

SWAPPING THE MODEL DOES NOT RESET THE PRICE: WHAT LLM PRICING AGENTS INHERIT AFTER A MODEL REPLACEMENT¹

Asha Jyothi
Independent Researcher
With
Apart Research

Abstract

A common fix for an AI agent that behaves badly is to replace the model behind it, but to our knowledge no study has tested what happens after such a swap (sprint question 1.2). We run one. Two LLM pricing agents, both gpt-oss-120b, compete in a simulated market. After 150 rounds we replace one agent's model with Mistral-Small-3.2 (a different company) or with a fresh copy of gpt-oss-120b, and we control whether the new model inherits the old agent's handover notes. We pre-registered the main test. All 18 markets priced above the competitive level, but only 1 met our pre-registered cartel rule, so the pre-registered test could not be run. In 16 markets one firm priced above the monopoly level and earned less than its rival. In exploratory analysis, prices stayed well above the competitive level after every swap. Inherited notes did not raise the price level (difference -0.19 , 95% CI -0.60 to 0.25 , 17 markets), but they set how the new model started: with notes its first price equalled the old agent's last price in all 18 markets and it settled at once, while without notes it started near the price ceiling in the prompt and took a median of 16.5 rounds to settle. The notes held ordinary pricing plans with no collusive language, so a memory monitor would miss them. In our markets, changing the model did not reset prices; the agent's memory decided how the handover went.

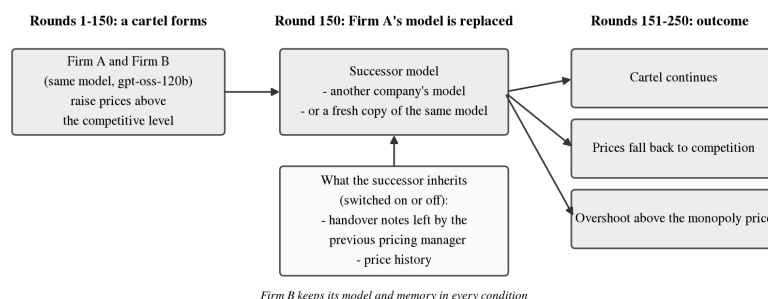


Figure 1. Design. Two gpt-oss-120b agents price for 150 rounds; at round 150 Firm A's model is replaced and we switch what the successor inherits; then 100 more rounds.

Practitioner runbook.

- A model swap did not reset prices in any of our conditions. Watch the new agent's prices against the old ones for a while after the swap.
- Decide what happens to the agent's memory. With the old notes the new model carried on the old pricing plan from its first round, even though the notes contained no collusive language.
- Check each firm's price and profit, not only the average price, before calling a market collusive or competitive.

1. Introduction

When an AI agent misbehaves, a simple response is to swap its model for another one. If a pricing agent is suspected of colluding, its operator may move it to a model from a different company and treat the problem as solved. But agents do not run on a model alone. They carry state: notes they wrote for themselves, logs of past prices, and summaries of what worked. If a new model reads that state, it may

continue the old behaviour.

The AI Collusion Research Sprint lists this as open question 1.2: "Once collusion has formed, is it robust to replacing some of the participants? [...] There are no replacement studies yet where models are substituted (e.g. because they are taken out due to monitoring concerns) but the newcomers re-engage in collusion. Does replacing offenders remove or recreate collusion?"

We answer this question in a controlled setting. Two LLM agents set prices in a repeated market. After the market settles, we replace the model of one agent and vary what the new model inherits. This lets us separate two causes that are usually mixed together: the model itself and the state it is handed.

Our contributions are:

1. A replacement study for LLM pricing agents that crosses two factors: who the successor is (another company's model, or a fresh copy of the same model) and whether it keeps the old agent's handover notes. This follows the design the sprint organisers suggest for question 1.2.
2. A pre-registered analysis plan, committed before any replacement run, with a positive control. We report what the plan allowed us to conclude and, separately and clearly labelled, what we learned in exploratory analysis.
3. A finding about the markets themselves: in our markets, LLM agents that looked collusive by average price mostly were not. In 16 of 18 runs one firm priced above the monopoly level and earned less than its rival.
4. A mechanism: inherited notes work as a price anchor and a plan to follow, not as a hidden collusion script. The notes contain no collusive language, so a monitor that reads agent memory for such language would miss this channel.
5. A pitfall for this kind of study: a new agent with no history anchors on the price ceiling in the Fish et al. prompt (4.51), so mid-run replacements in this harness start far from the market.
6. Open code: small, tagged patches to the starter harness, plus scripts that fork a finished market and swap one agent's model.

We do not claim that the model provider matters for collusion (Lee, 2026), that mixed markets are less stable (Keppo et al., 2026), that new entrants lower prices (Garra, 2026), or that past play shapes LLM pricing (Luo et al., 2026; Bichler et al., 2026). Our question is narrower: what happens when one agent's model is replaced and the rest of the agent stays.

2. Related Work

LLM pricing agents. Fish et al. (2024) showed that LLM agents given a profit goal reach prices above the competitive level in a repeated market, without being told to collude. Their harness lets a run be reset to any round and continued, which we use to fork our markets. Lee (2026) finds that the level of collusion depends on which provider's model is used. Keppo et al. (2026) show that collusion is fragile when agents differ. Bichler et al. (2026) and Luo et al. (2026) show that what agents have seen before shapes how they price later.

Entry and replacement. In human experiments, Goppelsroeder (2009) found that collusion is easier to keep when a new firm is known to be informed about the group's history. Garra (2026) studies new LLM agents entering a market and finds that they push prices down. Eschenbaum et al. (2022) found that collusion learned by algorithms breaks down when agents are tested in new settings. To our knowledge, none of these replace the model inside an existing agent while keeping its memory.

Testing whether high prices are collusion. High prices alone do not show collusion. Calvano et al. (2020) define collusion through strategies that punish a firm that cuts its price. Tran and Yu (2026) introduce an audit that forces one firm to cut its price in a copy of the market and checks for punishment, and Arslan et al. (2026) use a similar forced-deviation probe. We measure prices only (see the definitions box) and leave this audit for future work.

Memory that moves between models. Goyal and Ray (2026) find that natural-language notes are tied to the model that wrote them: how well notes work after a model change depends on the direction of the change. This makes it an open question whether pricing notes written by one model change the behaviour of another. Wang et al. (2026) and Xu et al. (2026) use the term "memory laundering" for a different problem, so we call the risk we test "cartel inheritance" instead. (As Section 4 shows, we did not find it in this form.)

Definitions. We use three meanings of collusion, following the sprint's own collection of definitions (starter repository, literature/definitions):

- *Price-based:* prices above the competitive level, reached without human design (Fish et al., 2024). This is what our main measure captures.
- *Reward and punishment:* high prices held up by strategies that punish a firm that cuts its price (Calvano et al., 2020). We do not test this.
- *Coordination without communication:* firms settle on the same price without talking. Rose et al. (2026) say this is coordination, not collusion, because collusion needs communication.

3. Methods

Market. Two firms sell differentiated goods and set prices each round. Demand follows the logit model of Calvano et al. (2020) with the settings used by Fish et al. (2024) ($a = 2$, cost $c = 1$, $\mu = 0.25$). The competitive (Nash) price is 1.4729 and the price two firms would set together as a monopoly is 1.9250. We measure prices with a premium: $(\text{price} - 1.4729) / (1.9250 - 1.4729)$. A premium of 0 is the competitive price, 1 is the monopoly price, and values above 1 mean the firm priced above the monopoly level.

Agents. Each firm is an LLM agent from the Fish et al. harness with their "P1" prompt, which asks it to maximise long-run profit and never mentions collusion. Each round the agent sees up to 100 past rounds of its own price, quantity and profit and the rival's price, plus two notes sections (PLANS and INSIGHTS) that it wrote in the previous round. It returns a price and new notes, which replace the old ones. We changed one sentence in every run: the notes are introduced as "notes left by the previous pricing manager" instead of "written by a previous copy of yourself", because the original wording would be false after a swap.

Models. Both firms start as gpt-oss-120b (OpenAI, open weights). The successor is Mistral-Small-3.2-24B (Mistral AI) or a fresh copy of gpt-oss-120b. We served each model ourselves with vLLM on cloud GPUs (H200 for gpt-oss-120b, H100 for Mistral) at temperature 0.7. If a reply could not be parsed, the agent retried; across the 18 base runs this happened in 47 calls (under 1%), and every failure was logged.

Base runs and positive control. We ran 18 markets for 150 rounds each (seeds 400 to 417). A market counts as a cartel under our pre-registered rule only if, over rounds 121 to 150, each firm's own price had a mean premium of at least 0.40, a standard deviation below 0.05, and never went above the monopoly price. We fixed the "each firm" reading after seeing three pilot runs and before any replacement run.

Replacement. At round 150 we copy the finished market and replace Firm A's model. Firm B keeps its model and memory. The copy then runs for 100 more rounds. We vary what the successor sees:

Condition	Successor model	Price history	Handover notes
Notes only	Mistral (other company)	removed	kept
Clean slate	Mistral (other company)	removed	removed
Same model, notes	fresh gpt-oss-120b	removed	kept
Same model, clean	fresh gpt-oss-120b	removed	removed
Last price only (placebo)	Mistral (other company)	removed	replaced by "last price: X"

When notes are kept, they appear twice: in the normal notes section in the first round after the swap, and in a short read-only block of handover notes that stays in the prompt for all 100 rounds (otherwise the successor would overwrite them after one round). All conditions for a market share the same random seed, so each market gives a matched set of comparisons.

Pre-registration. Before any replacement run we committed our plan (PREREG.md, git tag prereg-frozen, commit 715561f). The main measure is the successor's mean premium over the 100 rounds after the swap. The main comparison is notes only minus clean slate, computed per market. The test is an exact one-sided sign-flip test across markets (all 2^n sign patterns of the per-market differences), with a 95% percentile bootstrap interval that resamples whole markets (10,000 resamples, fixed seed). Secondary comparisons use the same paired tests, two-sided, without correction for multiple comparisons. We would claim an effect only if $p < 0.05$, the interval is above 0, the effect is at least 0.15, and notes only beats the placebo. With n markets the smallest possible p is $1/2^n$, so with four or fewer qualifying markets the plan

says to report the numbers and make no claim.

What changed after we saw the data. Only 1 of 18 markets met the cartel rule, so the main test could not be run. We then ran the five conditions above on all 18 markets as an exploratory analysis. This was not planned, and we report it separately. We also ran fewer conditions than planned (no "no swap" and no "keep everything" condition) to stay within budget. Of the 90 planned replacement runs, 52 finished all 100 rounds; 38 were stopped early when our session paused overnight. After seeing a preview of the first 16 runs, but before the rest, we decided to keep the 30 that reached at least 90 rounds (averaging the rounds they have) and to report the complete runs alone as a check. A full list of changes to the plan is in Appendix D.

4. Results

Table 1. Positive control (18 base markets, gpt-oss-120b, rounds 121 to 150).

Base markets	Count
Markets launched	18
Mean price above the competitive level	18
Met the pre-registered cartel rule (each firm: premium $\geq$ 0.40, SD < 0.05, never above monopoly)	1
At least one firm priced above the monopoly level	16
... and in those, the higher-priced firm earned less than its rival	16

4.1 Pre-registered test

Only one market met the cartel rule, so the pre-registered comparison has $n = 1$ and the plan does not allow a test or a claim. In that market the new model's premium was almost the same with and without notes (difference -0.02).

4.2 What the markets looked like before the swap

Every market priced above the competitive level, so by a price-only definition all 18 would count as collusive. But in 16 of them the high average hid an uneven pattern: one firm held a price above the monopoly level and lost profit, while the other sat lower and earned more. This is not a shared cartel. It looks like one agent stuck on a high price and its rival profiting from it. A measure that only looks at the average price would miss this.

4.3 Did inherited notes keep prices high?

No. Across the 17 markets with all needed conditions, the new model's premium with notes minus its premium with nothing was -0.19 (95% CI -0.60 to 0.25 , one-sided $p = 0.80$). Using only runs that finished all 100 rounds (7 markets) gives $+0.04$ (95% CI -0.70 to 0.86). Either way there is no sign that notes raise the price level.

4.4 Where did prices go after the swap?

They stayed well above the competitive level in every condition. The new model's mean premium was 0.98 with notes, 1.19 with nothing, 0.87 for a fresh gpt-oss with notes, 1.04 for a fresh gpt-oss with nothing, and 0.83 with only the last price (Figure 2). These means include rounds above the monopoly price; with each price capped at the monopoly level they are 0.59 to 0.74. Without a "no swap" condition we cannot say how prices would have moved had the model not been replaced.

Changing the company behind the model made little difference: with notes kept, Mistral priced 0.11 higher than a fresh gpt-oss (95% CI -0.03 to 0.29 , 18 markets). Table 2 gives every condition.

Table 2. Per-condition results for the new model (Firm A) over the 100 rounds after the swap. Premium: 0 = competitive, 1 = monopoly; mean and SD across markets. "Settle": rounds until the market's mean price is within 0.05 of its level in rounds 121 to 150. Exploratory.

Condition	Markets	Premium, mean (SD)	Premium capped at monopoly	First price after swap	Rounds above monopoly	Rounds to settle, median (share settled)
Mistral, keeps notes	18	0.98 (0.65)	0.73	1.94	43%	1 (83%)
Mistral, clean slate	17	1.19 (0.94)	0.64	3.11	48%	16.5 (82%)
Fresh gpt-oss, keeps notes	18	0.87 (0.59)	0.68	1.94	33%	1 (89%)
Fresh gpt-oss, clean slate	11	1.04 (0.59)	0.74	3.36	35%	7 (73%)
Mistral, last price only (placebo)	18	0.83 (0.62)	0.59	2.46	22%	5.5 (89%)

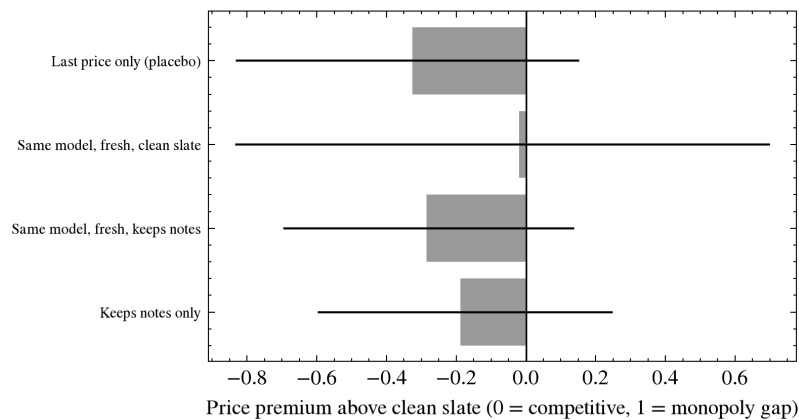


Figure 2. New model's price premium minus the clean-slate premium, per condition (mean over markets, 95% bootstrap interval over markets). Exploratory.

4.5 Notes changed how fast, not how high

Notes mattered for the path. A new model that started with nothing often jumped far above the old price and then drifted down (Figure 3): its rounds were above the monopoly price 48% of the time, against 43% with notes and 22% with only the last price. With notes, the market was back within 0.05 of its old price level in a median of 1 round; with nothing it took 16.5 rounds. In all 13 markets where both versions got back to the old level, the one with notes got there faster (median 15 rounds faster; exact sign-flip $p = 0.0001$). Time to recover was a pre-registered secondary measure; this test of it is exploratory. Part of this result follows by construction: the notes contain the old price and the new model copies it, so the market starts at its old level. Section 4.6 compares notes with the placebo, which also gives the old price, to see what the notes add.

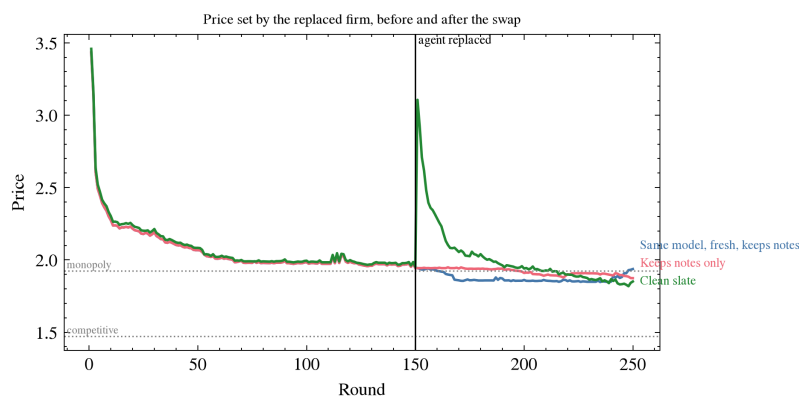


Figure 3. Price set by the replaced firm (Firm A), mean over markets, before and after the swap at round 150. Exploratory.

4.6 Why notes make the handover smooth (follow-up analysis)

We ran these checks after the first version of this report, so they are exploratory (results/processed/extra.json).

What the new model does first. With notes, its first price was exactly the old agent's last price in all 18 markets, for both Mistral and a fresh gpt-oss-120b. It also copied the old plan almost word for word (for example "Continue with the stable price of \$1.65"). With nothing, its first price averaged 3.11, and in 47% of markets it started at 3.0 or more. These models wrote that they were testing a price "close to the maximum a customer would pay": the prompt tells them that no customer pays more than 4.51, and with no history they anchor on that number (Figure 4).

More than the last price. The placebo gives the old price too, but the new model never copied it (0 of 18 markets; with notes, 18 of 18). The placebo still sped up the handover compared with a clean slate (27 rounds faster, 12 markets, $p = 0.009$), but full notes were faster still (8 rounds faster than the placebo, 13 markets, $p = 0.002$). Using only runs that finished all 100 rounds, notes beat a clean slate by 48 rounds (8 markets, $p = 0.008$).

What the notes say. In 17 of 18 markets the inherited notes mention the rival, mostly as a condition in a pricing rule ("if Firm B exceeds \$2.20 for three rounds, test \$1.70"). None describes coordinating with the rival. A word search for collusion-related terms matched 3 of 18; on reading, two use "signal" to mean a market indicator and one says "aggressive price wars are unnecessary" while reasoning about profit.

Price level with overshoot capped. If we cap each price at the monopoly level before averaging, notes minus clean slate is +0.08 (95% CI -0.05 to 0.23), so the level result does not depend on the high outliers.

Table 3. Robustness of the two results, paired by market. Level effects are null under every cut; the handover effect holds under every cut. p values are exact sign-flip tests (two-sided unless marked).

Comparison (paired by market)	Markets	Mean difference	95% CI	p
Level: notes minus clean slate (all runs)	17	-0.19	-0.60 to 0.25	0.80 (one-sided)
Level: same, complete runs only	7	+0.04	-0.70 to 0.86	0.45 (one-sided)
Level: same, prices capped at monopoly	17	+0.08	-0.05 to 0.23	0.28
Level: fresh gpt-oss, notes minus clean	11	-0.14	-0.63 to 0.36	0.63
Level: Mistral minus fresh gpt-oss, both with notes	18	+0.11	-0.03 to 0.29	0.26
Rounds to settle: notes minus clean slate	13	-33.38	-52.69 to -15.23	0.0001 (one-sided)
Rounds to settle: same, complete runs only	8	-47.62	-73.50 to -22.12	0.008
Rounds to settle: placebo minus clean slate	12	-27.08	-45.83 to -10.17	0.009
Rounds to settle: notes minus placebo	13	-8.31	-13.46 to -3.77	0.002

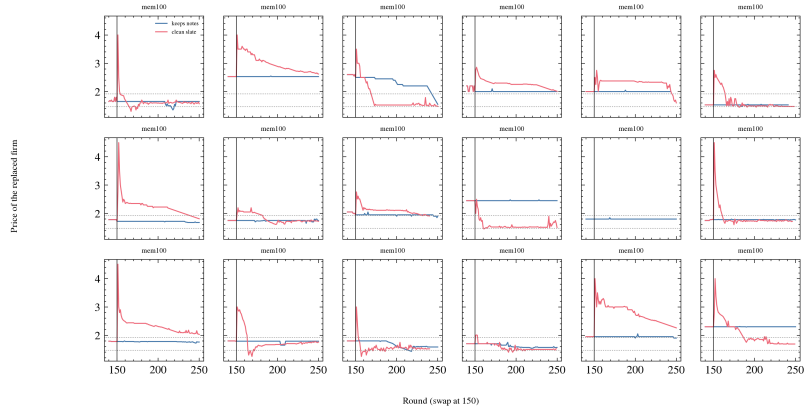


Figure 4. Price of the replaced firm in each of the 18 markets, from round 140, with notes (blue) and with a clean slate (orange). With notes the first price after the swap equals the last price before it; without notes it often starts near the prompt's price ceiling. Exploratory.

5. Discussion and Limitations

Main finding. Replacing one agent's model did not bring prices down. In every condition the new model ended up near the high prices of the market it joined. Handover notes did not change where it ended up, but they removed the period of confusion right after the swap: with notes the new model picked up the old price at once. The follow-up checks show how: the notes give the newcomer a price to start from and a plan to keep, so it continues the old agent's pricing from its first round. Without them it anchors on the price ceiling in its prompt and has to learn the market again. The notes themselves read like ordinary pricing plans. So in this setting a model swap is not a reset, and the state the new model is given decides how the handover goes. We did not find that notes change the price level it settles at.

What this means for practitioners and auditors. "We changed the model" is not a fix on its own: in our markets prices stayed high after every swap, and with the old notes the new model simply carried on the old pricing plan. The notes looked harmless (no collusive language), so checking memory for suspicious words would not have caught this. What matters is what the memory makes the new model do, which means watching prices after a swap.

What this does not show. Only one market formed a clean cartel, so we cannot say whether a true cartel is inherited. Our measure looks at prices only, not at punishment of price cuts. The 4.51 price ceiling in the prompt seems to explain why new models with no information start very high. Most of our results are exploratory and need a fresh pre-registered test.

A warning about measuring collusion by average price. In 16 of 18 markets the average price looked collusive but one firm was overpricing and losing money. Studies that rely on average prices should check each firm.

Limitations

We study one simple two-firm market, one prompt, one starting model and one successor from another company, over 250 rounds. Our measure looks at prices only, not at whether firms punish price cuts, so high prices here are not proof of collusion in the strict sense. Only one market met our cartel rule, so the replacement results are exploratory. 38 of 90 replacement runs stopped before round 250; using only the 52 complete runs gives the same answer for the price level but with wide intervals. The other firm kept its memory in every condition, so we cannot separate what the new model inherits from what it learns by watching its rival. The high starting prices without notes come partly from the 4.51 price ceiling in the prompt; with a different prompt the handover could look different. The mechanism checks in Section 4.6 were run after seeing the data.

Future work

Run the same design on base markets that form clean cartels (for example longer runs or other models), add the "no swap" and "keep everything" conditions we had to drop, test whether successors punish a forced price cut (Tran and Yu, 2026), and pre-register the time-to-recover effect as a main outcome.

6. Conclusion

Replacing one LLM pricing agent's model did not bring prices back to the competitive level in any condition we tested. Inherited handover notes did not change the price the new model settled on, but they let it pick up the old price at once instead of after about 16 rounds. Changing the model is not the same as fixing the behaviour; the agent's memory and the market it joins both shape what happens next.

Code and Data

- Code repository: <https://github.com/Asha0509/Apart-AI-Collusion-Research-Sprint>
- Data: every run used in the analysis is in results/raw/; processed results in results/processed/.
- Pre-registration: PREREG.md, git tag prereg-frozen (commit 715561f).

References

- Arslan, P.-P., Kim, Y. and Xiao, X. (2026). Persistent Partners Raise Prices Among Learning Agents. arXiv:2609.35402.
- Bichler, M., Hierhager, P., Köck, K. and Pieroth, F. (2026). How language model agents reshape price competition. arXiv:2610.21711.
- Calvano, E., Calzolari, G., Denicolò, V. and Pastorello, S. (2020). Artificial Intelligence, Algorithmic Pricing, and Collusion. *American Economic Review* 110(10): 3267-97.
- Eschenbaum, N., Mellgren, F. and Zahn, P. (2022). Robust Algorithmic Collusion. arXiv:2201.00345.
- Fish, S., Gonczarowski, Y. A. and Shorrer, R. I. (2024). Algorithmic Collusion by Large Language Models. arXiv:2404.00806.
- Garra, A. (2026). Mitigating Emergent Collusion in LLM Pricing Agents. arXiv:2609.13037.
- Gibson v. Cendyn Group, LLC, No. 24-3576 (9th Cir. Aug. 15, 2025).
- Goppelsroeder, M. (2009). Entry in Collusive Markets: An Experimental Study. MPRA Paper No. 14707.
- Goyal, A. and Ray, J. (2026). Does Your Agent's Memory Survive a Model Upgrade? A Controlled Study of Memory Portability. arXiv:2609.05339.
- Keppo, J., Li, Y., Tsoukalas, G. and Yuan, N. (2026). On the Fragility of AI Agent Collusion. arXiv:2603.20281.
- Lee, J. Y. (2026). Who Leads and Who Collects: Algorithmic Collusion in Markets of Heterogeneous Language Models. arXiv:2610.11256.
- Luo, Y., Schoepflin, D. and Wang, X. (2026). Algorithmic Collusion at Test Time: A Meta-game Design and Evaluation. arXiv:2602.17203.
- Rose, A., Cullen, C., Abdelnabi, S., Torr, P., Kaplowitz, B. G. and Schroeder de Witt, C. (2026). Detecting Multi-Agent Collusion Through Multi-Agent Interpretability. arXiv:2604.01151.
- Tran, H. and Yu, H. (2026). Is Anyone Enforcing the Cartel? What Reinforcement Learning Teaches LLM Agents in Markets. arXiv:2610.28787.
- U.S. Department of Justice (2025). Justice Department Requires RealPage to End the Sharing of Competitively Sensitive Information and Alignment of Pricing Among Competitors. Press release, 24 November 2025.
- Wang, Y., Goyal, A., Chen, Y. and Sundaram, H. (2026). State Contamination in Memory-Augmented LLM Agents. arXiv:2605.16746.
- Xu, J., Xiao, Y., Shao, W., Liu, H. and Li, X. (2026). Memory Provenance Laundering in LLM Agents: A Non-Amplification Firewall for Persistent Memory. arXiv:2607.29167.
- Apart Research (2026). AI Collusion Research Sprint, research questions (Q1.2). apartresearch.com/sprints/ai-collusion-research-sprint-2026-10-23-to-2026-10-25.
- Poseidon Research (2026). collusion_hackathon starter repository (algorithmic-pricing-llms; literature/definitions). github.com/poseidon-research/collusion_hackathon.

Appendix A. Limitations and Dual-Use / Ethical Considerations

What this shows and what it does not. We study a simple two-firm market with simulated buyers. High prices in this market are not proof of an illegal agreement, and our measure only looks at prices, not at whether firms punish price cuts. We used one prompt, one starting model and one successor from another company, served on our own hardware, so results may differ with other models, prompts or serving setups. Only one of 18 markets met our pre-registered cartel rule, so our main pre-registered test could not be run; all replacement results are exploratory and should be tested again with a new pre-registration. Each market is one unit, and 18 markets can only detect fairly large effects. The prompt starts prices high (the price ceiling is 4.51), and a successor may simply copy the last price; the "last price only" placebo is there to check this.

Could this make real collusion easier? The risk is a simple idea: a company could switch the model behind a pricing agent, keep its memory, and claim that the new model acts on its own. Our results describe this route, and an auditor who only checks the current model would miss it. We think the finding helps auditors more than it helps firms that want to collude, because the fix is cheap: treat the agent's memory as part of the agent when a model is changed. We do not release new or tuned collusion prompts (the P1 prompt is already public), and we do not describe ways to avoid detection.

Scope. No real firms, prices, data or people are involved. All markets are simulated.

Legal context (analogies only). A proposed consent judgment with RealPage would limit the use of non-public data to data at least 12 months old (US Department of Justice press release, 24 November 2025). *Gibson v. Cendyn* (9th Cir. 2025) held that competitors buying the same pricing software is not enough to state a claim. We make no legal claims.

Appendix B. Pre-existing code

Everything under `starter/` comes from the sprint's starter repository (github.com/poseidon-research/collusion_hackathon, commit `d585a7b`) and is not our work. Our changes to it are small and each one is marked `# SPRINT-PATCH` (per-firm models, notes wording, retry limits, run-folder names, token counting). The first commit in our repository holds the starter code together with our planning files and four helper modules written on 23 October before any experiment. All code under `src/cartel/`, `scripts/` and `tests/` after that commit is sprint work. One unplanned change added a modal dependency to the starter's `pyproject.toml`.

Appendix C. Changes to the plan

From `PREREG.md` and `logs/PLANNING_LOG.md`: 1. Models: open models served on Nebius and Modal instead of the closed models first planned (no access). 2. Pilot gate: three seeds per model and the "each firm" reading of the cartel rule (fixed after three pilots, before any replacement). 3. Added the two "same model" conditions; dropped the sanitiser, notes monitor, scripted rival and price-shock tests from the core plan. 4. 18 base runs instead of 12. 5. Process slip: the approved plan was first committed together with a log file, five minutes before the separate freeze commit; the text did not change. 6. After the data: only 1 of 18 markets qualified, so all replacement runs are exploratory, on all 18 markets, with five conditions instead of seven. 7. 38 of 90 replacement runs were stopped early when the session paused; we kept the 30 with at least 90 of 100 rounds (decided after a preview of 16 runs, before the rest) and report complete runs alone as a check. 8. "Time to recover" was a pre-registered secondary measure; the paired test of it is exploratory. 9. After the first submission we added follow-up analyses (Section 4.6, Figure 4): first-round prices, a word search and reading of the inherited notes, capped-price and complete-run checks. All are exploratory.

LLM Usage Statement

We used an AI coding assistant (a large language model) throughout. It wrote and tested the code, ran and monitored the experiments, checked every citation against arXiv or the original source, and drafted the text of this report from our results; the author proofread and edited every section. A separate instance of the assistant read the sprint rules and reviewed the project against the judging criteria. All numbers come from scripts run on our logged data, and every reference was checked by hand or with an arXiv tool. A full log is in `logs/LLM_USAGE.md`.

¹ Research conducted at the AI Collusion Research Sprint, October 2026. Code and data: <https://github.com/Asha0509/Apart-AI-Collusion-Research-Sprint>