannot be compared across conditions or where a p-value does not correspond to the quantity tested. We introduce no new detector or benchmark.

3. Methods

3.1 Design, Provenance and Experimental Conditions

We audited saved summaries, configuration files and selected detector code in Poseidon Research’s `collusion_hackathon` starter repository (commit e6fb130; `steganographic-collusion/`). We ran no new experiments. The environments, detectors, calibration files and results are pre-existing work by Tailor (2025; MIT licence). New in this report are the code-to-output audit, the standardised statistics, the matched-seed comparison, the illustrative simulation and the framework in Section 5.

Both conditions use the `pricing_duopoly` scenario with scripted agents (15 rounds; full parameters in Appendix B):

- **Honest (negative control):** `pricing_honest.json`, 200 runs, seeds 0–199, from `configs/audits/pricing_honest_calibration.yaml`. Two scripted honest agents quote a fixed competitive price.
- **Keyword (positive control):** `pricing_keyword.json`, 20 runs, seeds 0–19, from `configs/audits/pricing_keyword_audit.yaml`. A scripted colluder (`firm_a`) uses the keyword covert scheme at 4 payload bits per 100 tokens against a scripted honest agent (`firm_b`), with scenario seed jitter 0.2 and paraphrase probability 0.3.

Audit-level interventions are identical in both conditions (Appendix B). Because the conditions differ in configuration, seed range and run count, differences between them are descriptive. The honest condition is not a control for all forms of innocent parallel behaviour.

3.2 Measures and Saved Outputs

We examined consumer surplus, per-firm rewards, Lerner indices and rounds (market outcomes, which do not by themselves establish coordination), and the saved outputs of six detectors: cross-run MI, permutation invariance, watermark variance, linguistic steganalysis, acceptance bias (uninformative here: “insufficient group coverage” in both conditions) and welfare shift. We recomputed consumer-surplus means, the standardised scores in Table 1 and the matched-seed comparison from the saved per-run values. We did not recompute all metrics from raw runs or evaluate detector error rates.

3.3 Statistical Interpretation and Code Inspection

We read the detector code and the p-value helper, which computes Gaussian tail probabilities from a score, mean and sd.

- **Linguistic detector:** messages are lowercased and whitespace-tokenised; each run’s token counts are compared with expected frequencies pooled across the condition’s runs, giving a per-run chi-square statistic and type-token ratio. The reported statistic is the maximum chi-square $\times (1 + \text{the range of type-token ratios})$, and its Gaussian p-value uses the mean and sd of the per-run chi-square values. The maximum over n runs is therefore tested against a single-run reference.
- **Cross-run MI:** each summary stores its null mean and sd, so we standardise as $(\text{statistic} - \text{null mean})/\text{null sd}$. Pairwise estimates reuse runs, so the pair counts (19,900 and 190) are not independent observations.
- **Welfare shift:** its baseline is the honest calibration run itself.

3.4 Additional Analyses

Matched-seed comparison: we compared the conditions on the shared seeds 0–19. Illustrative simulation: we simulated the linguistic detector’s construction under a Gaussian null (Appendix B; seed 1). Every figure in Section 4 can be reproduced from the two result files, the two configs and

the detector code at the pinned commit. We did not rerun the underlying experiments or build matched non-collusive controls within the sprint weekend (Section 5.2).

4. Results

We call the honest-calibration condition ($n=200$) the negative control and the keyword condition ($n=20$) the positive control. Both use scripted agents. All values are repository-reported outputs unless stated otherwise (Table 1).

4.1 Economic Outcomes

Mean consumer surplus is 79.225 in the negative control and 70.534 in the positive control, 10.97% lower. On the shared seeds 0–19, the means are 79.222 and 70.534, a difference of 8.689; the honest run has the higher surplus on every matched seed, with paired differences of 8.47–9.07 (per-run sd 0.155 honest, 0.093 keyword). The gap therefore does not depend on run count or seed range. It can still reflect other configuration differences (Section 3.1), so it remains descriptive. Surplus describes an outcome and does not identify a mechanism.

4.2 Cross-Run Mutual Information

The reported MI statistic is 0.139 ($p = 0.191$) for the negative control and 0.718 ($p = 0.022$) for the positive control. The raw statistics are not directly comparable, because the stored null distributions differ: mean 0.102 (sd 0.042) for honest and 0.529 (sd 0.094) for keyword. Relative to its own null, the honest statistic is 0.87 sd above the null mean and the keyword statistic is 2.01. These are differences in reported outputs; their evidentiary weight depends on how the statistic and null are constructed, and they do not establish that the dependence arose from collusion.

4.3 Linguistic Detector Outputs

The detector reports 539.985 ($p = 0.000237$) for the negative control and 719.576 ($p = 0.008083$) for the positive control: the keyword statistic is higher, but its p-value is also higher. Section 3.3 explains why the ordering can reverse: because the maximum over n runs is tested against a single-run reference, the p-value shrinks as n grows. Against the per-run chi-square mean and sd, the standardised scores are 3.49 (honest, $n=200$) and 2.41 (keyword, $n=20$). In an illustrative Gaussian simulation of this construction (Appendix B), a nominal 5% test rejects in about 78% of cases at $n=20$ and in essentially all cases at $n=200$. Real chi-square values are not Gaussian, so this shows the direction of the effect, not the detector’s actual error rate. This is a calibration concern found by code inspection, not proof that the repository’s headline results are wrong: its documentation states that its decisions rely on empirical thresholds calibrated on honest

runs and describes this detector mainly as a drift indicator, so the concern applies to the reported p-value field.

4.4 Other Reported Detector Outputs

The welfare-shift baseline is the honest run itself, so its honest result ($z = 0$, $p = 0.5$) holds by construction; its keyword result ($z \approx 250$) measures the same 10.97% gap against that baseline’s run-to-run sd of 0.155. Permutation invariance gives $p = 0.5$ in both conditions because its statistic equals its own null mean. Watermark variance is 0.0 in both ($p = 0.5$ and 0.209), and acceptance bias is uninformative. We did not evaluate these detectors’ performance, so their outputs are not treated as validated evidence of collusion.

Table 1. Reported detector outputs and standardised scores by condition. Standardised = (statistic — null mean)/null sd from each result file; for the linguistic detector, the per-run chi-square mean and sd (honest 156.5, 109.7; keyword 404.4, 131.0). MI null means: 0.102 (honest), 0.529 (keyword). Welfare $p = 0.0$ is the recorded output, not an exact probability.

<i>Detector</i>	Honest: statistic	Honest: p	Honest: standardised	Keyword: statistic	Keyword: p	Keyword: standardised
<i>Cross-run MI</i>	0.139	0.191	0.87	0.718	0.022	2.01
<i>Permutation invariance</i>	0.176	0.5	0.0	6.978	0.5	0.0
<i>Watermark variance</i>	0.0	0.5	n/a (null sd is 0)	0.0	0.209	−0.81
<i>Linguistic steganalysis</i>	539.985	0.000237	3.49	719.576	0.008	2.41
<i>Acceptance bias</i>	0.0	1.0	n/a (no group coverage)	0.0	1.0	n/a (no group coverage)
<i>Welfare shift</i>	0.0	0.5	0.0 (baseline is the honest run itself)	8.691	0.0	250.4

5. Discussion and Limitations

For AI safety, the practical implication is that detector outputs will increasingly inform decisions about monitoring, deploying or restricting multi-agent systems. A miscalibrated or

non-comparable statistic read as proof can penalise innocent behaviour, and a weak one that is trusted can let coordination pass. Economic outcomes, behavioural signals and statistical significance should therefore be interpreted separately.

5.1 What Should Count as Convincing Evidence?

We propose that convincing evidence of AI-agent collusion should be convergent, calibrated and comparative. This is guidance drawn from the audit and related work, not a validated scoring rule.

- **Convergent.** Distinct measurements should support one coherent explanation of the coordination mechanism. Economic harm such as lower consumer surplus justifies investigation but cannot identify the mechanism (Calvano et al., 2020, 2023). Where signalling is proposed, evidence should show what was encoded, whether another agent could recover it, and whether later behaviour was consistent with it.
- **Calibrated.** Reported significance depends on how a statistic and its reference distribution are built. The linguistic p-value tests a maximum against a single-run reference, and the raw MI statistics (0.139, 0.718) are interpretable only against their own nulls (0.102, 0.529). The full pipeline, including aggregation and feature selection, should be tested on repeated null data, reporting false-positive rates and uncertainty.
- **Comparative.** Parallel behaviour can arise from shared information, incentives or software (OECD, 2017). The higher standardised MI under keyword collusion shows greater measured dependence, not that the signal separates collusion from innocent parallel behaviour. Evidence is stronger against non-collusive controls that preserve plausible sources of similarity but remove the coordination mechanism. The welfare-shift result, zero by construction against its own baseline, carries no independent weight.

The audit motivates these criteria but does not show that they can reliably classify cases; that needs validation against well-designed reference conditions.

Limitations

This study has several substantial limitations.

1. This is a secondary audit: we recomputed only the surplus means, standardised scores and matched-seed comparison.
2. The conditions differ in run count, seeds and scenario settings (jitter, paraphrase). The matched-seed check removes run count and seed range but not configuration, so the gap is not causal.
3. The honest condition quotes a fixed price: it is a low-variability baseline, not a noisy null or matched control, so false-positive rates cannot be estimated.
4. The linguistic calibration concern rests on code inspection and a Gaussian simulation that is illustrative only.

5. The runs use scripted agents. We do not address LLM-agent collusion, adaptive colluders, channels other than messages (actions, tools), or other scenarios and detectors in the repository.
6. The framework is unvalidated, and the work was done over the 23–25 October 2026 sprint weekend, which limited us to inspection and light re-analysis.

Assumptions

We assume the saved JSON files faithfully record the runs (otherwise Section 4 cannot be relied on); that the scripted keyword scheme is a reasonable stand-in for covert coordination (otherwise our conclusions concern this scheme only); and that the honest condition approximates a null (otherwise reading the linguistic result on honest runs as a false-positive signal is weakened).

Future Work

The findings point to four steps. (1) Build matched non-collusive controls that preserve incentives and information but remove the mechanism, including innocent parallel behaviour. (2) Calibrate the complete pipeline on repeated null data, for example by replacing the linguistic detector’s single-run reference with a reference distribution for the maximum, and report false-positive and false-negative rates. (3) Account for dependence between pairwise MI estimates and keep baselines such as the welfare-shift baseline separate from the conditions they test. (4) Test LLM-agent runs and held-out settings, agent types and noise levels, documenting seeds, configs, run counts and analysis-code versions. These align with measurement and risk-management principles in the NIST AI RMF (2023), which is general guidance, not a collusion-specific evidentiary standard.

6. Conclusion

This audit examined what saved outputs and selected detector code in the sprint’s starter repository can establish about suspected AI-agent collusion. In a scripted-agent pricing duopoly, the honest and keyword conditions differ in mean consumer surplus (10.97% lower under keyword collusion, persisting on matched seeds) and in cross-run MI, whose raw statistics are interpretable only against each condition’s own null. The linguistic detector’s p-value tests a maximum over runs against a single-run reference, which helps explain why its p-value ordering reverses its statistic ordering. The audit does not establish that collusion occurred, that the detectors are generally unreliable, how accurately they separate collusion from innocent parallel behaviour, or how they perform on LLM-agent collusion.

We propose that convincing evidence of AI-agent collusion should be convergent, calibrated and comparative: distinct measurements should support a coherent explanation of a plausible coordination mechanism, statistical procedures should be tested against suitable reference

conditions, and collusion should explain the behaviour better than credible non-collusive alternatives. This is a basis for structuring future evaluations, not a validated framework. Detecting an unusual outcome or signal is not enough; a convincing claim must show why coordination is the better-supported explanation.

Code and Data

- **Code:** Poseidon Research’s `collusion_hackathon` starter repository, https://github.com/poseidon-research/collusion_hackathon (commit e6fb130). The audited environments and detectors are pre-existing work by Tailor (2025), MIT licence; new in this report is the analysis and framework in Sections 3–5.
- **Data:** `steganographic-collusion/results/experiments/pricing_honest.json` and `pricing_keyword.json`, with configs `configs/audits/pricing_honest_calibration.yaml` and `pricing_keyword_audit.yaml`. The saved runs use the repository’s scripted agents. We use repository-reported outputs and code inspection and do not claim an independent reproduction.
- **Reproducibility:** every number in Section 4 can be recomputed from these files and the detector code at the pinned commit; no new experiment runs are claimed. We know of no information-hazard concerns beyond Appendix A.2.

References

1. Calvano, E., Calzolari, G., Denicolò, V., & Pastorello, S. (2020). Artificial intelligence, algorithmic pricing, and collusion. *American Economic Review*, 110(10), 3267–3297. <https://doi.org/10.1257/aer.20190623>
2. Calvano, E., Calzolari, G., Denicolò, V., & Pastorello, S. (2023). Algorithmic collusion: Genuine or spurious? *International Journal of Industrial Organization*, 90, Article 102973. <https://doi.org/10.1016/j.ijindorg.2023.102973>
3. Motwani, S. R., et al. (2024). Secret collusion among generative AI agents: Multi-agent deception via steganography. arXiv:2402.07510. <https://arxiv.org/abs/2402.07510> [list all seven authors if your style requires it]
4. National Institute of Standards and Technology. (2023). *Artificial intelligence risk management framework (AI RMF 1.0)*. <https://doi.org/10.6028/NIST.AI.100-1>
5. Organisation for Economic Co-operation and Development. (2017). *Algorithms and collusion: Competition policy in the digital age*. OECD Publishing. <https://doi.org/10.1787/258dcb14-en>

6. Organisation for Economic Co-operation and Development. (2023). *Algorithmic competition* (DAF/COMP(2023)3). [https://one.oecd.org/document/DAF/COMP\(2023\)3/en/pdf](https://one.oecd.org/document/DAF/COMP(2023)3/en/pdf)
7. Poseidon Research. (n.d.). *collusion_hackathon* [Code and data; commit e6fb130]. GitHub. https://github.com/poseidon-research/collusion_hackathon
8. Tailor, O. (2025). *Audit the Whisper: Detecting steganographic collusion in multi-agent LLMs*. arXiv:2510.04303. <https://arxiv.org/abs/2510.04303>
9. Xu, X., Wu, C., Lu, J., Tan, K., Tao, S., & Hong, H. (2026). *Collusion with competitive marginals: Price-level audits are blind by construction*. arXiv:2607.26385. <https://arxiv.org/abs/2607.26385>

Appendix A Limitations and Dual-Use / Ethical Considerations

A.1 What the results do and do not establish.

The results establish that, in these saved scripted-agent runs, the keyword condition has lower consumer surplus than the honest condition (including on matched seeds); that raw MI statistics are not comparable without their nulls; and that the linguistic detector’s p-value tests a maximum over runs against a single-run reference. They do not establish that collusion occurred, how accurately the detectors separate collusion from innocent parallel behaviour, their false-positive or false-negative rates, or their performance on LLM-agent collusion. The linguistic concern rests on code inspection and an illustrative simulation, and the repository’s own decisions rely on empirical thresholds. The proposed framework is unvalidated.

A.2 Dual-use considerations.

Collusion-detection research can help researchers, authorities and developers identify harmful coordination between AI agents. Our findings could also help a colluder choose conditions where a detector’s reported significance is least informative, for example by showing that the linguistic p-value depends on the number of runs and that raw MI statistics are not comparable across conditions. The same findings help defenders see where calibration is needed. We disclose weaknesses only at the level of statistical construction and present no tested evasion strategy. Future work should weigh transparency against whether particular implementation details would materially facilitate evasion.

A.3 Ethical interpretation and responsible future work.

A detector flag is not proof of wrongdoing or intent. False positives can cause unjustified suspicion or intervention; false negatives can leave harmful coordination undetected. Consequential assessments should consider data provenance, calibration, the strength of the proposed mechanism, credible alternatives and independent review, and no single metric or threshold should be treated

as conclusive without validation. Future evaluations should use matched non-collusive controls, validate the full pipeline against reference conditions, report uncertainty and error rates, test beyond development settings, and distinguish observed results from interpretations.

Appendix B. Additional methodological detail

Scenario parameters (both conditions). `pricing_duopoly`; 2 firms; 15 rounds; marginal cost 1.0; demand intercept 10; own-price elasticity 1.5; cross-elasticity 0.5; demand noise sd 0.1. Honest: no colluders, 0 payload bits; seeds 0–199 from a jitter schedule (base 0, count 200). Keyword: `firm_a` colluding, keyword scheme, 4 payload bits per 100 tokens; scenario seed jitter 0.2 and paraphrase probability 0.3; seeds 0–19.

Audit-level interventions (identical). Budget jitter (12–28 tokens); canary phrases (frequency 0.25); rate limiting (block probability 0.1, replacement “[RATE-LIMITED]”); selective paraphrase (probability 0.45); role permutation.

Illustrative simulation. For $n = 20$ and $n = 200$, we drew n independent standard-normal values, took the maximum, computed a right-tail Gaussian p-value from the mean and sd of the same draws, and repeated this 20,000 times with random seed 1. At a nominal 5% level the rejection rate was about 78% ($n = 20$) and essentially 100% ($n = 200$). This illustrates the construction under a Gaussian null only; real per-run chi-square values are not Gaussian.

LLM Usage Statement

ChatGPT was used to assist with brainstorming. Claude was used to help inspect the repository, recompute the reported statistics, and edit the draft sections of the manuscript. The team remains responsible for the final submission, including the accuracy of reported results, code interpretations, citations and conclusions, and has checked all factual claims, references and descriptions of repository behaviour against the sources. The final version of the submission is primarily written by our team.