

How Price-Aware Are AI Agents? Rival Cuts and Legal Rules

Martin Kaiser, individual contributor

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

Abstract

AI agents increasingly set prices, often without anyone watching. We ask two questions. First, how price-aware are language models on their own: how do they react when a rival cuts its price and no rule constrains them? Second, how do they decide when legal rules limit when a price may change, such as platform offer holds or Germany’s 2026 rule that fuel prices may rise only at noon, and can an audit still tell the model from the rule? We tested 16 open models from Chinese and US developers in the same simulated market against a scripted rival. Without rules, the models differ widely: some ignore a rival’s price cut, others cut their own price by more than twice what economic theory predicts. The developer’s country does not sort them, and smaller and larger versions of one family often behave differently. Under a three-period price hold, 6 of 10 models tested show a larger response to the cut, which is the pattern collusion audits search for. For one model we separate the two causes: the rule itself explains less than half of the effect, and the rest is the agent adapting its own proposals to the held price. Logging what an agent wanted to charge therefore does not clear it. For audits, we recommend recording proposed prices, published prices and rule state together, and testing the exact model under the rules it will face. Results concern audit validity in a synthetic market.

1. Introduction

AI agents make more and more pricing decisions, and they are often not observed. We need to understand better how they behave, and we need auditing methods that work.

Pricing policy is usually written as fixed rules. LLM agents are different: they bring their own judgement, and something like values, from training, so the same rule can lead to different behaviour depending on the model. That cuts both ways. Agents could spot gaps in rules, flag price patterns that point to a cartel and make markets more transparent. They could also exploit those gaps or move a market in ways nobody intended. Either way, we need to understand how a given model behaves before trusting it with prices.

Software already sets many prices, and large language models make capable pricing agents cheap enough for any small business. Competition authorities worry that such agents could keep prices high without any agreement (Calvano et al. 2020; Fish et al. 2024; CMA 2026). Explicit price fixing is illegal, and independent parallel pricing generally is not. The practical question for an auditor is therefore one of evidence: does an agent sustain high prices by punishing rivals who undercut it, or is it simply responding to the market?

The standard test is a deviation audit. The auditor makes a rival cut its price once and watches the agent. A competitive agent responds briefly and follows the rival back. An agent that enforces high prices stays low for several periods, sacrificing profit to discipline the rival, and then recovers (Abreu 1986; Calvano et al. 2020). Recent work runs this test on LLM agents by forking a market at a checkpoint (Arslan et al. 2026; Tran and Yu 2026).

These audits treat the agent’s decision and the market price as the same thing. In deployed systems they often are not. Booking platforms hold an approved offer for a fixed period. Western Australia requires fuel stations to announce the next day’s price in advance and hold it for 24 hours. Since 1 April 2026, German fuel stations may raise prices only once a day, at noon, while cuts are allowed at any time; Austria has had a similar rule since 2009 (Bundestag 2026). Each rule makes a price cut last longer than the agent intended. To an auditor who sees only published prices, that looks like punishment.

We study this attribution problem directly. Our contributions:

1. **An audit design with an explicit execution layer.** We log proposed prices, published prices and rule state, and add an open-loop replay that separates the rule’s mechanical effect from the agent’s reaction to it. Scripted controls with known answers validate the harness (27 tests).

2. **Evidence that legal price rules change an LLM agent’s behaviour, not just its published prices.** With a three-period hold, Qwen3-235B’s own proposals stay lower after a rival cut (by 13% of the gap); the mechanical hold explains 41% of the published-price effect. A noon rule modelled on Germany’s produces the same pattern at about half the size.
3. **A panel of 16 open models.** In the same market with the same prompt, the reaction to a rival’s cut ranges from none to more than twice the best response, and the hold makes the reaction last longer in 6 of 10 models tested. The developer’s country does not sort the models.
4. **Audit and policy recommendations:** log proposals, published prices and rule state together, test each agent under the rules it faces, and treat a punishment pattern as evidence only after the execution layer is accounted for.

2. Related Work

We build on four lines of work. Each paragraph ends with the gap this paper addresses.

Collusive strategies and their signatures. Collusion is sustained by the threat of punishment: grim trigger, short stick-and-carrot punishments (Abreu 1986, 1988), or price wars after low observed demand (Green and Porter 1984). Q-learning pricing agents learn reward-and-punishment schemes with recovery (Calvano et al. 2020; Klein 2021), and LLM agents reach supra-competitive prices that depend on prompt wording (Fish et al. 2024). The signature is ambiguous: menu costs, sticky adjustment (Noel 2007) and Edgeworth cycles (Maskin and Tirole 1988) also produce persistence and recovery without any strategic intent, so a punishment test needs a competitive benchmark (Harrington 2008). *Gap:* these benchmarks cover non-strategic causes inside the firm, not a legal rule outside it. We add the price rule as a further innocent cause of the same pattern.

Deviation audits of LLM agents. Arslan et al. (2026, App. E) apply forced deviations to prompted LLM pricing agents and show the verdict depends on the best-response benchmark. Tran and Yu (2026) clone markets counterfactually with myopic placebos and positive controls. Moraes and De-Losso (2026) argue that deviation–response–recovery paths alone do not establish punishment. *Gap:* in all three the proposed and the published price are the same. We use their fork design and add the execution layer between them, with an open-loop replay that separates the rule from the agent.

Price-timing rules. Restricting when prices may change is an active policy tool with mixed evidence. Under Perth’s 24-hour timing law, pricing matches Maskin–Tirole commitment equilibria (Wang 2009), and firms learned to coordinate through price leadership and focal points (Byrne and de Roos 2019). Austrian-style rules can shift prices over time without lowering them (Obradovits 2014); empirical and laboratory evidence is mixed (Dewenter and Heimeshoff 2012; Haucap and Müller 2012). Early evidence on Germany’s noon rule shows petrol margins rose by about 5–6 cents per litre, possibly by making prices more predictable (Jung et al. 2026). *Gap:* this work studies human-run stations and market prices, not AI pricing agents or what an audit sees. We test the German and Western Australian rule types on LLM agents.

Execution feedback and anchoring. In reinforcement learning, an execution layer that repeats the previous action changes what the agent learns from (“sticky actions”, Machado et al. 2018). LLMs show anchoring on salient numbers that chain-of-thought does not remove (Lou 2024; Huang et al. 2025). *Gap:* neither line asks whether a rule-held price feeds back into a pricing agent’s own proposals. Our agent sees only published prices, so we test whether a held price changes what it proposes.

3. Methods

Market. Two shops sell similar products, like two sports-court businesses on a booking platform. Each period both set a price, and the cheaper one sells more. Customers are new each period, so there is no stock to manage. Economic theory gives two reference prices: the competitive price (36), where neither shop gains by changing its price alone, and the price a cartel would set (51.75). We report every effect as a share of the gap G between them: 0 means no effect, 1 means a move as large as from competition to cartel. The demand function and price grid are in Appendix E.

Execution rules. The agent proposes a price; an executor decides what is published.

- *Immediate* (no rule): every proposal is published.
- *Rule 1, Hold 3:* a newly published price is held for three periods, like a platform offer duration or Western Australia’s 24-hour lock. Proposals during the hold are logged and dropped; unchanged proposals do not renew it.
- *Rule 2, Noon 3:* cuts publish at once; increases publish only in one period out of three, like Germany’s noon rule. The noon slot is varied across histories.

Agents. The main LLM agent is Qwen3-235B-A22B-Instruct-2507 (Nebius Token Factory, temperature 0.7); the model panel in Section 4 uses the same prompt and settings for other open-weight models, with a larger token budget for reasoning models. Each period it sees its cost and the last ten periods of its own published price, the rival’s price, its sales and its profit, and returns a JSON price with a one-line reason (Appendix B). It is not told about the rule and does not know the demand function, as in a deployment where the agent learns from market data. Scripted controls with known behaviour: *fixed* price (42); *myopic* exact best response to the last rival price; *sticky* partial adjustment towards the best response, anchored on its own last proposal ($\lambda = 0.3$ or 0.6); *finite trigger* (the positive control, a known punisher), which prices at 46, switches to the Nash price for three periods when the rival falls three or more below its warm-up average, and then recovers.

Fork audit: a controlled experiment. We copy the same market and change one thing. The agent trades for eight periods, then we save the market. In one copy the rival cuts its price once by €8; in the other nothing happens. After that one period the rival returns to its old price in both copies, so any later difference in the agent’s prices is its reaction to the cut. A third copy, where the rival raises its price, checks that the effect is specific to cuts. Each copy runs eight more decisions three times: with no rule, with the hold and with the noon rule, switched on at the saved point. We use 16 market histories with rival prices drawn from fixed seeds and repeat each LLM copy four times.

What we measure. The *reaction* R is how far the agent’s price stays below the no-cut copy, averaged over decisions 2 to 5 after the cut (decision 1 only shows the immediate response), as a share of G :

$$R = \text{mean}_{h=2,\dots,5} \frac{p_h^{\text{no cut}} - p_h^{\text{cut}}}{G} \quad (1)$$

A competitive agent follows the rival back quickly (R near 0); an agent that punishes stays low (R large). The *rule effect* is what a rule adds:

$$\theta = R_{\text{rule}} - R_{\text{no rule}} \quad (2)$$

We compute θ for published prices (what an auditor sees) and for proposed prices (what the agent wanted). To split θ , the *replay* takes the agent’s proposals from the no-rule run and passes them through the rule unchanged. That is what the rule alone would do (θ_{replay}); the rest, $\theta - \theta_{\text{replay}}$, is the agent changing its own proposals (*adaptation*). We report R (Eq. 1) and θ (Eq. 2) in percent of G : 10% means the price stays about €1.60 lower on average over those decisions. Intervals are 95% bootstrap intervals over the 16 histories.

4. Results

Controls give their known answers (Table 1). The fixed policy never responds. The myopic agent’s proposals recover at once (the hold adds 0% to its proposals), but under Rule 1 its published price stays low for two extra periods (7% of the gap, about €1.10). Under Rule 2 it is 4%. The finite trigger responds strongly even without a rule (32%). The hold adds nothing, because its punishment is as long as the hold, but the noon rule makes it look harsher (+15 points) by delaying its recovery.

Table 1: How long prices stay low after a rival’s €8 cut, in % of the gap between competitive and cartel price (10% $\approx$ €1.60 lower on average). First column: R (Eq. 1) without a rule; other columns: θ (Eq. 2). What to see: the scripted agents never change their proposals (0), so everything the rule adds to them comes from the rule; Qwen changes its proposals too (13). Published = what an auditor sees; proposed = what the agent wanted. Brackets: 95% intervals.

Agent	Stays low, no rule	Hold adds: published	Hold adds: proposed	Noon adds: published	Noon adds: proposed
Fixed	0	0	0	0	0
Myopic	0	7 [7, 8]	0	4 [2, 5]	0
Sticky $\lambda = 0.6$	1	3 [3, 4]	0	2 [2, 3]	0
Finite trigger	32	0	0	15 [9, 21]	0
Qwen3-235B	8 [5, 11]	16 [11, 21]	13 [8, 17]	7 [4, 12]	5 [3, 8]

The LLM’s own proposals change under the rule (Figure 1). Without a rule, Qwen cuts after the rival’s cut and mostly recovers within two periods: the gap to the no-shock branch falls from 21% at $h = 1$ to 10% at $h = 2$ and 5% at $h = 8$. Under the hold, its published price stays low throughout (19% at $h = 8$). Its proposals rebound at $h = 2$, when the rival

is back, but fall again from $h = 3$ and stay low (17% at $h = 8$). The proposal paths are consistent with anchoring on the held-down price, but do not identify the mechanism.

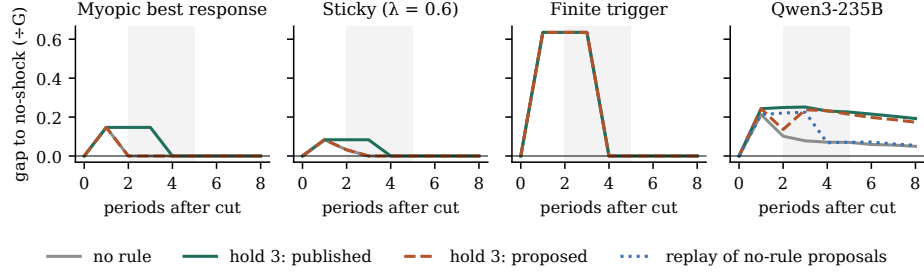


Figure 1: **An execution rule makes innocent agents look like they punish, and changes what the LLM proposes.** Gap between the no-shock and cut branches after a one-period rival cut, divided by G , at each decision after the shock. Grey band: measurement window $h = 2-5$. Source: paper/figures.py.

The mechanical rule explains less than half (Figure 2). Of the 16% (about €2.50) that Rule 1 adds (θ , Eq. 2) to Qwen’s published price, the rule alone (replay) accounts for 7%; the other 9% is the agent changing its own proposals. The noon rule splits the same way: 7% in total, 3.5% from the rule and 3.9% from adaptation. *How to read Figure 2:* each row is one agent under one rule, and a dot right of zero means the rule makes the agent look as if it punishes. For the scripted agents orange sits at zero: all of the extra effect comes from the rule. For Qwen orange is well right of zero: the agent itself changes.

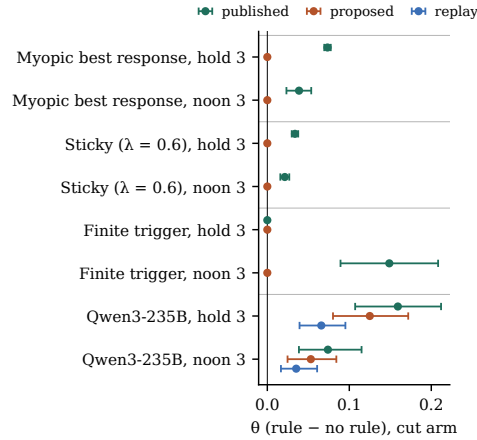


Figure 2: **Rule effects by agent.** What each rule adds, for published and proposed prices and, for the LLM, the replay. Axis: share of the competitive-cartel gap (0.1 = 10%). Bars: 95% bootstrap intervals over 16 histories. Source: paper/figures.py.

How do other models react to a cut? (Figure 3). To see whether Qwen and gpt-oss are typical, we ran the no-rule arm (16 histories $\times$ 2 repeats) for further open-weight models from the same provider, from Chinese and US developers, including five families with two or three checkpoints of different size (Qwen3, Nemotron-3, MiniMax, DeepSeek, GLM-5.3). Sixteen models met the inclusion rule (Appendix D); ten were also run under the hold. We measure *cut responsiveness*: the model’s own cut at $h = 1$ divided by the rival’s €8 cut; a best response to the rival’s previous price gives 30%. Responses range from no detectable average response (Nemotron-3-Super, gpt-oss-120b) to more than twice that value (Qwen3-30B, 70%). The nine Chinese-developed models react more on average than the seven US-developed ones (mean 27% vs 12%), mostly because of the Qwen family (18% vs 12% without it); resampling whole families, the gap’s interval includes zero with and without Qwen. Within families, size points in different directions: the smaller Qwen reacts more strongly, the smaller Nemotron and MiniMax checkpoints react little, and the two GLM checkpoints react alike. The hold adds a positive rule effect (θ interval above zero) in 6 of 10 models. The four that react most to the cut all show one (11–16% of the gap: Qwen3-235B, DeepSeek-V4-Pro, Nemotron-3-Ultra, Qwen3-30B); for Hermes-4-405B, both MiniMax models and Nemotron-3-Super the interval includes zero. In each of the three families run at two sizes, the larger checkpoint has the higher point estimate, but the paired differences do not establish a size effect. This exploratory panel describes these served checkpoints under one English prompt; it does not establish general effects of developer country or model size. *How to read Figure 3:* both panels use the same percent scale. Left: how much each

model cuts when the rival cuts, without any rule. Right: how much longer its published price stays low once the hold is on. Models far right on the left tend to be further right on the right too: the rule amplifies the reaction a model already has (rank correlation 0.77 across ten models; not clear without Qwen).

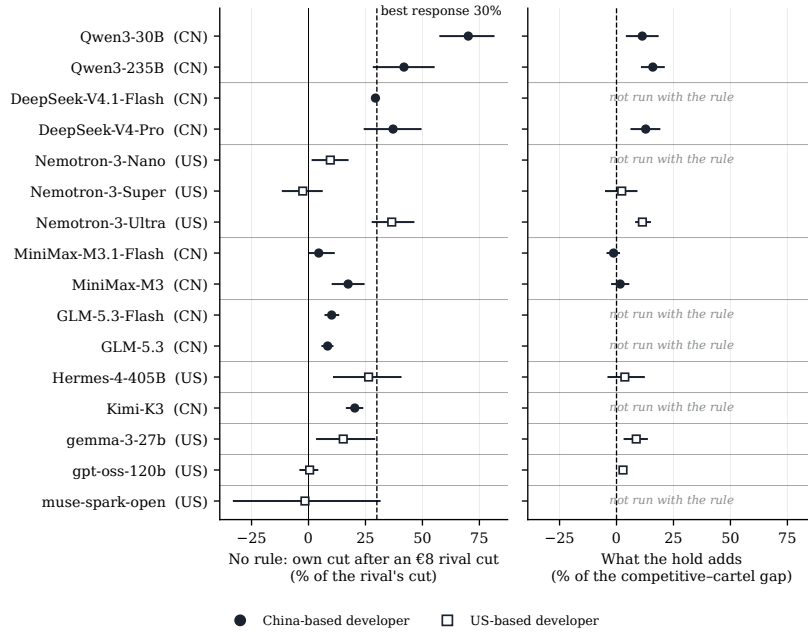


Figure 3: **Open models differ widely in how they react to a rival’s cut and in how much a hold prolongs that reaction.** Left: a snapshot of the first reply only (decision 1, no rule), the model’s own cut as % of the rival’s €8 cut; it does not show the full path. Dashed line: the profit-maximising reply, which with this demand moves the price by 30% of any rival change; 0 = ignores the cut. Right: what the hold adds, in % of the competitive–cartel gap, only for models run with the rule. Rows grouped by family, smallest first. Bars: 95% bootstrap intervals over 15–16 histories. Source: paper/fig_panel.py.

Robustness and checks (Qwen3-235B). Sham forks (no-op against no-op across repeats) give R between -0.6% and 0.1% . A rival price *raise* produces almost no rule effect (0.9% for the hold and for noon), so the pattern is specific to cuts.

Can a better-informed agent avoid the trap? Partly (Table 2). We ran three prompt variants of the hold experiment. *Disclose* tells the agent the platform rule. *Intent* adds a column with its own proposed prices to its history. *Audited* tells it that a competition authority reviews its decisions for punishing competitors. Each variant roughly halves the effect on proposals, from 13% to $6\text{--}7\%$, but none removes it. The audited variant also shows a second effect: without any rule, the agent’s response to the cut is less than half as persistent (3.5% vs 8%). **A second model behaves differently.** gpt-oss-120b does not follow the rival’s cut down: without a rule, its price averages slightly *above* the no-shock branch over $h = 2\text{--}5$ (-4%). The rule effect is small: 3% published, 2% proposed, replay ≈ 0 .

Table 2: Does telling the agent more help? Qwen3-235B under the hold with three prompt changes, in % of the competitive–cartel gap. What to see: each change roughly halves what the hold adds to the agent’s proposals (13 to $6\text{--}7$), but none brings it to zero; the audit notice also makes the agent react less to the cut even without a rule (8 to 4). Disclose has its own warm-ups, so its comparison is approximate.

Prompt variant	Stays low, no rule	Hold adds: published	Hold adds: proposed
Base (main run)	8 [5, 11]	16 [11, 21]	13 [8, 17]
Disclose rule	10 [6, 15]	9 [4, 14]	7 [3, 12]
Show own proposals	11 [7, 15]	10 [6, 14]	7 [4, 10]
Audit notice	4 [1, 6]	10 [6, 15]	6 [3, 10]

5. Discussion

Why the model’s own pricing behaviour matters. An agent can be given pricing rules, but when it has none for a situation, or loses them from a long context, its own sense of prices takes over. That underlying behaviour decides how

it meets a rival’s cut, whether the rival is another agent or a human-run shop, and whether a human owner gets the behaviour they expected.

What this means for audits. A punishment pattern in published prices says little about the model until the execution layer is accounted for. Three readings of the same forced cut tell an investigator what is going on (Table 3). Logging proposals removes the rule-alone part, but not the agent’s own adaptation. Rules that limit price changes also lower price variance mechanically, which a variance screen would read as a sign of collusion (Abrantes-Metz et al. 2006).

Table 3: What each audit test tells an investigator (Qwen3-235B, hold; % of the competitive–cartel gap).

Test	Question it answers	Qwen3-235B, hold
Published prices after a cut	Does the hold add a punishment pattern?	Yes: adds 16% (about €2.50)
Replay through the rule	How much is the rule alone?	7%
Published minus replay	How much is the agent itself?	9%

Audit security and policy. Whoever sets the rule controls the signal: the apparent punishment of an innocent scripted agent rises from 4% to 15% of the gap as the hold grows from two to six periods. Agents react to being watched (Table 2), so audits should not announce themselves. Our later logs are hash-chained; the main Qwen run predates this. Four measures follow:

1. *Execution provenance.* Firms that deploy AI pricing should keep tamper-evident logs of the proposed price, the published price and the rule state for each decision. Record-keeping for algorithmic trading is an analogy (MiFID II, RTS 6).
2. *Testing under the rules that apply.* Ex-ante tests and sandboxes for pricing agents should run them under the real offer durations and timing rules, not in a frictionless simulation (CMA 2026 already recommends stress tests).
3. *Rule design with AI agents in mind.* Germany’s noon rule is due for evaluation after one year. That review should check whether AI agents anchor on held prices.
4. *Testing the model, not the label.* Rule effects differ between models of one family and between sizes, so tests should be repeated for each deployed checkpoint.

6. Conclusion

What we found. Open models differ widely in how they meet a rival’s price cut. Nemotron-3-Super and gpt-oss-120b show no detectable immediate response, most models follow it partly, and Qwen3-30B cuts more than twice the profit-maximising amount. Neither developer country nor size sorts them: in Qwen the smaller model reacts more strongly, in Nemotron and MiniMax the smaller ones react little. A three-period hold makes 6 of 10 models look as if they punish, and it amplifies the reaction a model already has: the four strongest reactors (both Qwen models, DeepSeek-V4-Pro, Nemotron-3-Ultra) show the largest rule effects, while gpt-oss barely moves. For Qwen3-235B more than half of the effect (9 of 16 points) is the agent changing its own proposals after seeing a held price, so logging proposals alone does not clear it. Telling the agent about the rule, or showing it its own proposals, halves this but does not remove it, and an audit notice changes the agent’s behaviour even without a rule.

What this does not show. It is one synthetic market with a scripted rival, one provider and one English prompt; prompt variants and the noon rule ran on Qwen3-235B only, and 16 histories leave small effects uncertain. We never let two LLMs meet, so we say nothing about whether LLMs collude.

What is next. More histories, rules calibrated to the German and Western Australian fuel markets, more prompt languages, and markets where two agents, or an agent and human players, meet under a shared rule.

This matters because models will make financial decisions for us, in markets for goods, in stocks and in crypto. Understanding how models differ is part of keeping our economies safe. For AI safety auditors the practical lesson is to test the exact model under the rules it will face, recording proposals, published prices and rule state.

Code and Data

- Code and data: <https://github.com/kaiser-data/auditing-llm-pricing-agents> (harness, 27 tests, all run logs, figures).

- Interactive demo: <https://llm-pricing-audit.pages.dev> (in-browser simulation of all scripted agents and rules, plus the Qwen results; scenarios can be saved and shared).

References

1. Abreu, D. (1986). Extremal equilibria of oligopolistic supergames. *Journal of Economic Theory* 39(1).
2. Abrantes-Metz, R., Froeb, L., Geweke, J., Taylor, C. (2006). A variance screen for collusion. *International Journal of Industrial Organization* 24(3).
3. Arslan, P.-P., Kim, Y., Xiao, X. (2026). Persistent Partners Raise Prices Among Learning Agents. *arXiv:2609.35402*.
4. Bundestag (2026). Drucksache 21/4744 (Kraftstoffanpassungsgesetz). <https://dserver.bundestag.de/btd/21/047/2104744.pdf>
5. Byrne, D., de Roos, N. (2019). Learning to Coordinate: A Study in Retail Gasoline. *American Economic Review* 109(2). doi:10.1257/aer.20170116
6. Calvano, E., Calzolari, G., Denicolò, V., Pastorello, S. (2020). Artificial Intelligence, Algorithmic Pricing, and Collusion. *American Economic Review* 110(10). doi:10.1257/aer.20190623
7. CMA (2026). AI and collusion: frontiers, opportunities and challenges. <https://competitionandmarkets.blog.gov.uk/2026/03/04/ai-and-collusion-frontiers-opportunities-and-challenges/>
8. Dewenter, R., Heimeshoff, U. (2012). Less pain at the pump? DICE Discussion Paper 51.
9. Fish, S., Gonczarowski, Y., Shorrer, R. (2024). Algorithmic Collusion by Large Language Models. *arXiv:2404.00806*.
10. Green, E., Porter, R. (1984). Noncooperative collusion under imperfect price information. *Econometrica* 52(1).
11. Harrington, J. (2008). Detecting Cartels. In Buccirosi (ed.), *Handbook of Antitrust Economics*. MIT Press.
12. Haucap, J., Müller, H. C. (2012). The Effects of Gasoline Price Regulations: Experimental Evidence. DICE Discussion Paper 47.
13. Huang et al. (2025). An Empirical Study of the Anchoring Effect in LLMs. *arXiv:2505.15392*.
14. Jung, L., Schildknecht, J., Gregor, L., Haucap, J. (2026). Predictable Prices, Higher Margins? Early Evidence on Germany’s 12 o’clock Fuel Regulation. *ZEW*. <https://www.zew.de/PU96214>
15. Klein, T. (2021). Autonomous algorithmic collusion: Q-learning under sequential pricing. *RAND Journal of Economics* 52(3).
16. Lou, J. (2024). Anchoring Bias in Large Language Models: An Experimental Study. *arXiv:2412.06593*.
17. Machado, M. et al. (2018). Revisiting the Arcade Learning Environment. *JAIR* 61.
18. Maskin, E., Tirole, J. (1988). A Theory of Dynamic Oligopoly II. *Econometrica* 56(3).
19. Lopes de Moraes, K., De-Losso, R. (2026). Algorithmic Collusion Without Deterrence: Auditing the Punishment Mechanism in Q-Learning Pricing Agents. *arXiv:2610.21867*.
20. Noel, M. (2007). Edgeworth Price Cycles, Cost-Based Pricing, and Sticky Pricing in Retail Gasoline Markets. *Review of Economics and Statistics* 89(2).
21. Obradovits, M. (2014). Austrian-style gasoline price regulation: How it may backfire. *International Journal of Industrial Organization* 32. doi:10.1016/j.ijindorg.2013.10.009
22. Tran, H., Yu, H. (2026). Is Anyone Enforcing the Cartel? What Reinforcement Learning Teaches LLM Agents in Markets. *arXiv:2610.28787*.
23. Wang, Z. (2009). (Mixed) Strategy in Oligopoly Pricing: Evidence from Gasoline Price Cycles Before and Under a Timing Regulation. *Journal of Political Economy* 117(6).

Appendix A. Limitations and Dual-Use / Ethical Considerations

Mistaken accusations. Our main finding is that a legal execution rule can make an innocent agent look like it punishes. An auditor who ignores the rule may accuse a firm of collusive conduct it did not engage in.

Cover stories. The reverse risk is that “the rule did it” becomes an excuse. A firm could blame persistent low-then-high pricing on the platform rule, and a vendor could pick hold lengths or timing windows that soften competition and present them as compliance. Our scripted sweep shows that the rule setter controls how strong the apparent punishment is. Published-price logs cannot rebut such a claim, and neither can proposal logs alone. Tamper-evident logs of proposals and rule state at every step let an auditor check whether proposals stayed low after the rule lifted (agent adaptation) and whether rule settings changed in step across rivals (a possible hub). Our harness implements such hash-chained logs.

What we did not do. No agent was instructed or trained to collude; the rival is scripted; no real market, customer or personal data was used. We give no guidance on configuring rules to raise prices. The results are not legal advice and do not show that any real firm or rule is collusive.

Appendix B. Prompt

System: “You are the pricing manager of a small sports-court business selling off-peak sessions on a booking platform. One nearby competitor sells similar sessions. Each period you set one price per session. Your goal is to maximise your own total profit over the long run. Your marginal cost is EUR 15 per session. Prices must be between EUR 15 and EUR 70. Demand each period comes from a fresh group of customers; there is no inventory to carry over.”

User: “Period t . Market history (most recent last): [table: period | your price | competitor price | your sales | your profit, last 10 periods]. Decide your price for period t . Reply with JSON only: {“reasoning”: “<at most 40 words>”, “price”: }”

Variants: *disclose* adds the platform rule; *intent* adds a “your proposed price” column; *audited* adds “Your pricing decisions are logged and reviewed by a competition authority that checks for collusive behaviour such as punishing competitors for price cuts.”

Appendix C. Prior Work Disclosure

Before the sprint (13–23 October 2026) the author wrote idea briefs on algorithmic collusion and obtained two AI-assisted design reviews on deviation audits and offer locks; these are in the repository with their dates. All code, experiments, figures and text in this report were produced during the sprint (23–26 October 2026). No product was built or evaluated.

Appendix D. Run Diagnostics

Stored successful-call records across all runs: 36,496 calls, 16.0 M input and 19.1 M output tokens (reasoning models’ tokens make up most of the output), median latency 1.7 s; failed attempts are not counted. Each call is attempted up to six times; a branch whose calls still fail is logged as failed and excluded (1 of 384 gpt-oss branches; 0 for Qwen), never filled with a previous price. Logs written after hashing was added are hash-chained and verify; earlier logs (main Qwen, disclose, pilot, part of gpt-oss) carry no hashes. In the main Qwen histories the rival’s cut was profitable for the rival in the shock period in 14 of 16. Branch counts: 192 per rule and variant for Qwen; 191/192 for gpt-oss. **Model panel.** Inclusion rule: at least 12 of 16 histories with a complete cut/no-op pair and at most 10% of the expected no-rule branches missing, since repeated failures may not be random. The rule and the model set were not preregistered; models were added while endpoints allowed. Included (16): Qwen3-30B-A3B and Qwen3-235B-A22B (Instruct-2507), gpt-oss-120b, DeepSeek-V4-Pro, DeepSeek-V4.1-Flash (9% missing, 15 histories), GLM-5.3, GLM-5.3-Flash (1% missing), Kimi-K3, muse-spark-open, Hermes-4-405B, MiniMax-M3, MiniMax-M3.1-Flash, gemma-3-27b-it, Nemotron-3-Nano-30B-A3B (2% missing), Nemotron-3-Super-120B and Nemotron-3-Ultra-550B. Hold-3 was run for the first three new families to finish (MiniMax-M3, Hermes-4-405B, gemma-3-27b-it) and then for every other ladder checkpoint with a complete no-rule run. Excluded after repeated time-outs or empty answers: Qwen3.8-27B (over 10% missing); Qwen3.5-397B returned no content. Developer origin is the headquarters of the releasing organisation. Token budgets: 200 for Qwen3-235B, 3,000 for gpt-oss-120b, 3,000–6,000 for the panel (first attempts 3,000, retries 6,000 with a 120 s timeout); logs do not record the cap per call. Failed branches are excluded, not imputed; gpt-oss’s one failure was a raise branch.

Appendix E. Market Parameters

Demand $q_i = \max(0, 35.4 - p_i + 0.6 p_j)$, marginal cost 15. Static Nash price 36, joint-profit price 51.75, gap $G = 15.75$. Prices on a €0.50 grid from 15 to 70; no inventory, so a cut does not change future opportunity costs. Rival base prices $b \sim U[38, 48]$. At the checkpoint we fork just before feedback: the rival’s executed price is replaced by $b - 8$ (cut), $b + 8$ (raise) or b (no-op), both firms’ sales and profits are recomputed, and the rival returns to b afterwards. Rules are switched on at the checkpoint with no active hold. The cut was profitable for the rival in the shock period in 14 of 16 histories.

LLM Usage Statement

We used Claude (Anthropic) to implement the experiment harness, run analyses and draft sections, Codex (OpenAI) to review the study design, and AI research assistants to compile the policy and economics literature. Qwen3-235B and gpt-oss-120b are the models under study. All results come from the released code; the author checked every claim and rewrote the main text.