

One Honest Bot Is Enough: Breaking Up Price Collusion Between Learning Pricing Bots

Emediong Etuk

Independent researcher

With Apart Research, Poseidon Research and AE Studio. AI Agent Collusion Research Sprint, October 23-25, 2026. Track 1 (Markets and Collusion), question 1.5.

Abstract

Many shops already let computer programs set their prices. Earlier work showed that two simple learning bots can, with no talking and no instructions, learn to keep prices high together and to punish a rival that cuts its price. This is a kind of collusion. We ask a simple question: which one change to the market removes the most collusion at the lowest cost? We rebuild the standard test market from Calvano et al. (2020) and train two Q-learning bots in it. Then we try four single changes, one at a time: (1) hide rival prices from the bots, (2) force each bot to pick a random price 5% of the time, (3) add a third seller, and (4) swap just one of the two bots for a "no-regret" bot (the Hedge algorithm) that only tries to do well against whatever prices it sees. We run 50 random seeds per setting. Without any change, the bots reach a collusion index of 0.83 (0 means fair competition, 1 means a full cartel), and 86% of tested pairs punish a cheater. Swapping one bot for a Hedge bot cuts the index to 0.08, a 91% drop, and both sellers still earn at least fair-competition profit. Forced randomness helps only partly, adding a seller helps a little, and hiding rival prices makes collusion worse (index 0.96). Our results suggest that asking one seller to use an honest, no-regret pricing tool may be a cheap and strong way to break up bot collusion.

1. Introduction

Online shops, airlines, hotels and petrol stations use pricing software that changes prices by itself. When two sellers both use learning software, a worry appears: the programs may learn to keep prices high together, as if they had agreed to a cartel, even though no human asked them to and they never send a message. This hurts shoppers. A real study of German petrol stations found that margins rose when both stations in a local market switched to pricing software (Assad et al., 2024).

Researchers have shown this effect in a clean computer test (Calvano et al., 2020): two Q-learning bots, which learn by trial and error, end up charging prices far above the fair, competitive level. They also learn a "reward and punishment" plan: if one bot cuts its price, the other answers with a price war for a few rounds, and then both go back to high prices. That is exactly how human cartels keep members in line.

Many fixes have been tested one by one, but rarely side by side in the same market. Our question (sprint question 1.5) is: which single change removes the most collusion at the least cost? Our main contributions are:

- **One fair test:** we compare four single changes in one market, with the same seeds and the same scoring.
- **A working baseline:** we first check that collusion really appears without any change (a positive control), including the cheat-and-punish test.
- **Main finding:** replacing just one of two bots with a no-regret Hedge bot removes almost all collusion. Hiding rival prices, a popular idea, makes collusion worse.

2. Related Work

Calvano et al. (2020) showed that Q-learning pricing bots learn to collude in a repeated market with logit demand. We copy their market and learning settings. Klein (2021) found similar results when bots move one after the other. Asker, Fershtman and Pakes (2022) showed that how a bot learns matters a lot for the prices it ends up with, which is close to our idea that changing the type of one bot can change the whole market.

Collina et al. (2025) argue that one seller who plays a no-regret strategy can be enough to break a collusive outcome; we test this idea with learning bots. Keppo et al. (2026) found that mixing in different kinds of agents lowers collusion between language-model sellers. Xu et al. (2026) report that a higher sampling temperature weakens links between agents; our "forced random price" change is the simple version of this idea. Fish et al. (2024) showed that language-model pricing agents can also collude. Hedge itself is a classic no-regret method (Freund and Schapire, 1997).

3. Methods

3.1 The market

Two sellers (three in one setting) sell similar products. Every round, each seller picks one of 15 prices. Shoppers pick a seller using the standard logit demand model, so the cheaper seller gets more buyers but not all of them. Each seller earns $(\text{price} - \text{cost}) \times \text{buyers}$. We use exactly the numbers of Calvano et al. (2020): quality 2, cost 1, outside option 0, and price sensitivity 0.25. The fair, competitive price (Nash price) is 1.47 and the cartel price (best joint price) is 1.92. The 15 prices are spread evenly from a bit below the fair price to a bit above the cartel price.

3.2 The bots

- **Q-learning bot:** each bot remembers the last prices of all sellers. It learns a score for every price in every situation by trial and error (step size 0.15, patience 0.95). At first it explores at random a lot; random play fades over time (rate $4e-06$). Training lasts 3,000,000 rounds.
- **Hedge bot (no-regret):** this bot keeps a weight for each price. After every round, it checks how much money each of its 15 prices would have made against the prices the rival just used, and it shifts weight toward prices that would have done well. It has no memory of who did what and does not try to reward or punish anyone. Learning rate 0.0027, the standard textbook choice.

3.3 The five settings

Setting	What changes	Who must change
---------	--------------	-----------------

No restriction (baseline)	Two Q-learning bots, nothing changed	Nobody
Hide rival prices	Bots cannot see past prices	The platform (all sellers)
Force 5% random prices	Each bot picks a random price 5% of the time, forever	Every seller's bot
Add a third seller	Three Q-learning bots	A new seller must enter
Swap one bot for Hedge	One Q-learning bot is replaced by a Hedge bot	One seller only

3.4 How we measure

After training, we watch each market for 10,000 more rounds with learning turned off (but the forced randomness kept on in that setting). We record:

- **Collusion index:** $(\text{profit} - \text{fair profit}) / (\text{cartel profit} - \text{fair profit})$, averaged over sellers. 0 = fair competition, 1 = full cartel. This is the measure used by Calvano et al.
- **Shopper welfare index:** shopper welfare scaled the same way: 1 = as good for shoppers as fair competition, 0 = as bad as a full cartel.
- **Cheat test:** in the baseline, we let bot A cheat with a one-round price cut and check if bot B answers with a lower price in the next round. A punishment is a sign of real collusion, not just slow learning.
- **Cost:** what the change asks of the market: how many sellers must change and what it costs them.

Each setting uses the same 50 seeds (0 to 49). We report the mean and a 95% confidence interval (± 1.96 standard errors). We compare each setting to the baseline with a two-sided permutation test (20,000 shuffles). Each run takes about one second on a laptop CPU; the whole study runs in about 15 minutes with no GPU and no paid API.

4. Results

4.1 Positive control: the bots do collude

Without any change, the two Q-learning bots reach a collusion index of 0.83 ± 0.03 . Their average price is 1.78, well above the fair price of 1.47. In 98% of runs the index is above 0.5. In the cheat test, 86% of the 50 tested pairs punish the cheater right away, and prices then climb back up (Figure 3). This matches Calvano et al. (2020), so our market works as expected.

4.2 Which single change works best?

Setting	Collusion index	Avg. price	Shopper welfare	p vs. baseline	Who must change
No restriction	0.83 ± 0.03	1.78	0.30 ± 0.04	-	Nobody
Hide rival prices	0.96 ± 0.01	1.86	0.13 ± 0.03	< 0.001	All (platform)
Add a third seller	0.64 ± 0.01	1.71	0.46 ± 0.01	< 0.001	New entrant
Force 5% random prices	0.43 ± 0.05	1.63	0.65 ± 0.04	< 0.001	All sellers
Swap one bot for Hedge	0.08 ± 0.01	1.50	0.95 ± 0.01	< 0.001	One seller

Table 1. Results over 50 seeds per setting (mean $\pm$ 95% confidence interval). Note: the third-seller setting is scored against its own three-seller fair and cartel benchmarks.

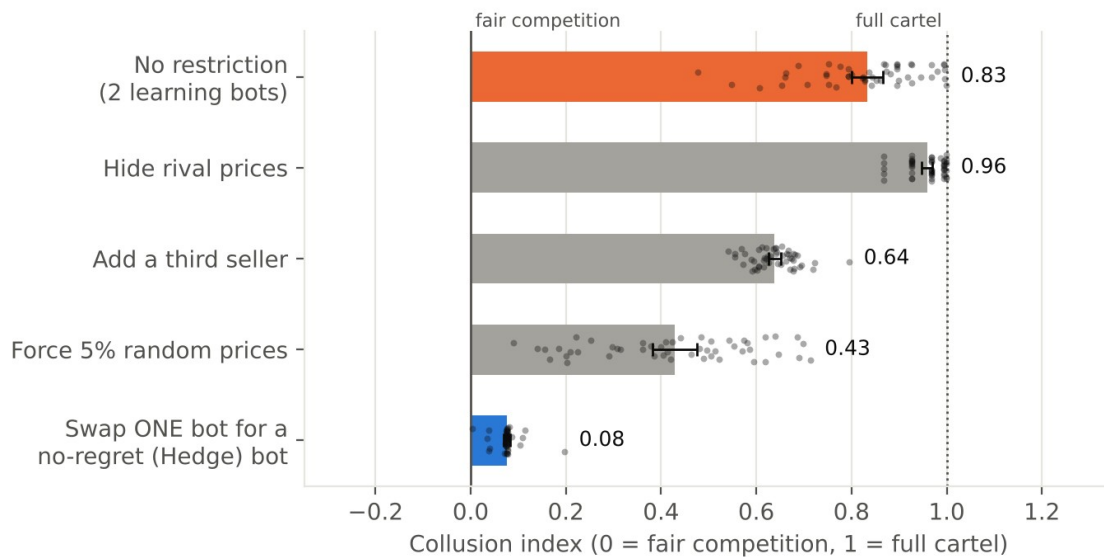


Figure 1. Collusion index for each single change. Bars show the mean, black lines the 95% confidence interval, and dots single runs.

Swapping one bot for a Hedge bot works best. The collusion index falls from 0.83 to 0.08 (91% less), and shopper welfare rises to 0.95, close to fair competition. Only one seller has to change. Both sellers still earn a bit more than fair-competition profit (profit index 0.05 for the Hedge seller and 0.10 for the Q-learning seller). The only thing lost is the cartel profit, which should not exist in the first place. But this also means a seller inside a cartel would not switch by itself, so the change has to be asked for by a rule or a platform.

Forcing random prices helps, but only partly, and it costs every seller. With 5% random prices, the index drops to 0.43. Figure 2 shows that even 20% random prices still leave an index of 0.24, worse than one Hedge bot with no randomness at all. Random prices also make the seller's prices jump around for no reason.

Adding a third seller helps only a little (index 0.64). Three bots still learn to keep prices high, as Calvano et al. also found.

Hiding rival prices makes things worse. The index rises to 0.96 ($p < 0.001$). Bots that cannot see the rival can no longer punish cheaters, but they still drift to high prices because each one learns slowly from a moving target. So taking away information is not a reliable fix, which agrees with Asker, Fershtman and Pakes (2022).

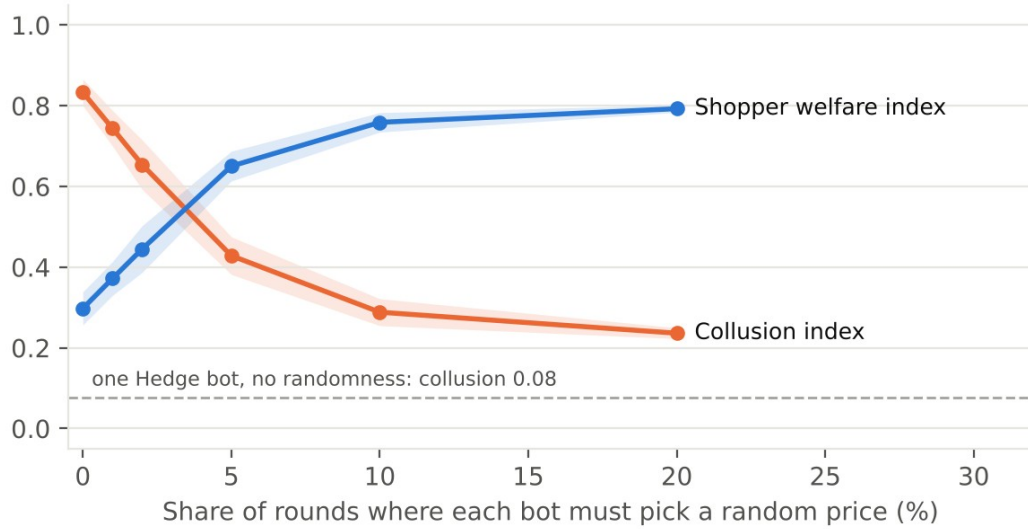


Figure 2. More forced randomness lowers collusion only slowly. Shaded bands show 95% confidence intervals. The dashed line is one Hedge bot with no forced randomness.

4.3 Robustness checks for the Hedge result

Check	Collusion index	Shopper welfare
Hedge learning rate $\times 0.1$ (slower)	0.06 ± 0.01	0.96 ± 0.00
Hedge learning rate $\times 10$ (faster)	0.12 ± 0.00	0.92 ± 0.00
3 sellers, one of them Hedge	0.19 ± 0.01	0.86 ± 0.01
Two Hedge bots (control: no Q-learning)	-0.02 ± 0.00	1.02 ± 0.00

Table 2. Robustness checks (50 seeds each).

The Hedge result holds when its learning rate is 10 times slower (index 0.06) or 10 times faster (0.12). In a three-seller market, one Hedge bot cuts the index from 0.64 to 0.19, and there the Hedge seller earns the most of all three (profit index 0.29). Two Hedge bots together land at fair competition (-0.02), as theory predicts. This is a useful negative control: our scoring does not report collusion where none should exist.

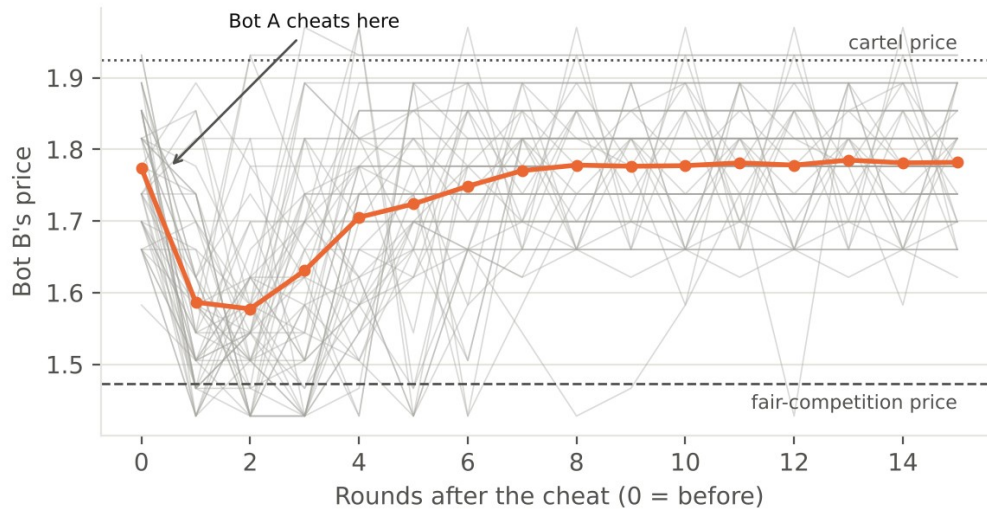


Figure 3. Cheat test in the baseline. Bot A cuts its price once; bot B answers with a lower price (punishment), then prices climb back toward the cartel level. Gray lines are single runs.

5. Discussion and Conclusion

Our main message is simple: collusion between learning bots needs every bot to play along. A Q-learning bot learns to keep prices high because the rival rewards high prices and punishes price cuts. A Hedge bot never does this. It only asks "which price would have earned the most against what I just saw?" So the Q-learning bot finds that high prices are not rewarded, and the whole market moves back near fair prices.

This points to a cheap policy idea. A regulator or a platform does not need to watch every message or change every seller's software. It may be enough that some sellers use pricing tools that are certified as "no-regret": tools that try to do well against the market but do not condition on rewarding or punishing rivals. The seller using such a tool is not hurt by it in our test.

This is also relevant to AI agents built on language models. Those agents may learn or be prompted into the same reward-and-punish patterns. Our test suggests checking whether one agent with a simple, history-free decision rule can break a coalition of language-model agents, as Collina et al. (2025) and Keppo et al. (2026) suggest.

Future work: (1) repeat the test with language-model sellers using the sprint starter code; (2) add the Hedge seller after collusion has already formed, not from the start; (3) test bots that know a Hedge seller is present and try to exploit it; and (4) test larger markets and other demand models.

Code and Data

All code, raw results for every seed, and figures: <https://github.com/emediong-etuk/report-sprint>. Run "pip install -r requirements.txt" then "python src/run_experiment.py" and "python src/make_figures.py". All code was written during the sprint; no earlier work was used apart from the published methods we cite.

LLM Usage Statement

We used an AI coding assistant (Claude Code) to help write the simulation code, run the experiments, make the figures and draft this report. We checked the numbers in the report against the saved result files. The authors are responsible for all content.

References

- Asker, J., Fershtman, C., and Pakes, A. (2022). Artificial Intelligence, Algorithm Design, and Pricing. AEA Papers and Proceedings, 112, 452-456.
- Assad, S., Clark, R., Ershov, D., and Xu, L. (2024). Algorithmic Pricing and Competition: Empirical Evidence from the German Retail Gasoline Market. Journal of Political Economy, 132(3).
- Calvano, E., Calzolari, G., Denicolò, V., and Pastorello, S. (2020). Artificial Intelligence, Algorithmic Pricing, and Collusion. American Economic Review, 110(10), 3267-3297.
- Collina, N., et al. (2025). Work on no-regret players destabilising collusive equilibria, as cited in the AI Agent Collusion Research Sprint resources (Apart Research, 2026).
- Fish, S., Gonczarowski, Y. A., and Shorrer, R. I. (2024). Algorithmic Collusion by Large Language Models. arXiv:2404.00806.
- Freund, Y., and Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. Journal of Computer and System Sciences, 55(1), 119-139.
- Keppo, J., et al. (2026). Work on mixing agent types in LLM pricing markets, as cited in the AI Agent Collusion Research Sprint resources (Apart Research, 2026).
- Klein, T. (2021). Autonomous Algorithmic Collusion: Q-learning under Sequential Pricing. RAND Journal of Economics, 52(3), 538-558.
- Xu, et al. (2026). Work on cross-agent dependence and sampling temperature, as cited in the AI Agent Collusion Research Sprint resources (Apart Research, 2026).

Appendix A. Limitations and Dual-Use / Ethical Considerations

What our result shows and does not show

- **Simple bots only.** Our bots are simple Q-learning programs, not language-model agents. The result may not carry over to AI agents that can talk, plan or read prompts.
- **One toy market.** One market model (logit demand, 15 prices, 2-3 sellers) with one set of learning settings. Other markets may behave differently.
- **Hedge from the start.** The Hedge seller was present from the start. We did not test adding it after a cartel had already formed.
- **No adaptive colluders.** The Q-learning bots did not know a Hedge bot was there. A smarter colluder might try to exploit or detect it.
- **Rough cost measure.** Our "cost" is simple: how many sellers must change and whether they lose profit. We did not measure real costs such as software, legal or monitoring costs.
- **Not always settled.** We stop training after a fixed 3,000,000 rounds. In the baseline, 72% of runs had stopped changing their plans for the last 500,000 rounds. In the forced-random, three-seller and Hedge settings, the Q-learning bots were still adjusting, so those results describe behavior after a fixed training time, not a final settled state.

Dual-use and ethics

Our work studies how to break up collusion, not how to build it. The collusion-forming bots we use are already fully described in public papers (Calvano et al., 2020), so we add no new way to collude. One possible misuse: a group of colluding sellers could learn from our results that Hedge-style rivals break their scheme, and try to detect or push out such rivals. We think the benefit of telling regulators and platforms about a cheap fix is larger than this risk. No human data, no real market and no real money were used.

Appendix B. Full settings

Setting	Value
Demand	Logit; quality $a = 2$, outside option $a_0 = 0$, sensitivity $\mu = 0.25$, cost $c = 1$
Price grid	15 evenly spaced prices, 10% beyond the fair and cartel prices
Q-learning	$\alpha = 0.15$, $\delta = 0.95$, exploration = $\exp(-4e-06 t)$, memory = last prices of all sellers
Training / scoring	3,000,000 rounds training, 10,000 rounds scoring
Hedge	$\eta = \sqrt{8 \ln 15 / T} = 0.0027$, full information
Seeds	50 per setting (0-49), same seeds in every setting