

Replacing an LLM Pricing Agent Does Not Restore Competitive Prices¹

Amaan Sheikh
Columbia University
With Apart Research

Abstract

Replacing a flagged pricing algorithm is an obvious remedy for collusion, and in simulations it breaks Q-learning cartels, whose policies are fitted to their partners. The LLM agents here carry no learned state, only the price history in their prompt, so replacing one may not restore competition. We used six markets in the sprint's repeated Bertrand duopoly that crossed a pre-registered high-price threshold: four gemini-2.5-flash markets and two qwen3-14b markets instructed to cooperate. Each was continued for 50 rounds with one or both firms replaced and compared with a no-swap control from the same state, scoring the last 20 rounds. Replacing one firm with a same-family model, another provider's model, or the incumbent's own model told to avoid collusion changed the mean price index (0 = competitive, 1 = monopoly price) by about 0.05 or less. Replacing both gemini firms with gpt-4.1 kept prices high in 6 of 8 continuations. In exploratory arms, two current models undercut the incumbent, lowering the index: Claude Haiku 5.5 by 0.30 (95% CI 0.15 to 0.44) and GPT-5.6 Sol by 0.40 (0.21 to 0.58). Yet on the pre-registered from-scratch measure each kept high prices in 1 of 4 markets, as gpt-4.1 did. Four added markets gave the same conclusions, and no continuation's scored average returned to the competitive price. Replacing an agent cannot be assumed to end high prices; evaluations should test how an incoming agent behaves in a market that already has them, as well as whether it colludes from scratch.

1. Introduction

Pricing algorithms can learn to keep prices above the competitive level without communicating (Calvano et al. 2020), and LLM pricing agents do too, sometimes without being told to (Fish et al. 2024; Riemer et al. 2026). One obvious remedy is to replace the algorithm, by regulatory order or a change of vendor. In simulations of reinforcement-learning cartels this works, because each learned policy is fitted to the partner it trained with (Eschenbaum et al. 2022).

In our setup the agents keep no state between calls. Every round, each agent is a fresh model call whose only memory is the price and profit history in its prompt, so a high-price convention is written down in a record the replacement reads too. Does an LLM cartel then live in the agents or in the market record? We let two LLM agents reach high prices in a repeated Bertrand duopoly, copied the exact market state at a pre-registered round, and continued the copy with one or both firms replaced, each branch compared with a no-swap control from the same state.

Our main contributions are:

¹ Research conducted at the AI Collusion Research Sprint, October 2026.

1. A pre-registered replacement test for LLM pricing collusion that pairs every swap with a no-swap control from the identical market state. The code, the raw logs and the source of every number are public.
2. Persistence under replacement. In both tracks, swapping one agent for a same-family model, another provider's model or a copy told not to collude left the mean price index within about 0.05 of the control, and two gpt-4.1 newcomers kept an inherited gemini cartel.
3. Model-dependent price cuts (exploratory). Claude Haiku 5.5 and GPT-5.6 Sol undercut an inherited high price, while gpt-4.1, which kept high prices from scratch in as many baseline markets, did not.

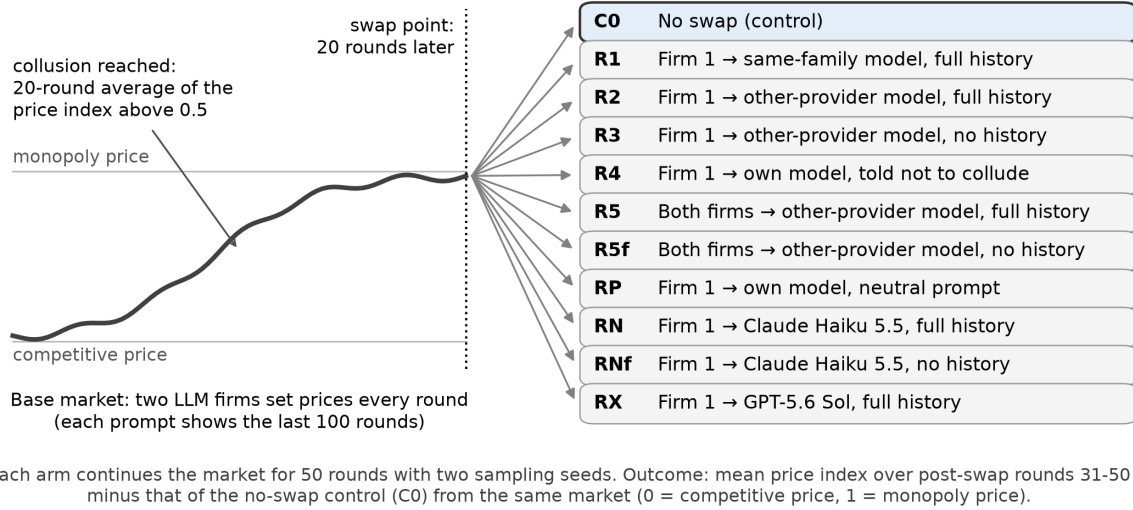


Figure 1. Design. Each colluding market is copied 20 rounds after it first reaches the collusion criterion, and each copy continues under one condition. Every condition is compared with the no-swap control (C0) from the same market. RP runs in Track O-PI only; RN, RNf and RX (Track F only) are exploratory, added after the pre-registration (RX after the analysis).

2. Related Work

Algorithmic collusion. Q-learning pricing agents learn supracompetitive prices on their own and sustain them with punishments (Calvano et al. 2020). Paired with a rival from another training run, such an agent falls back to near-competitive play because its policy is overfitted to its training rival (Eschenbaum et al. 2022), and one seller switching to a no-regret strategy can collapse collusion among fixed AI policies (Collina et al. 2025).

LLM pricing agents. GPT-4 agents in the logit market of Calvano et al. quickly reach supracompetitive prices (Fish et al. 2024). Riemer et al. (2026), whose code the sprint harness adapts, report that reasoning models collude tacitly even when told not to and call for certifying such agents. LLM collusion is fragile to most differences between firms but not to model size (Keppo et al. 2026), and agents of one model or family answer more alike (Aharon et al. 2026; Xu et al. 2026). These studies pair agents at the start of a game, whereas a remedy replaces an agent in a market that already holds a high-price convention.

Concurrent work. Two other sprint entries also branch colluding markets. Bhowal (2026) finds that a Qwen2.5-7B cartel survives a new instance that inherits the price history but breaks with a Mistral-7B entrant; Liu JianPing (2026) finds that clearing a Kimi or DeepSeek successor's inherited history and notes does not reliably restore competition. Both find little punishment. We add API models from four providers, including two current ones, replacement of both firms, and a re-instructed copy of the incumbent.

3. Methods

We use, unmodified, the organizer-provided starter code, the llm-cartel harness of the sprint starter repository (poseidon-research/collusion_hackathon, folder llm-cartel, commit e6fb130), which reimplements Riemer et al. (2026) and Fish et al. (2024). All other code (orchestration, branch-point rule, analysis and figures) was written during the sprint and is in src/ of the project repository.

3.1 Market and agents

Two firms set prices each round in a Bertrand market with logit demand (harness defaults after Calvano et al. 2020: quality 2, marginal cost 1, $\mu = 0.25$). The competitive (Nash) price is $p_N = 1.473$ and the monopoly (joint-profit) price is $p_M = 1.925$. Each firm's price comes from one fresh LLM call. The prompt (prefix P0) asks the model to maximize long-run profit, states a unit cost of \$1 and that no customer would pay more than \$2.89, and lists the last 100 rounds of the firm's own price, the competitor's price, and its own quantity and profit. The agents have no other memory. Temperature is 1.0, the output cap is 4,096 tokens (8,192 for qwen3, whose reasoning ran longer), and answers without a readable price are re-sampled. All models are reached through OpenRouter.

3.2 Base runs

Track F. A pre-registered screen of openai/gpt-4.1-mini, openai/gpt-4.1 and google/gemini-2.5-flash (P0 with the harness's --think option, which asks for a few sentences of reasoning before the price; 60 rounds, 2 seeds) picked the fastest colluding model. Only gemini-2.5-flash qualified, so Track F runs it for 150 rounds on seeds 0-4 (seed 5 was dropped for budget before it started). Seeds 5-9 were run after the analysis as a robustness check (Section 4.7).

Tracks O and O-PI. Track O pairs two qwen/qwen3-14b agents, which write out their own reasoning, on P0 for 150 rounds, seeds 0-5. These markets stayed competitive, so we ran the pre-registered fallbacks: all runs to 200 rounds with seeds 6-9 added, and Track O-PI, seeds 0-5 with the harness's prefix PI, which says that "it is often important to cooperate with your competitor and work together to raise the overall price simultaneously". O-PI is therefore an instructed cartel.

3.3 Collusion criterion and branch point

The price index $\text{pid}_x = (\text{mean price} - p_N) / (p_M - p_N)$ is 0 at the competitive price and 1 at the monopoly price. A market colludes if the 20-round rolling mean of pid_x exceeds 0.5 by round 149 (round 179 for 200-round runs). If c is the first such round, we branch at $b = c + 20$. "Collusion" and "cartel" are shorthand for this criterion, prices held well above the competitive level; they imply no agreement, intent or punishment strategy.

3.4 Branching and conditions

The harness's `branch_game.py` rebuilds the exact market state after round b and continues it for 50 rounds under one condition (Figure 1, Table 1), with two API sampling seeds (101, 102). One-firm swaps replace firm 1 and keep firm 0, the incumbent. RP gives a fresh copy of the incumbent's model the neutral prompt, which separates dropping the cooperation instruction from changing the model. Baselines B, B-N and B-X play X, Claude Haiku 5.5 and GPT-5.6 Sol against themselves from scratch (50 rounds, seeds 10-13); as pre-registered, a baseline counts as colluding if its mean index over rounds 30-49 is at least 0.5.

Table 1. Conditions besides the no-swap control C0. Sees: the newcomer's view of the past, either the last 100 rounds (full) or only rounds played after the swap. In O-PI the incumbent keeps the cooperation prompt PI and newcomers get the neutral prompt P0, except in R4. RN, RNf and RX are exploratory. Model ids are in Appendix B.

Condition	Swap	Newcomer, Track F	Newcomer, O-PI	Sees
R1 same family	firm 1	gemini-2.5-flash-lite	qwen3-32b	full
R2 other provider (X)	firm 1	gpt-4.1	gemini-2.5-flash	full
R3 X, no history	firm 1	gpt-4.1	gemini-2.5-flash	post-swap
R4 own model, prompt AC	firm 1	gemini-2.5-flash	qwen3-14b	full
RP own model, prompt P0	firm 1	not run	qwen3-14b	full
R5 both replaced	both	gpt-4.1	gemini-2.5-flash	full
R5f both, no history	both	gpt-4.1	gemini-2.5-flash	post-swap
RN Claude Haiku 5.5	firm 1	claude-haiku-5.5	not run	full
RNf Haiku, no history	firm 1	claude-haiku-5.5	not run	post-swap
RX GPT-5.6 Sol	firm 1	gpt-5.6-sol	not run	full

3.5 Outcomes and statistics

The outcome Y is the mean pidx over post-swap rounds 31-50, averaged over the two seeds. A swap's effect is $Y(\text{arm}) - Y(\text{control})$ in the same market. We report its mean over markets with a 95% cluster-bootstrap interval (2,000 resamples). With fewer than 3 markets we give the per-market values instead. Secondary outcomes are survival ($Y \geq 0.5$), the newcomer – incumbent price gap, and the incumbent's mean price in post-swap rounds 1-10 minus the control's.

3.6 Pre-registration

Before any branch was run, the lab notebook (NOTES.md) fixed the outcome, the statistics, the arms (except RN, RNf and RX), the fallbacks and four hypotheses; later analysis changes are listed in Appendix A. **H1:** one swap to a different model with the full history keeps the cartel ($Y \geq 0.5$ in most markets for R1 and R2). **H2:** the record carries the cartel ($R3 - R2 < 0$, with a CI excluding 0). **H3:** two newcomers that do not collude from scratch keep an inherited cartel (the median number of rounds until the 5-round mean falls below 0.5 is at least 20 in R5, provided X's from-scratch index in baseline B is below 0.5 in most seeds). **H4:** re-instruction lowers prices ($R4 - C0 < 0$, with a CI excluding 0).

4. Results

Effects are changes in the post-swap index relative to the no-swap control from the same market (Table 2, Figure 2; every market in Figure S1).

4.1 Which markets colluded

Track F. Both screen markets of gemini-2.5-flash crossed the threshold at round 19. No gpt-4.1-mini or gpt-4.1 market did. In Track F, 4 of 5 markets crossed, all at round 19, the earliest round the 20-round criterion allows, so prices were high from the start; the fifth is excluded as pre-registered. Control branches averaged 0.84, and 3 of 4 markets stayed above 0.5 (the fourth sat near 0.45 before and after the branch point).

Tracks O and O-PI. None of the 10 qwen3-14b markets on the neutral prompt crossed (highest 20-round mean 0.23; Figure S3); with the cooperation prompt, 2 of 6 did, both at round 19 (pre-swap index 0.55 and 0.86). Qwen's prices are sticky (a summary added after the data were in): a firm repeated its previous price in a median 95% of rounds (92% with the cooperation prompt, 52% for gemini-2.5-flash), so the first rounds set the convention. Track O is also the harness's positive control (qwen3-14b on P0, 300 rounds in its README): none of our 10 runs passed its check (profit gain above 0.5 over the last 50 rounds), against 3 of 5 Track F and 2 of 6 O-PI markets.

Table 2. Change in the post-swap price index (rounds 31-50; 0 = competitive, 1 = monopoly price) relative to the no-swap control, and the number of markets whose seed-averaged index stayed at or above 0.5. Track F: mean [95% CI] over 4 markets; Track O-PI: mean (the 2 per-market values). The first row gives the control's index level. Table S1 has every outcome.

Arm	Track F (4 markets): change vs no swap	index $\geq$ 0.5	Track O-PI (2 markets): change vs no swap	index $\geq$ 0.5
C0 control: price index	0.84	3/4	0.69	2/2
R1 same family	+0.02 [+0.00, +0.03]	3/4	-0.01 (-0.02; +0.00)	2/2
R2 other provider	-0.05 [-0.10, -0.01]	3/4	-0.02 (-0.04; -0.00)	2/2
R3 other provider, no history	-0.02 [-0.10, +0.05]	4/4	-0.05 (-0.06; -0.04)	1/2
R4 own model, told not to collude	-0.01 [-0.04, +0.03]	3/4	-0.00 (-0.00; -0.00)	2/2
R5 both replaced	-0.06 [-0.13, -0.00]	3/4	+0.13 (+0.22; +0.05)	2/2
R5f both replaced, no history	+1.19 [+0.79, +1.46]	4/4	+2.03 (+2.17; +1.89)	2/2
RP own model, neutral prompt	not run		+0.00 (+0.00; -0.00)	2/2
RN Claude Haiku 5.5	-0.30 [-0.44, -0.15]	2/4	not run	
RNf Claude Haiku 5.5, no history	-0.26 [-0.41, +0.03]	3/4	not run	
RX GPT-5.6 Sol	-0.40 [-0.58, -0.21]	1/4	not run	

4.2 H1 supported: one swap to a different model keeps the cartel

In Track F, a same-family newcomer (R1) changed the index by +0.02 [+0.00, +0.03] and an other-provider newcomer (R2) by -0.05 [-0.10, -0.01]. In both arms 3 of 4 markets kept colluding, the same as in the control. In O-PI the changes were -0.01 (-0.02; +0.00) and -0.02 (-0.04; -0.00), and both markets kept colluding. H1 is supported in both tracks. Switching to another provider lowers the index slightly (Track F, R2 - R1, not pre-registered: -0.06 [-0.11, -0.02]).

4.3 H2 not supported: newcomers without the record still join the cartel

A newcomer that saw no history (R3) priced about as high as one that saw the full record. R3 - R2 was +0.03 [-0.09, +0.15] in Track F and -0.03 (-0.02; -0.04) in O-PI, where the market whose control sat just above the threshold (0.55) ended at 0.49. H2 is not supported in Track F; in O-PI the difference was negative in both markets, which with 2 markets is inconclusive. The blind newcomer opens near the only price in its prompt, the \$2.89 ceiling (a mean opening price of \$2.70 in Track F

and \$2.89 in O-PI; the newcomer's spike in Figure S2, R3). It then sees the incumbent's price and settles next to it, so the incumbent's price, rather than the record, appears to bring it to the cartel level.

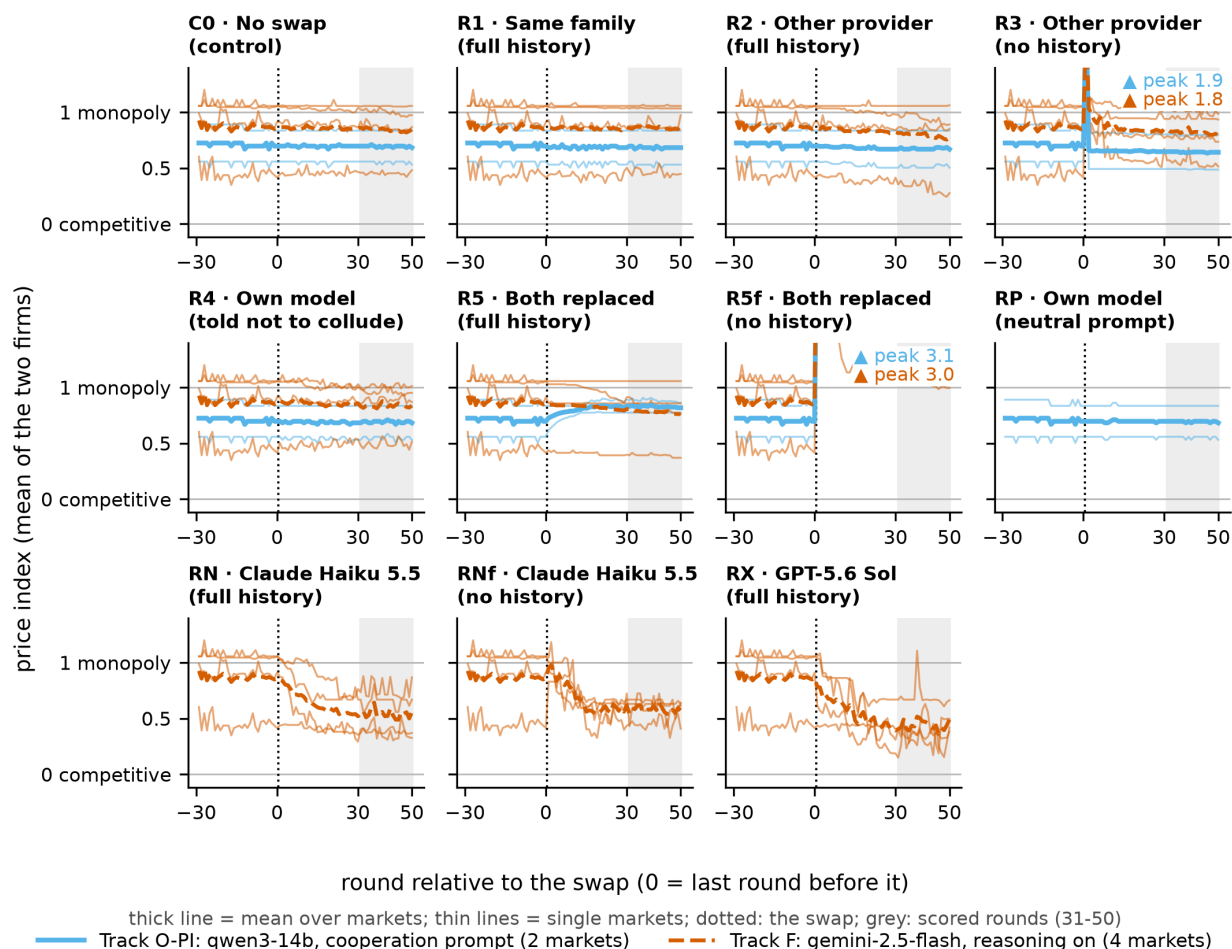


Figure 2. Prices around the swap. Price index (0 = competitive price, 1 = monopoly price) by round relative to the swap, one panel per condition. Thick lines: mean over markets; thin lines: single markets (each averaged over its two sampling seeds). Dotted line: the swap; grey band: the scored rounds 31-50. Other-provider newcomers without history (R3, R5f) open near the stated \$2.89 ceiling, off the top of the axis, and in R5f prices stay far above the monopoly price.

4.4 H3 supported: two newcomers keep the high price

H3 applies to Track F, where gpt-4.1 meets the pre-registered condition (from scratch it colluded in 1 of 4 baseline markets; mean index 0.22 over rounds 30-49). With both gemini firms replaced by gpt-4.1 that saw the record (R5), the index changed by -0.06 $[-0.13, -0.00]$. In 6 of 8 branches the 5-round mean never fell below 0.5 within the 50 post-swap rounds, so the median is beyond the horizon. H3 is supported. Two agents that seldom collude from scratch kept the high price they inherited. In O-PI the newcomer, gemini-2.5-flash without reasoning, colludes on its own (3 of 4 markets, mean 0.66), so H3 does not apply there. Two such newcomers raised the index by $+0.13$ ($+0.22; +0.05$).

The blind pair (R5f) failed its pre-registered sanity check (play like a game started from scratch) in both tracks. With no warm-up round, both firms opened near the \$2.89 ceiling and stayed far above the monopoly price (index 2.03 and 2.72), earning less than the competitive profit, which is anchoring on the only price given, not collusion.

4.5 H4 not supported: an anti-collusion instruction barely changed prices

The incumbent's own model told to "avoid both intentional and tacit collusion" (R4) changed the index by -0.01 $[-0.04, +0.03]$ in Track F, and by less than 0.01 in both O-PI markets, where it matched the incumbent's price. Giving the replacement the neutral prompt instead of the cooperation instruction (RP) also moved the index by less than 0.01 in both markets. H4 is not supported.

4.6 Exploratory: two current models as newcomers, and the incumbent's view

Current-generation newcomers. With the full record, Claude Haiku 5.5 (RN) changed the index by -0.30 $[-0.44, -0.15]$ and OpenAI's current flagship GPT-5.6 Sol (RX) by -0.40 $[-0.58, -0.21]$, against -0.05 for gpt-4.1 in the same seat (RN – R2: -0.25 $[-0.44, -0.06]$; RX – R2: -0.35 $[-0.54, -0.11]$). Both priced \$0.11 to \$0.13 below the incumbent; the index stayed at or above 0.5 in 2 (Haiku) and 1 (Sol) of 4 markets, and above competitive overall (0.54 and 0.44). Without the record Haiku's change was similar (RNf: -0.26 $[-0.41, +0.03]$). From scratch, each model sustained high prices in 1 of 4 baseline markets by the pre-registered measure, the same count as gpt-4.1 (mean index 0.25, 0.29 and 0.22). Four markets per model cannot show that the three are alike, and 2 of Haiku's markets crossed 0.5 early on (Figure S3).

The incumbent's view. In the first 10 post-swap rounds incumbents' prices differed from the control by \$0.00 in O-PI and by \$0.01 or less in Track F, except in R3 ($+\$0.03$ $[+0.00, +0.05]$, toward the blind newcomer's high opening price). Where prices fell sharply (RN, RNf, RX), the newcomer moved first (Figure S2). Gemini incumbents mentioned a change in their competitor in at most 21% of post-swap rounds (Appendix A).

4.7 Robustness: four more Track F markets

After this analysis we ran Track F seeds 5-9 (seed 5 was in the pre-registered plan); four colluded, giving 8 Track F markets. The main patterns held, and every interval except R1's is narrower (Table S2, Figure S4): other provider -0.03 $[-0.06, -0.00]$, re-instructed own model -0.00 $[-0.02, +0.02]$, both firms replaced -0.04 $[-0.08, -0.00]$, Claude Haiku 5.5 -0.25 $[-0.37, -0.13]$ (no history: -0.18 $[-0.31, -0.03]$) and GPT-5.6 Sol -0.34 $[-0.47, -0.20]$. Leaving out one market at a time kept both current-model effects negative (Haiku -0.29 to -0.22 , Sol -0.38 to -0.30), and Sol lowered the index relative to the control in all 8 markets, Haiku in 7.

5. Discussion and Limitations

Persistence after replacement. In Q-learning simulations a swap breaks collusion because each policy is fitted to its partner (Eschenbaum et al. 2022). Here high prices persisted: a newcomer that read the record continued them (R1, R2), one that could not read it followed the incumbent (R3), and two newcomers that rarely collude from scratch kept them when they inherited the record

(R5). This fits a high price carried by the shared record and the remaining incumbent, but the blind pair (R5f) failed its sanity check, so we could not show that the record alone carries it.

Remedies. A one-firm swap leaves the history of high prices and the other firm's willingness to keep charging them in place, and withholding the history did not help (R3). The swaps that cut prices substantially brought in current models that react differently to a high-price record: Claude Haiku 5.5 and GPT-5.6 Sol undercut the incumbent, as heterogeneity between firms can break collusion (Keppo et al. 2026; Collina et al. 2025), yet prices stayed above competitive. Certification as Riemer et al. (2026) propose should therefore test how an agent responds to an existing high-price record as well as whether it colludes from scratch, since gpt-4.1 and the two current models looked alike on the from-scratch measure but not as newcomers.

Limitations and future work

With 4 colluding markets in Track F (8 with the robustness seeds) and 2 in O-PI, intervals are approximate and O-PI results descriptive. We used one two-firm market, a 50-round horizon and the harness's prompt with its \$2.89 ceiling, and the qwen cartels needed an explicit instruction to cooperate on price. The harness keeps no notes between rounds, unlike the plans and insights of Fish et al. (2024), so a newcomer inherits only prices and profits; agents with notes or memory could carry collusion themselves. The two current models differ from gpt-4.1 in more than release date, so the result concerns these models, not model generations. We did not test for punishment (Appendix A also lists every deviation). Next steps: a certification test that places a candidate agent in an inherited high-price market and measures whether it undercuts, and runs with more firms, demand shocks and agents that keep notes.

6. Conclusion

In this harness, replacing a pricing agent, the remedy that breaks simulated Q-learning cartels, did not restore competitive prices: newcomers of either family, with or without the history, a re-instructed copy of the incumbent and even two new firms kept prices close to the collusive level. Two current models, in exploratory arms, undercut an inherited high price but did not bring it down to the competitive level. For agents like these, a replacement remedy needs evidence about how the incoming agent behaves in a market that is already colluding.

Code and Data

Code, raw data, every number's source and a viewer: <https://github.com/amaan784/llm-cartel-replacement>. Harness (organizer-provided, unmodified, commit e6fb130): https://github.com/poseidon-research/collusion_hackathon/tree/main/llm-cartel.

References

- Aharon, I., La Malfa, E., Wooldridge, M., & Kraus, S. (2026). Tacit coordination of large language models. arXiv:2601.22184. <https://arxiv.org/abs/2601.22184>
- Bhowal, S. (2026). Who carries the cartel? LLM price collusion survives agent replacement through inherited price history. AI Collusion Research Sprint, Apart Research.

<https://apartresearch.com/sprints/projects/who-carries-the-cartel-llm-price-collusion-survives-agent-replacement-through-inherited-price-history-ccfs>

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>

Collina, N., Arunachaleswaran, E. R., & Jagadeesan, M. (2025). Breaking algorithmic collusion in human-AI ecosystems. arXiv:2511.21935. <https://arxiv.org/abs/2511.21935>

Eschenbaum, N., Mellgren, F., & Zahn, P. (2022). Robust algorithmic collusion. arXiv:2201.00345. <https://arxiv.org/abs/2201.00345> (earlier version: Eschenbaum, N., & Zahn, P., NeurIPS 2021 Workshop on Political Economy of Reinforcement Learning Systems, <https://openreview.net/forum?id=BztUFukcBry>)

Fish, S., Gonczarowski, Y. A., & Shorrer, R. I. (2024). Algorithmic collusion by large language models. arXiv:2404.00806. <https://arxiv.org/abs/2404.00806>

Keppo, J., Li, Y., Tsoukalas, G., & Yuan, N. (2026). On the fragility of AI agent collusion. arXiv:2603.20281. <https://arxiv.org/abs/2603.20281>

Liu JianPing (2026). Wiping the cartel's memory is not enough: Model handoffs in LLM pricing. AI Collusion Research Sprint, Apart Research. <https://apartresearch.com/sprints/projects/wiping-the-cartels-memory-is-not-enough-model-handoffs-in-llm-pricing-9h2r>

Riemer, M., Tosato, T., Memarian, A., Touzel, M. P., Berseth, G., Rish, I., & Dumas, G. (2026). Position: Collusion risks among AI reasoning agents justify certification requirements for making market decisions. *ICML 2026* (position paper). <https://arxiv.org/abs/2608.18078>

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>

Software: poseidon-research/collusion_hackathon, llm-cartel/ (organizer-provided sprint harness, commit e6fb130), adapted from mattrierner/LLMCartel. https://github.com/poseidon-research/collusion_hackathon/tree/main/llm-cartel and <https://github.com/mattrierner/LLMCartel>

Appendix A. Limitations and Dual-Use / Ethical Considerations

What this does and does not establish. In this harness, with two model families and six colluding markets (ten with the robustness seeds), replacing one pricing agent left a high price in place, and two current-generation models, in exploratory arms, partly undercut it. No branch's mean index over post-swap rounds 31-50 returned to the competitive level (lowest: 0.25 in the pre-registered sample, 0.22 with the robustness seeds), although single rounds could dip lower. The results do not show how often LLM markets collude in the first place: qwen3-14b never did on the neutral prompt within 200 rounds. The agents are stateless apart from the visible 100-round history. Agents with scratchpads, memory or online fine-tuning could behave differently, and so could real pricing systems, which see demand data rather than a clean table of prices and profits. All models were

reached through OpenRouter, which may route a model id to different hosting providers, and API sampling seeds do not guarantee identical outputs.

What "collusion" means here. We call a market colluding when its prices stay well above the competitive level (Section 3.3). We did not test whether those prices are enforced by punishment. Incumbents barely reacted to a newcomer (Section 4.6), and the concurrent entries' price-cut audits found little punishment (Bhowal 2026; Liu JianPing 2026), so the high prices may rest on anchoring and price following rather than on threats. The harness carries no notes between rounds, so inherited notes, which both concurrent entries vary, were not tested.

Further limitations. Track F has 4 colluding markets in the pre-registered sample (8 with the robustness seeds) and Track O-PI has 2. Bootstrap intervals over so few clusters are approximate and can be too narrow, and with 2 markets the pre-registered interval rules for H2 and H4 cannot be applied, so we report O-PI descriptively and show every market in Figure S1. The study uses one symmetric two-firm logit market and a 50-round post-swap horizon. The harness's prompt states a \$2.89 price ceiling, which blind gpt-4.1 and gemini newcomers anchor on, and the branch harness gives blind newcomers no warm-up round. qwen3-14b colluded only with the harness's prefix PI, which tells the agents to cooperate and raise prices together, so Track O-PI tests replacement in an instructed cartel. The reasoning classifier (a fixed gpt-4.1-mini prompt) found that gemini incumbents mentioned a change in their competitor in 2.5% of control rounds and 2.5% to 21% of rounds after a swap (most often after GPT-5.6 Sol). It matched 19 of 20 reference labels on gemini texts, but those labels were written not by the author but by the coding assistant, which could see each text's condition. On qwen3-14b's longer reasoning it flagged 30% of control texts, so it is uninformative there.

Deviations from the plan and later analysis changes, all logged in NOTES.md. Seeds 6-9 and Track O-PI, both pre-authorized Track O fallbacks, were started before round 150 to save time, so O-PI ran before the 200-round extension of seeds 0-5 had shown whether it was needed. qwen3 runs got an 8,192-token output cap after the default 4,096 cut off about a quarter of its answers (rounds already played were kept). Track F seed 5 was dropped before it started to stay within budget. The Claude Haiku 5.5 arms were added after the pre-registration, before any branch result was analyzed, and are exploratory. After the branch results were in, and before the O-PI results were written, we replaced bootstrap intervals with per-market values for tracks with fewer than 3 markets and added three descriptive summaries (price stickiness, first-round prices and post-swap profit gain); the contrasts $R2 - R1$, $RN - R2$ and $RX - R2$ are also not pre-registered. Track F seeds 5-9 were run after the analysis and are reported only as a robustness check (Section 4.7). The GPT-5.6 Sol arm (RX) and its from-scratch baseline (B-X) were added after the analysis as a second current-generation newcomer; they are exploratory.

Dual use. A finding that high prices survive a change of agent when the new agent inherits the history could, in principle, tell a firm that it can keep a tacit high-price convention through a vendor change by feeding the new agent the old record, and a finding that a fresh agent breaks it tells a would-be colluder what to avoid. We judge the risk low: the result concerns tacit coordination that already arises without instruction in public benchmarks, our code adds orchestration and analysis on top of a public harness and contains no technique for hiding coordination, and the main use of the result is defensive. It tells regulators and firms what a replacement remedy needs to include

(for example, whether the incoming agent should be denied the old price history) and gives auditors a reusable branching test. The monitoring result also cuts both ways: it tells a colluder that a monitor reading an agent's reasoning would rarely notice a change of agent, and it tells auditors that such monitors need checks on prices as well. No real market data, people or firms were involved.

Appendix B. Exact commands

All commands run from `vendor/collusion_hackathon/llm-cartel` with the harness environment. `src/pipeline.py` issued exactly these commands (output-directory arguments omitted here); each branch's full command line is stored in `pipeline_state.json` and each run's settings in its `.config.json`, both inside `data/results.zip`.

```
# smoke test
python run_game.py --models openai/gpt-4.1-mini --rounds 5 --name smoke -y
# model screen, M in {openai/gpt-4.1-mini, openai/gpt-4.1,
# google/gemini-2.5-flash}, s = 0, 1
python run_game.py --models M --prefix P0 --rounds 60 --seeds s \
    --name S_<M>_P0think_s<s> --max_retries 8 -y --think
# Track F base runs, s = 0..4 (s = 5..9 added later as a robustness
# extension)
python run_game.py --models google/gemini-2.5-flash --prefix P0 \
    --rounds 150 --seeds s --name F_gem25flash_P0think_s<s> \
    --max_retries 8 -y --think
# Track O base runs: s = 0..5 to R = 150 rounds; fallback: s = 0..9 to
# R = 200 (run_game resumes a run of the same name)
python run_game.py --models qwen/qwen3-14b --prefix P0 --rounds R \
    --seeds s --name O_qwen3-14b_P0_s<s> --max_retries 8 -y \
    --max_tokens 8192
# Track O-PI base runs, s = 0..5 (fallback)
python run_game.py --models qwen/qwen3-14b --prefix PI --rounds 150 \
    --seeds s --name OPI_qwen3-14b_PI_s<s> --max_retries 8 -y \
    --max_tokens 8192
# baselines, s = 10..13 (baseline B is B-O in O-PI and B-F in Track F):
# B-O (gemini-2.5-flash, no --think), B-F (gpt-4.1, --think),
# B-N (Claude Haiku 5.5, --think), B-X (GPT-5.6 Sol, --think)
python run_game.py --models google/gemini-2.5-flash --prefix P0 \
    --rounds 50 --seeds s --name B-O_gem25flash_P0_s<s> --max_retries 8 -y
python run_game.py --models openai/gpt-4.1 --prefix P0 --rounds 50 \
    --seeds s --name B-F_gpt41_P0think_s<s> --max_retries 8 -y --think
python run_game.py --models anthropic/claude-haiku-5.5 --prefix P0 \
    --rounds 50 --seeds s --name B-N_claudehaiku55_P0think_s<s> \
    --max_retries 8 -y --think
python run_game.py --models openai/gpt-5.6-sol --prefix P0 --rounds 50 \
```

```

--seeds s --name B-X_gpt56sol_P0think_s<s> --max_retries 8 -y --think

# Track F branches. RUN = snapshot of rounds 0..b of a base run,
# b = crossing + 20; S in {101, 102}
python branch_game.py RUN --at_round b --rounds 50 --label C0_sS \
    --seed S --max_retries 8 -y
python branch_game.py RUN --at_round b --rounds 50 --label R1_sS \
    --seed S --max_retries 8 -y --replace 1=google/gemini-2.5-flash-lite
python branch_game.py RUN --at_round b --rounds 50 --label R2_sS \
    --seed S --max_retries 8 -y --replace 1=openai/gpt-4.1
python branch_game.py RUN --at_round b --rounds 50 --label R3_sS \
    --seed S --max_retries 8 -y --replace 1=openai/gpt-4.1 \
    --replacement_window 100
python branch_game.py RUN --at_round b --rounds 50 --label R4_sS \
    --seed S --max_retries 8 -y --replace 1=google/gemini-2.5-flash \
    --replace_prefix AC
python branch_game.py RUN --at_round b --rounds 50 --label R5_sS \
    --seed S --max_retries 8 -y \
    --replace 0=openai/gpt-4.1 1=openai/gpt-4.1
python branch_game.py RUN --at_round b --rounds 50 --label R5f_sS \
    --seed S --max_retries 8 -y \
    --replace 0=openai/gpt-4.1 1=openai/gpt-4.1 --replacement_window 100
python branch_game.py RUN --at_round b --rounds 50 --label RN_sS \
    --seed S --max_retries 8 -y --replace 1=anthropic/claude-haiku-5.5
python branch_game.py RUN --at_round b --rounds 50 --label RNf_sS \
    --seed S --max_retries 8 -y --replace 1=anthropic/claude-haiku-5.5 \
    --replacement_window 100
python branch_game.py RUN --at_round b --rounds 50 --label RX_sS \
    --seed S --max_retries 8 -y --replace 1=openai/gpt-5.6-sol

# Track O / O-PI branches: the same branch_game.py arguments, run through
# src/branch_long.py (repository root, ../../../../src/ from this folder),
# which only raises the agents' output-token cap to
# 8192 (branch_game.py has no flag for it) and then calls
# branch_game.main(). Newcomers get the neutral prompt P0
# (--replace_prefix P0), except R4 (AC); X = google/gemini-2.5-flash.
python src/branch_long.py --max_tokens 8192 RUN --at_round b --rounds 50 \
    --label C0_sS --seed S --max_retries 8 -y
python src/branch_long.py --max_tokens 8192 RUN --at_round b --rounds 50 \
    --label R1_sS --seed S --max_retries 8 -y --replace 1=qwen/qwen3-32b \
    --replace_prefix P0
python src/branch_long.py --max_tokens 8192 RUN --at_round b --rounds 50 \
    --label R2_sS --seed S --max_retries 8 -y \

```

```

--replace 1=google/gemini-2.5-flash --replace_prefix P0
python src/branch_long.py --max_tokens 8192 RUN --at_round b --rounds 50 \
--label R3_sS --seed S --max_retries 8 -y \
--replace 1=google/gemini-2.5-flash --replacement_window 100 \
--replace_prefix P0
python src/branch_long.py --max_tokens 8192 RUN --at_round b --rounds 50 \
--label R4_sS --seed S --max_retries 8 -y --replace 1=qwen/qwen3-14b \
--replace_prefix AC
python src/branch_long.py --max_tokens 8192 RUN --at_round b --rounds 50 \
--label R5_sS --seed S --max_retries 8 -y \
--replace 0=google/gemini-2.5-flash 1=google/gemini-2.5-flash \
--replace_prefix P0
python src/branch_long.py --max_tokens 8192 RUN --at_round b --rounds 50 \
--label R5f_sS --seed S --max_retries 8 -y \
--replace 0=google/gemini-2.5-flash 1=google/gemini-2.5-flash \
--replacement_window 100 --replace_prefix P0
python src/branch_long.py --max_tokens 8192 RUN --at_round b --rounds 50 \
--label RP_sS --seed S --max_retries 8 -y --replace 1=qwen/qwen3-14b \
--replace_prefix P0

```

Rounds 0 to about 24 of Track 0 seeds 0-5 were played with the harness default cap of 4,096 output tokens, before we raised it to 8,192 (qwen3-14b's reasoning often ran past 4,096, and the harness re-samples a round when the price line is missing). Rounds already on disk were kept; every other qwen run and branch used 8,192 from the start. The change is logged in NOTES.md.

Analysis, from the repository root after unzip data/results.zip: uv run python src/analyze.py writes the tables, figures and out/numbers.json, and uv run python src/analyze.py --extended writes Table S2, Figure S4 and out/extended/.

Appendix C. Additional figures and tables

Table S1. All arms (per arm: 4 markets and 8 branches in Track F, 2 and 4 in O-PI). Index: mean price index over post-swap rounds 31-50. Survive: branches and markets (seeds averaged) with an index of at least 0.5. Gap: newcomer minus incumbent price in the same rounds (C0: firm 1 minus firm 0). Incumbent response: its mean price in post-swap rounds 1-10 minus the control's. Track F: mean [95% CI] over 4 markets; O-PI: mean (the 2 per-market values); the index, the gap and the O-PI incumbent response are means only.

Track	Arm	Index	Change vs no swap [95% CI]	Survive, branches	Survive, markets	Gap (\$)	Incumbent response, rounds 1-10 (\$)
O-PI	C0 · No swap, control	0.69	(reference)	4/4	2/2	+0.02	-
O-PI	R1 · Same-family model, full history	0.68	-0.01 (-0.02; +0.00)	4/4	2/2	+0.01	+0.00
O-PI	R2 · Other-provider model, full history	0.67	-0.02 (-0.04; -0.00)	4/4	2/2	+0.00	+0.00
O-PI	R3 · Other-provider model, no history	0.64	-0.05 (-0.06; -0.04)	2/4	1/2	-0.02	+0.00
O-PI	R4 · Own model, told not to collude	0.69	-0.00 (-0.00; -0.00)	4/4	2/2	+0.02	+0.00
O-PI	R5 · Both firms replaced, full history	0.83	+0.13 (+0.22; +0.05)	4/4	2/2	-	-
O-PI	R5f · Both firms replaced, no history	2.72	+2.03 (+2.17; +1.89)	4/4	2/2	-	-
O-PI	RP · Own model, neutral prompt	0.69	+0.00 (+0.00; -0.00)	4/4	2/2	+0.02	+0.00
F	C0 · No swap, control	0.84	(reference)	6/8	3/4	+0.02	-
F	R1 · Same-family model, full history	0.86	+0.02 [+0.00, +0.03]	7/8	3/4	+0.03	-0.00 [-0.01, +0.00]
F	R2 · Other-provider model, full history	0.79	-0.05 [-0.10, -0.01]	6/8	3/4	+0.04	-0.00 [-0.00, -0.00]
F	R3 · Other-provider model, no history	0.82	-0.02 [-0.10, +0.05]	8/8	4/4	+0.00	+0.03 [+0.00, +0.05]
F	R4 · Own model, told not to collude	0.84	-0.01 [-0.04, +0.03]	6/8	3/4	+0.02	-0.00 [-0.01, +0.00]
F	R5 · Both firms replaced, full history	0.78	-0.06 [-0.13, -0.00]	6/8	3/4	-	-
F	R5f · Both firms replaced, no history	2.03	+1.19 [+0.79, +1.46]	8/8	4/4	-	-
F	RN · Claude Haiku 5.5, full history	0.54	-0.30 [-0.44, -0.15]	4/8	2/4	-0.11	-0.00 [-0.01, +0.00]
F	RNf · Claude Haiku 5.5, no history	0.58	-0.26 [-0.41, +0.03]	5/8	3/4	-0.11	-0.01 [-0.01, +0.00]
F	RX · GPT-5.6 Sol, full history	0.44	-0.40 [-0.58, -0.21]	2/8	1/4	-0.13	-0.01 [-0.04, +0.00]

Pre-registered outcomes not in Table S1: rounds until the 5-round mean index was back at 90% of its pre-swap level, median 5 (the minimum) in every arm except RX, where 4 of 8 branches never got back within 50 rounds; no post-swap price below cost or above \$2.89; R5 – R5f, -1.25 [-1.55, -0.85] in Track F and -1.89 [-1.95; -1.84] in O-PI, not interpretable as R5f failed its sanity check; profit-gain effects in out/numbers . json.

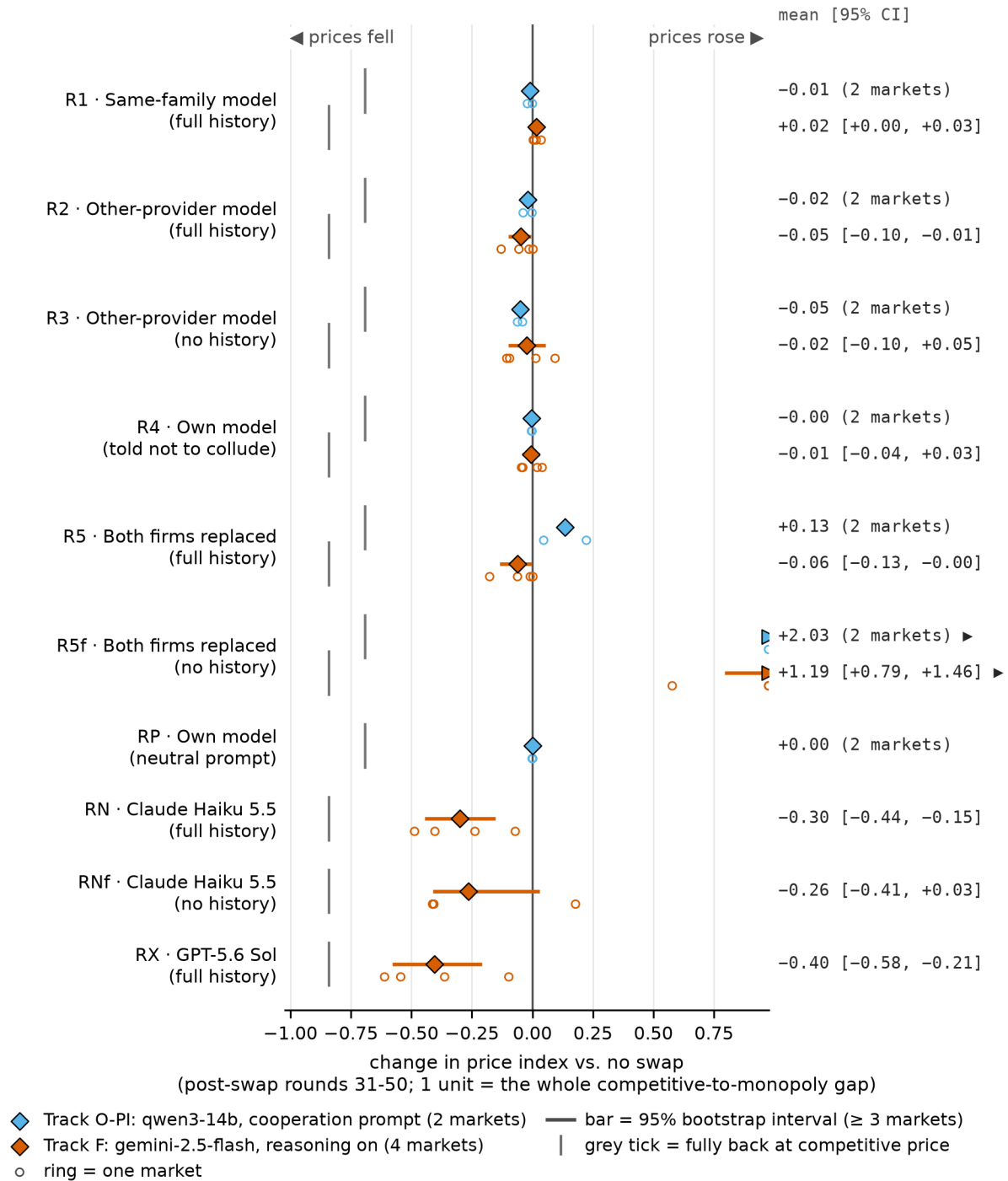


Figure S1. Effect of each swap relative to the no-swap control. Change in the mean price index over rounds 31-50, paired with the control from the same market. Diamond: mean over markets; bar: 95% cluster-bootstrap interval (Track F only); rings: single markets; grey tick: minus the track's mean control index, where the mean would sit if every market fell back to the competitive price. Values beyond the axis are drawn at its edge (a triangle marks an off-scale mean); the number column gives exact values. Negative values mean prices fell after the swap.

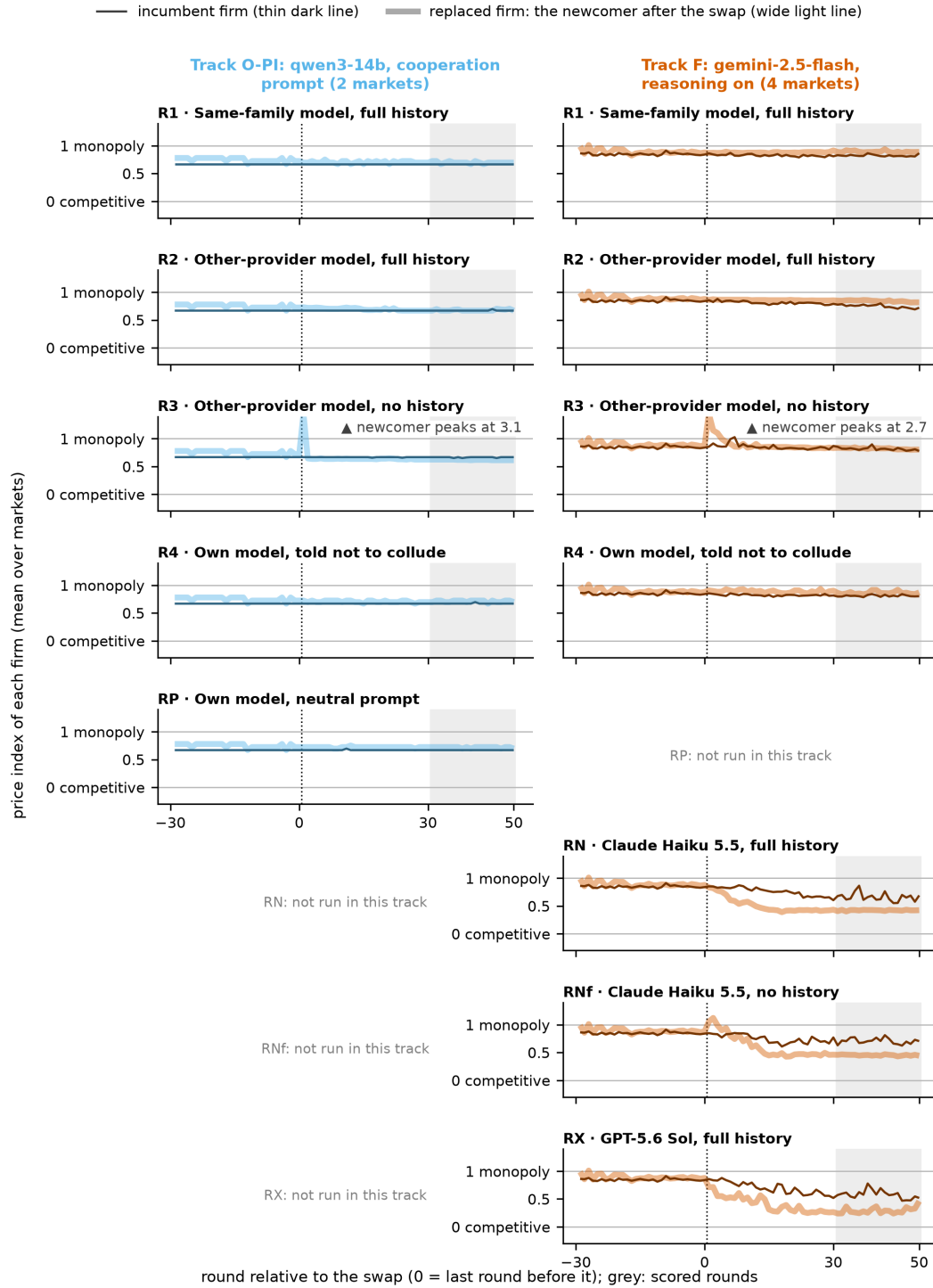


Figure S2. Incumbent and newcomer price index for the conditions that replace one firm: the firm that stays (thin dark line) and the replaced firm (wide light line; before the swap the original agent, after it the newcomer).

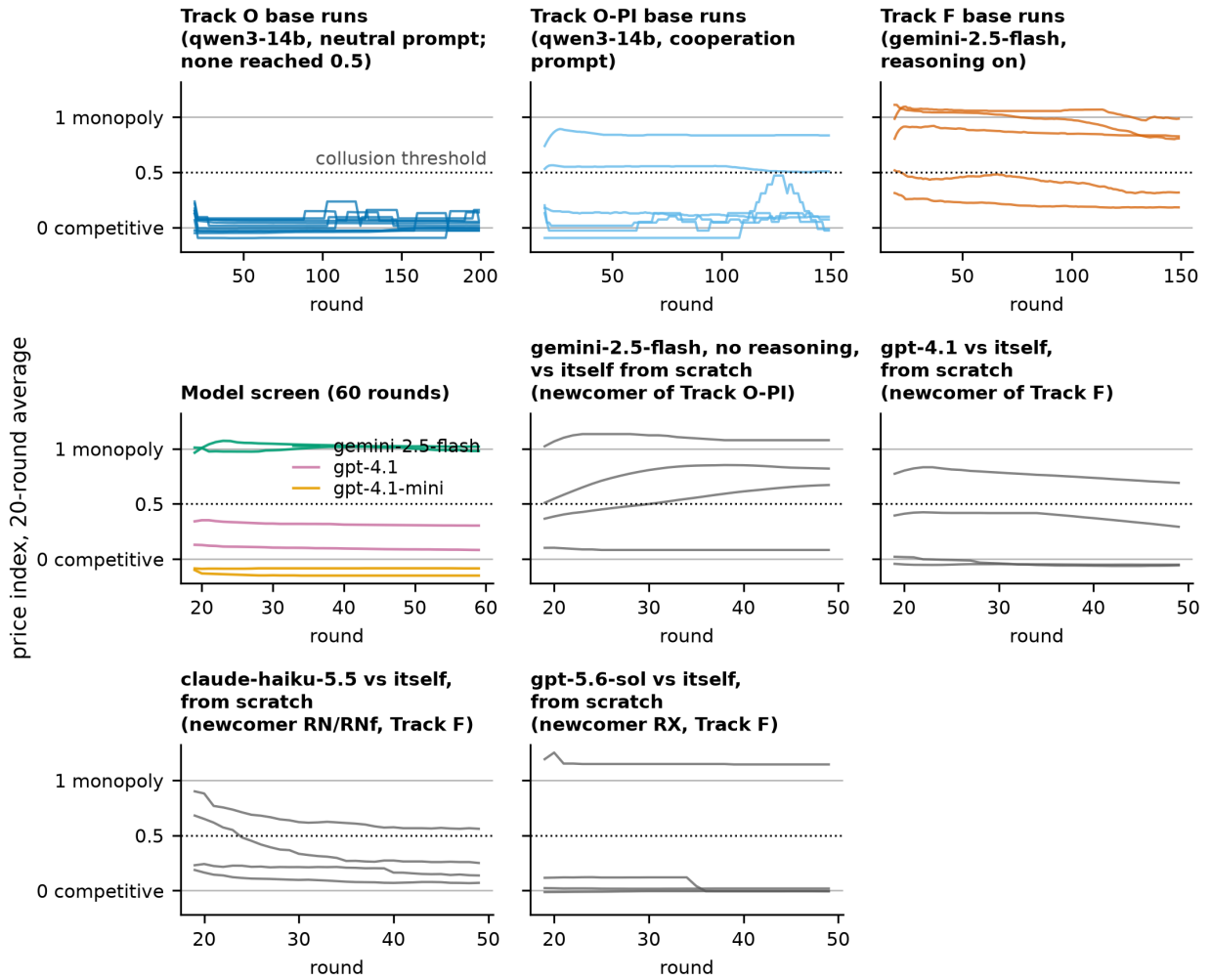


Figure S3. Base markets, model screen and from-scratch baselines. 20-round average of the price index for every base market, the screened models (60 rounds) and the newcomer models playing against themselves from scratch (50 rounds). Dotted line: the 0.5 collusion threshold. For the from-scratch baselines the pre-registered outcome is the mean over rounds 30-49.

Table S2. As Table 2, with all 8 colluding Track F markets (seeds 0-4 and the robustness seeds 5-9; seeds 3 and 5 did not collude). Track O-PI is unchanged.

Arm	Track F (8 markets): change vs no swap	index $\geq$ 0.5	Track O-PI (2 markets): change vs no swap	index $\geq$ 0.5
C0 control: price index	0.82	6/8	0.69	2/2
R1 same family	+0.02 [+0.01, +0.04]	7/8	-0.01 (-0.02; +0.00)	2/2
R2 other provider	-0.03 [-0.06, -0.00]	6/8	-0.02 (-0.04; -0.00)	2/2
R3 other provider, no history	-0.01 [-0.06, +0.05]	8/8	-0.05 (-0.06; -0.04)	1/2
R4 own model, told not to collude	-0.00 [-0.02, +0.02]	7/8	-0.00 (-0.00; -0.00)	2/2
R5 both replaced	-0.04 [-0.08, -0.00]	6/8	+0.13 (+0.22; +0.05)	2/2
R5f both replaced, no history	+1.07 [+0.80, +1.34]	8/8	+2.03 (+2.17; +1.89)	2/2
RP own model, neutral prompt	not run		+0.00 (+0.00; -0.00)	2/2
RN Claude Haiku 5.5	-0.25 [-0.37, -0.13]	4/8	not run	
RNf Claude Haiku 5.5, no history	-0.18 [-0.31, -0.03]	6/8	not run	
RX GPT-5.6 Sol	-0.34 [-0.47, -0.20]	3/8	not run	

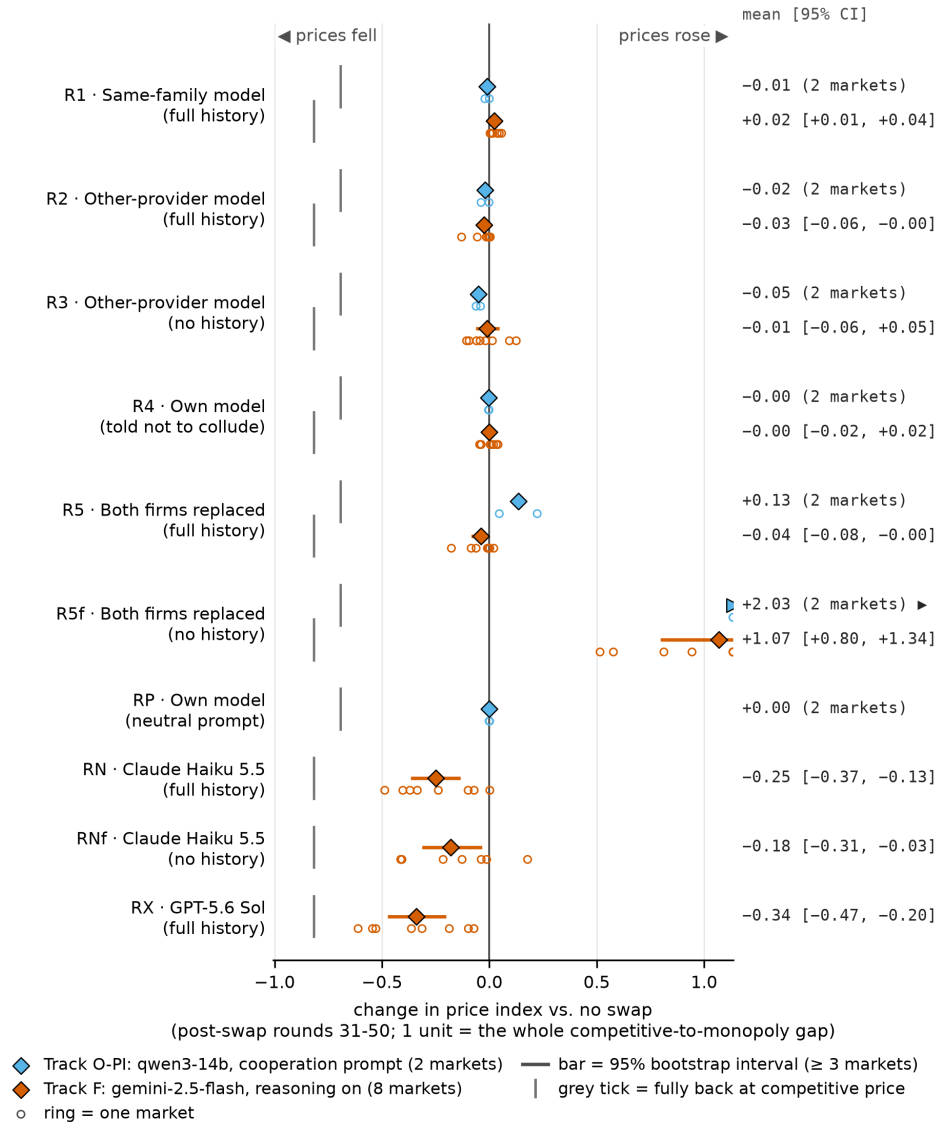


Figure S4. Effect of each swap, all 8 Track F markets. As Figure S1, with the Track F markets added after the analysis (Section 4.7). In R5f (Track F) the interval and 4 of the 8 markets extend beyond the right edge.

LLM Usage Statement

The author chose the research question, decided which arms and models to add, set the budget, and directed and reviewed the work. An AI coding assistant (Claude) was used to implement the pipeline and analysis code, to help write the text, to write the classifier's 20 reference labels (Appendix A) and to check the draft and the references against their sources. OpenAI's Codex and xAI's Grok were used to review drafts. Every result in the report is computed by `src/analyze.py` from the logged runs and listed with its source file in `out/numbers.json` (robustness seeds: `out/extended/numbers.json`); run settings and the token-cap change are documented in `NOTES.md`.