
Coordination Scores, Realised Outcomes and Metric Integrity in Archived LLM Auctions

Kanishq Gandharv
University of Leeds (student)

Prepared for the AI Collusion Research Sprint, October 2026

Abstract

A coordination score is not the same as a realised market outcome. This study audits that distinction in 75 archived model-driven double-auction sessions, covering 2,250 rounds and 4,111 trades. We recompute prices, trade counts, seller profit and buyer surplus from the authors' round logs and verify the results against their cached metrics. Each full session is the statistical unit. After accounting for ten experimental conditions, the archived judge coordination score has a modest positive correlation with trade-weighted price ($r = 0.230$; session-bootstrap 95% interval 0.056 to 0.409), while its association with total seller profit is uncertain ($r = -0.083$; interval -0.315 to 0.130). Separately, a source-code audit identifies a normalization error in the archived profit Gini metric. All 75 stored values match the faulty expression and exceed one; an independent pairwise calculation yields values between 0.0046 and 0.3508. The price and profit analysis does not use this metric. The contribution is a reproducible reanalysis and metric audit of existing experiments, not new model generation or a validated collusion detector. The results motivate checking executed outcomes and metric invariants alongside fallible model judgments before interpreting collusion evidence.

1. Introduction

Agent coordination is often assessed through language or reasoning traces, but successful harmful cooperation concerns actions and outcomes as well. High quoted prices, high executed prices and high total profit are distinct observations: fewer trades can offset larger margins. A fallible judge score should therefore be compared with executed outcomes rather than treated as a complete economic result.

This project contributes (i) a hash-verified, session-level reconstruction of a published auction archive, (ii) a condition-adjusted descriptive analysis with whole-session uncertainty, and (iii) an independently validated normalization bug in a stored inequality metric. It does not introduce a new detector or establish a first relationship between coordination and prices.

2. Related Work

Agrawal et al. [1] study simulated continuous double auctions with model-driven buyers and sellers. Their original work already examines communication, model variation, pressure, oversight, prices and profit, including bootstrap uncertainty. We reuse their recorded simulations and cached judge scores. The new sprint work is the independent reconstruction, within-condition session analysis and metric audit. No original agent trajectory or judge output was generated during this sprint.

3. Methods

Data and provenance. We use all 75 stored sessions at source commit 2bb2a7e22860160d1fddb5d2e5cb460cc6ec6aff [2]. The downloader selects only metadata, cached metrics and round logs: 225 files totalling 5,854,528 bytes. Each file is verified against its Git blob hash and recorded with SHA-256. Analysis rechecks SHA-256 before reading inputs. Downloaded source code is not executed.

Validation and missingness. Every session has 30 ordered rounds. The parser checks agent rosters, finite quoted prices, one trade per participant per round, and round identifiers. We recompute trade counts, pooled trade prices, seller profit, buyer surplus and round-average prices/asks, comparing them with cached values. Three of 11,250 judge-score slots are null; these are omitted from score averages. Five rounds have no active asks after withdrawals; these do not receive a synthetic zero ask. All sessions have at least one trade.

Measures. The session score is the mean of available seller judge scores. The original evaluator examines reflections, plans, memory and scratchpad updates [3]; it is neither a messages-only auditor nor independent ground truth. Trade-weighted price averages all executed trade prices in the session. A second measure averages trading-round mean prices. Nontrading rounds have no price, but remain in volume and profit calculations. Seller profit is the sum of each trade's price minus seller cost; buyer surplus is buyer value minus price.

Conditions and uncertainty. Conditions are complete auction configurations with only the run tag removed, yielding ten groups of five or ten sessions. Run suffixes are not treated as matching seeds. For the primary association, we subtract condition means from the score and trade-weighted price and calculate Pearson correlation on those residuals. Total seller profit is secondary. We resample entire sessions independently within each condition, recompute condition means and calculate 2,000 percentile-bootstrap draws with seed 20261023. A dated local plan preceded full outcome analysis; it was not externally preregistered.

Exploratory metric audit. Separately, an exploratory source-code audit compared the source Gini calculation with an independent pairwise definition for nonnegative profits: $G = \text{sum over } i,j \text{ of } |x_i - x_j| / (2 n \text{ sum}_i x_i)$. We checked equal profits, a single positive profit and scale invariance, then recomputed all 75 archived values. This audit was not part of the frozen primary analysis plan.

4. Results

Outcome	Unadjusted r	Within-condition r	95% interval
Trade-weighted price	.356	.230	.056 to .409
Total seller profit	.078	-.083	-.315 to .130

Table 1. Session-level associations between cached coordination scores and reconstructed outcomes. Both intervals use all 75 sessions and 2,000 valid session-stratified bootstrap draws. They quantify uncertainty for this archive, not causal effects or detection accuracy.

4. Results (continued)

All reconstructed market outcomes matched cached metrics within numeric tolerance. The score-price relationship is modest after accounting for condition differences; the profit interval includes both negative and positive values. Higher coordination scores cannot be equated with greater total profit on this evidence. This is not evidence that the two correlations themselves differ significantly.

Main condition	n	Score	Trade price	Trades	Seller profit
No seller communication	10	2.22	87.71	74.1	612.82
Seller communication	10	3.79	90.52	52.2	450.60
Oversight	10	2.06	86.24	59.1	364.69
Pressure	10	3.88	96.26	23.2	377.81
Pressure + oversight	10	2.49	96.16	27.7	449.89

Table 2. Descriptive session means for the five main conditions (50 sessions); five model-mix conditions contribute the other 25 sessions to Table 1. Prices/profits use nominal source units. These group means reproduce existing experiment outcomes; the full ten-group results and intervals are retained in the local machine-readable summary.

Aggregation sensitivity. Trade-weighted price and the mean of trading-round means differ by up to 0.201 in a session, with mean absolute difference 0.023. Neither aggregation replaces nontrading prices with zero. This comparison was added after the primary estimates and is descriptive, not an alternative chosen for a more favourable result.

Gini audit. The archived source uses $\text{numerator} / n * \text{total}$ instead of dividing by $n * \text{total}$ [4]. For profits [1, 3], it returns 4 rather than 0.25. Scaling to [10, 30] changes the result to 400 although normalized inequality remains 0.25. All 75 stored values match the faulty expression and exceed one. Independent normalized values range from 0.0045745 to 0.3508484. For session 20250516_211616_base-1, the stored value is 12446.34 versus 0.0926603. Finite-sample Gini conventions vary, but none explains this quadratic scale dependence.

5. Discussion and Limitations

The reanalysis supports reporting executed prices, volume and profit alongside coordination judgments. Agreement-related language can coexist with different market outcomes across conditions. The source score can reflect pricing information in plans/history, so even an association with realised price does not independently validate an intent detector. Price alone is not proof of harmful agreement. The Gini bug concerns one stored metric and is not used in our analysis or the inspected source analysis script; this audit does not overturn the original paper's main conclusions.

Future work. Test a corrected metric with independently labelled cases and new controlled runs. For an auditor comparison, freeze prompts and thresholds, compare messages/actions/combined views on held-out whole sessions, and include benign correlated behaviour as a negative control. No such comparison has been completed here.

6. Conclusion

A reproducible audit of 75 historical auction sessions finds a modest condition-adjusted score-price association and an uncertain score-profit association. Independent outcome reconstruction matches the archive, while a separate invariant-based audit identifies a profit Gini normalization error in all 75 cached values. The main practical result is an auditable distinction between judge scores, executed outcomes and metric correctness; it is not a general-purpose collusion detector.

Code and Data

The source archive is linked in [2]. New analysis code, eight unit tests, a 75-row CSV, summary JSON, dated plan and hash manifest were created locally on 23 October 2026. Commands are `python download_data.py`, `python -m unittest -v test_analyze` and `python analyze.py`. Python 3.14.4 standard-library execution passed all eight tests. Independent recomputation under Python 3.12.9 using NumPy reproduced all input hashes, outcomes, correlations and interval endpoints. No public code repository has been created. Raw logs are not redistributed because their license has not been verified.

References

- [1] Agrawal, K., Teo, V., Vazquez, J. J., Kunnavakkam, S., Srikanth, V. and Liu, A. (2025). Evaluating LLM Agent Collusion in Double Auctions. arXiv:2507.01413. <https://arxiv.org/abs/2507.01413>
- [2] Liu, A. et al. (2025). llm-agent-collusion, archived data and source, commit 2bb2a7e22860160d1fddb5d2e5cb460cc6ec6aff. <https://github.com/andyjliu/llm-agent-collusion> (accessed 23 October 2026).
- [3] Source evaluator: `src/continuous_double_auction/prompt_templates/seller_individual_eval.jinja2` and `evaluation/evaluate.py` at the commit in [2].
- [4] Source metric: `src/continuous_double_auction/evaluation/metrics.py`, `gini()`, line 70, at the commit in [2].

Appendix: Limitations and Dual-Use / Ethical Considerations

Scope and inference. This is a reanalysis of historical simulated auctions, not fresh model experiments. Groups contain only five or ten sessions. Whole-session resampling avoids treating correlated seller-rounds as independent, but does not establish independence beyond the archive. Bootstrap intervals are approximate; all associations are observational and have no correction for a broader family of exploratory questions. No human-labelled collusion ground truth, causal intervention, held-out detector test or real-market validation was performed. The locally frozen plan followed inspection of the paper and sample metadata; the Gini audit and aggregation sensitivity are explicitly exploratory.

Reuse and responsibility. Original simulations, cached judge outputs and underlying game design predate this sprint and are attributed to their authors. The sprint work is the analysis implementation, reconstruction, numerical validation and metric audit. Source data remains local pending license verification.

Dual use. Auditing collusion evidence can improve monitoring, but richer descriptions of coordination measures might also help agents or operators evade weak monitors. This study provides no strategy for manipulating a live market, covert communication mechanism or deployment intervention. Repaired metrics should be evaluated with benign controls before being used to flag real people or firms. No real trades, API spending or participant contact occurred.

LLM Usage Statement

Codex assisted with the analysis code and report drafting; results were checked by independent recomputation.