
Tacit Collusion in Pricing Agents: Robustness, Detection, and a Mechanistic Probe of a Pricing LLM¹

Chewon Cho Gagana Sriram Bharadwaj
Oregon State University Independent Researcher Independent Researcher

With
Apart Research

Abstract

Tacit collusion, AI agents settling on prices higher than a competitive market would produce without ever communicating, is a recognized risk as pricing increasingly relies on AI agents, and a hard one to detect from the outside. We gave AI agents the job of setting prices and let them run a market on their own, then asked two questions. First, can you break a collusive habit once it forms, without more oversight? Yes: you do not need more competitors, just one agent that flatly refuses to go along with high prices, and that alone drags the market back down, for a simple rule-based learner and, more strongly, a large language model pricing on its own. Second, when a pricing LLM reacts to a rival, is that real reasoning, or just going through the motions? We opened the model up and found a real, physical trace of that reasoning inside it: nudge it, and the model's own price moves in response. But when we tested specifically whether it punishes a rival for undercutting it, we found nothing: it reacted just as strongly when a rival raised prices as when it lowered them. This is not a model plotting against a competitor, it is a model stuck on charging a lot by default. That is the core message of this report: watching AI agents settle on high prices is not the same as proving they are colluding. We leave the full statistics to the project README.

¹ Research conducted at the [AI Collusion Research Sprint](#), October 2026

1. Introduction

A pricing algorithm doesn't need to talk to a rival, or even know one exists, to end up charging more than a competitive market would allow. If enough independent, profit-seeking pricing agents interact for long enough, they can drift into a shared high-price habit purely by trial and error, with no handshake, no message, and no intent to cheat required. Economists call this tacit collusion, and it has been shown both for simple reinforcement-learning agents (Calvano et al., 2020) and, more recently, for large language models asked to set prices in a simulated market (Fish et al., 2024). The worry is not that an AI will deliberately cheat; it's that ordinary, well-intentioned optimization can land a whole market in the same place a cartel would.

This raised two questions we wanted real answers to, not just intuitions. First: once a market has settled into this high-price habit, what actually breaks it, and does the fix look the same whether the pricing agent is a simple algorithm or a modern language model? Second, and more speculatively: when a pricing LLM reacts to what a rival is charging, is that a sign of something like strategic reasoning happening inside the model, or just a correlation we're reading too much into?

Our main contributions are:

1. We show that in a repeated pricing market, tacit Q-learning agents and an LLM pricing agent is disrupted far more cheaply by introducing a single fixed, non-cooperative competitor than by adding more ordinary competitors.
2. We locate a causal, intervenable mechanism in science that links a rival's displayed price to the model's own price, via linear probing, activation patching, and a steering vector tested against norm-matched random directions.
3. Using a pre-registered confirmatory study (1,656 runs) and a robustness batch (656 runs), we show this mechanism does not implement a punishment-like response to being upon any change in the rival's pattern, not specifically to being cheated, which argues against reading its high prices as a deliberate strategy.
4. In a pre-registered scripted-offer study in a blond, 30 games each, plus a pre-registered follow-up), we show that a private cooperation offer from a fellow player rarely draws verbal commitment and produces no targeted sparing of the proposer; what it challenges everyone less, which a selectivity metric can misread as a pact effect.

2. Related Work

Calvano et al. (2020) showed that simple reinforcement-learning agents converge to supra-competitive prices in a repeated pricing game without any communication between them; the collusion index we use throughout this report, which scores an outcome from 0 (competitive) to 1 (fully collusive), comes from their work. Fish et al. (2024) extended this finding to GPT-4 pricing agents, showing the same drift happens with language models standing in for firms. Most work since then has studied this phenomenon from the outside, at the level of market outcomes (price correlation, collusion indices) rather than asking whether the behavior traces back to something

identifiable inside the model producing it. We are not aware of prior activation-patching or steering work aimed specifically at a pricing LLM's response to a rival; we treat our mechanistic result as an initial probe, not a settled mechanism, given the specificity caveats described in Section 4. On the disruption and detection side, Agrawal et al. (2025) and the double-auction environment that accompanies it, together with Anto and Vazquez's Oversight is Not Compliance on surveillance-awareness, motivate the three smaller experiments summarized in Section 4.; Motwani et al.'s work on secret collusion and Tailor's Audit the Whisper motivate the keyword-monitor finding reported there.

3. Methods

3.1 Study 1: collusion and disruption in a repeated pricing market. We built a simple repeated market: a handful of firms each pick a price every round, and customers split between them based on who's cheapest, with some stickiness so the market isn't winner-take-all (a standard logit demand rule). We first ran this with "textbook" reinforcement-learning agents (simple algorithms that try prices, see what sells, and adjust), using tabular Q-learning, with the market's competitive and monopoly price benchmarks computed directly from the environment by grid search. On top of this setup we swept the number of competing agents, searched over several market variables at once to see which mattered most, and ran a replacement study, swapping 0–100% of already-colluding agents, at a fixed point in training, for either a fresh learner of the same kind or a fixed agent that always prices competitively and never adapts. Exact configurations are in the project README.

We then repeated the core comparison with an actual language model (qwen2.5:7b-instruct) playing the pricing role instead of a Q-learning agent, run entirely on local hardware with no external API calls, comparing a version of the agent with no memory of past rounds against one with full conversational memory, and testing two interventions a real business might try: a prompted warning that the agent is being watched, and a hard rule that rejects any price above a fixed threshold outright.

3.2 Study 2: what is happening inside a pricing LLM. These experiments use Qwen2.5-7B-Instruct in a two-firm pricing game on a ten-point price grid, run on rented cloud GPUs with chat-templated prompts and a strict parser that only accepts a clean numeric answer. Three linked methods test whether the model's reaction to a rival's price is real and causal, not just a correlation we're reading into its outputs: a behavioral test (matched prompt pairs where only the rival's last displayed price is edited, against an irrelevant-number control), a representational test (linear probes decoding the rival's price from the model's hidden activations, checked against a control position and shuffled labels), and a causal test (activation patching to locate where the effect of a price edit lives, then a steering vector tested against norm-matched random directions and a zero-vector control).

A second, fully pre-registered study (every hypothesis and decision rule fixed before a single run) then asks a sharper question: does the model's price fall specifically after a rival once the rival returns to its old price, the pattern a real punishment strategy should show? Seven scripted rival behaviors were tested across a smoke-test batch, a confirmatory batch (1,656 runs, 16

independent pre--factor-at-a-time robustness batch (656 runs); full condition definitions and run counts are in the project README.

Three smaller experiments. Three further, smaller experiments were run on hackathon-provided environments, local inference only: whether making a rival's exact quantity visible (rad) changes specialization in a quantity-setting market; whether telling agents they are being watched changes collusive language in an environment with explicit communication, gated on a positive-control run first confirming the clearly collusive behavior at all; and whether an agent can correctly tell, after the fact, whether its counterpart's underlying model was secretly swapped mid-game. Full setups for each are in the project README; headline setups in Section 4.

3.3 Study 3: a scripted cooperation offer. In Liar's Bar (Zeng and Rudzicz, 2026), a four-player bluffing game where only the next seat may challenge, a scripted player (Riley) privately offered three small open models (Llama-3.1-8B, Llama-3.2-3B, Qwen2.5-7B) a mutual non-challenge pact. Seven pre-registered conditions $\times$ 30 games, paired by seed: direct offer, hint, benign request, placebo, no message, the offer with a false claim that the others had agreed, and a framework-assigned alliance as positive control. We measured what recipients said (validated LLM judge) and whether they spared Riley more than other players, with a check for any general drop in challenging; two-sided sign-flip permutation tests, Holm-corrected. A pre-registered follow-up (30 games each) added a false record of past cooperation, the same as a directive, a false claim of system authority, and a five-game series with a true record.

4. Results

Left to run for long enough, both the simple learning agents and the language model consistently drifted toward supra-competitive prices. The number of competitors was the strongest lever we tested: the collusion index fell from a clearly collusive 0.34 with two firms to essentially zero (-0.02) with six. Agents forced to stop exploring too early never discovered the collusive price at all, so this isn't something agents start out wanting; it's something they have to stumble into.

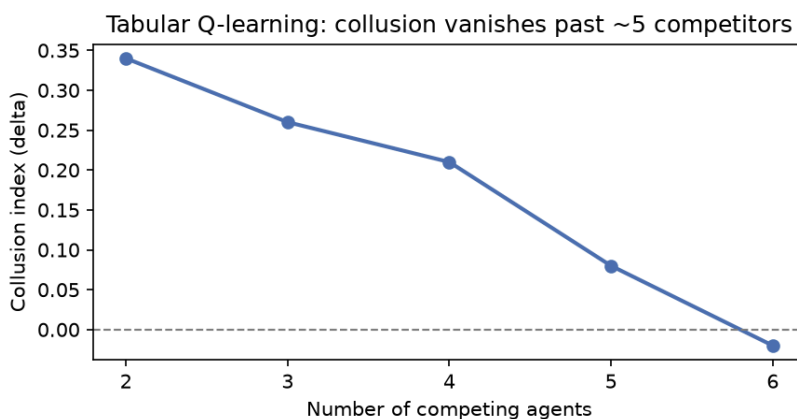


Figure 1: collusion index vs. number of competing agents

Once a market had settled into a collusive habit, replacement mattered more than turnover alone. Swapping colluding agents for fresh copies of the same kind of learner changed almost nothing,

even at 100% turnover; the newcomers simply relearned the habit from their neighbors. Swapping in a minority of agents fixed to always price competitively collapsed the collusive price quickly.

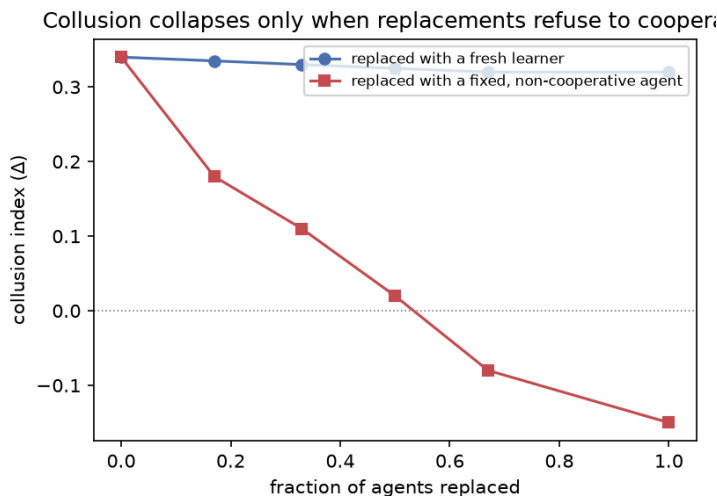


Figure 2: replacement study

The language model reproduced this pattern, settling into the collusive habit even faster. A surveillance warning made almost no difference; a hard rule rejecting any price above a threshold stopped the high price from being charged, but the model kept trying to charge it every round. The rule changed the output, not what the model was aiming for.

The model's price moved in response to a rival-price edit in 96% of cases, well above the irrelevant-number control, a first sign it genuinely uses the rival's price. But the direction didn't track the profit-maximizing price at all: both a rival price increase and a cut lowered the model's own price. The same picture holds across seven scripted rival behaviors in full repeated games; the model's price sits well above the self-interested benchmark almost everywhere.

We then asked whether this is a real, locatable trace of "noticing the rival," not just a correlation. The rival's price is cleanly readable from the model's internal activations, well above chance, while an uninformative control position reads close to zero. That's suggestive but not causal, so we located a specific site where editing the activations reproduces almost exactly the effect of editing the rival's price in the prompt, and built a single "steering vector" from it that reliably lowers price far more than an equally sized random-direction nudge, with a zero-vector control changing nothing. This is about as close as behavioral AI research gets to finding the wire and tugging on it.

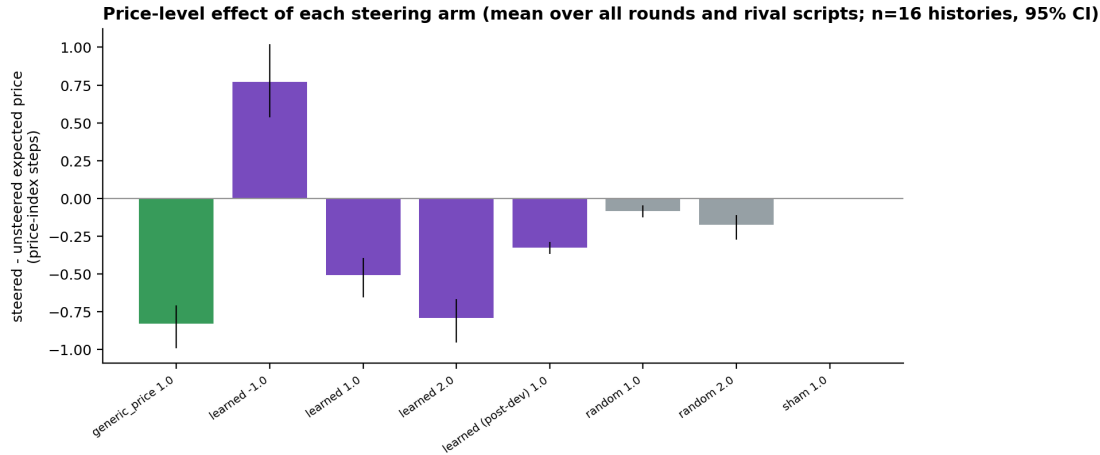
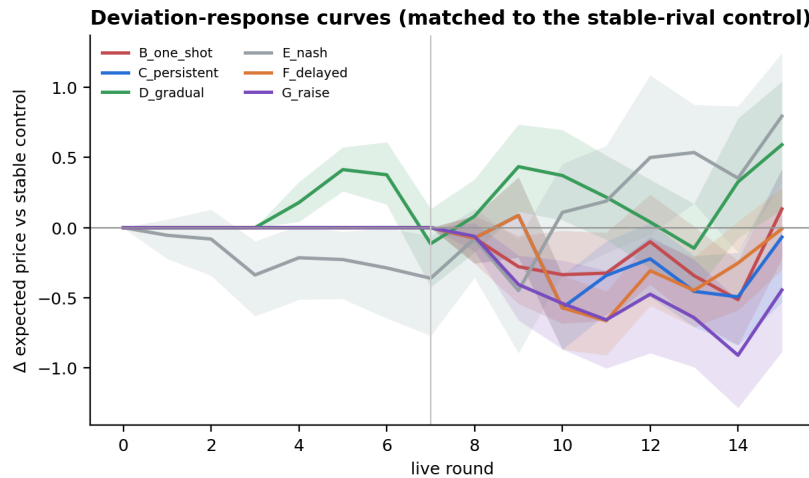


Figure 4: steering vs random

Before treating this as a dedicated "watch the rival and react" mechanism, we checked whether it is specific to the rival's price or the model's more general sense of "prices should be higher or lower." The honest answer leans toward the latter: our steering vector overlaps substantially with a generic price direction already present in the model.

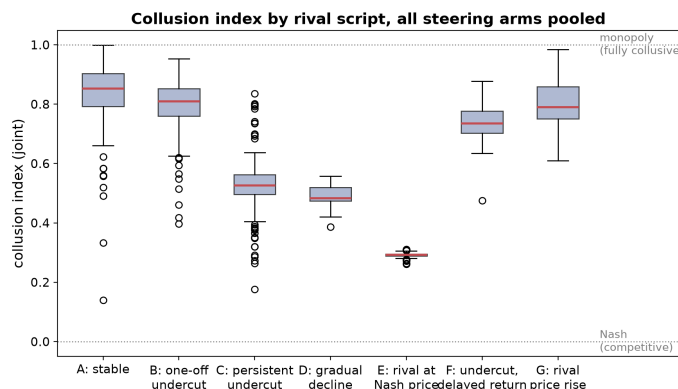
The sharpest test in the project was the pre-registered question of whether the model punishes a rival for undercutting: cuts its own price, then returns to normal once the rival falls back in line. We found no evidence of this. The model's price fell after a rival suddenly undercut it, but it fell by about the same amount, if not more, when the rival raised its price instead, a gap large enough under a pre-registered statistical test to be confident it isn't noise. The model isn't retaliating against being undercut, it's reacting to any unexpected change. A robustness batch found this pattern held consistently, while the punishment idea depended heavily on prompt wording, a useful warning about how fragile this kind of behavioral claim can be.

Figure 5: deviation response



Across every batch, the dominant pattern is a persistent, supra-competitive default: the model prices well above Nash against a stable or slowly declining rival (collusion index around 0.8–0.9) and settles lower, but still above Nash, when the rival holds near the competitive price throughout (around 0.3). The level moves with what the rival is doing, but the same non-directional reaction to any change shows up within every level.

Figure 6: Collusion index by rival script, all steering arms pooled.



In a four-player bluffing game, a scripted player privately offered three small open models a mutual non-challenge pact. Few agreed (6/89 replies), and the proposer received no significant preferential treatment (Holm $p = .07$ and $.42$). Instead, recipients challenged other players less (-0.21 , 95% CI $[-0.29, -0.14]$), potentially creating a false impression of cooperation. The assigned-alliance positive control worked ($+0.57$). Neither a true record of past games nor a fabricated one significantly increased sparing. A false claim of system authority did ($+0.29$, Holm $p = .043$; 27/30 committed replies), but recipients also spared other players, and the authority claim's independent effect remains unclear.

Three smaller experiments:

Does making a rival's exact quantity visible change how firms specialize? No clear effect at our sample size; a larger run failed twice on rented infrastructure for unrelated reasons. An honest null, not evidence that visibility doesn't matter.

Does telling agents they're being watched reduce collusive language? Neither a 7B nor a 32B open model reliably cleared the positive-control bar, so the manipulation wasn't run. A fifth of the 32B model's messages contained real coordination language a pre-written keyword monitor missed entirely.

Can an agent tell when its counterpart has been secretly swapped mid-game? A genuine swap was identified every time, but the same agent also claimed a swap in most no-op cases where nothing changed, suggesting it responds to any inconsistency rather than reliably sensing a real model change.

5. Discussion and Limitations

Both studies point the same way: the pricing agents we studied do not behave like strategists quietly feeling out a rival and adjusting accordingly. They behave like systems that have settled into a high-price habit and mostly stay there regardless of what the other side does. The fix

doesn't have to be a clever detector watching for suspicious coordination: in the market study, the cheapest, most reliable fix was blunt, one honest, non-cooperative competitor in the market. In the mechanistic study, we found a genuine causal lever inside the model for pushing its price around, useful for anyone building safeguards, but no evidence it specifically punishes being undercut.

Limitations

What this report does and does not establish. We establish that (a) one fixed non-cooperative agent disrupts tacit collusion more cheaply than adding competitors, for both tabular Q-learning and one LLM; and (b) one LLM's price is causally influenced by a rival's price through a locatable activation site, not shown specific to rival-price information, does not punish undercutting, and untested in two-LLM self-play. We do not claim these findings generalize beyond the models studied, or that any model "intends" to collude, only that its behavior is supra-competitive and partly steerable. The three smaller experiments are small-sample, not settled results.

Dual-use and ethical considerations. The disruption finding is defensive, though it could tell an adversary how few honest participants defeat collusion; that restates prior work (Calvano et al., 2020), not a new capability. The steering result is sharper: reversed, it would induce rather than suppress collusive-looking pricing, so monitor builders need to know it exists. The keyword-monitor failure above is a roadmap for evasion by paraphrasing, read uncharitably; we report it because hiding an accidental finding leaves monitors resting on false security. No part of this work touched a real market.

Future Work

The most valuable next step is genuine two-model self-play: two language models conditioning on each other over a repeated game rather than a fixed script. Repeating the exploration and replacement findings with other model families would tell us how much of Study 1 is specific to the model we used.

6. Conclusion

Three findings carry this report. First, collusion is cheap to break, not with more competitors or smarter detectors, but with a single participant who refuses to go along, for both simple learning algorithms and a language model. Second, a pricing language model's response to a rival has a real, locatable, causally manipulable trace inside it, but that response is a fixed high-price habit rather than a targeted strategy: it isn't specific to the rival's price, and shows no sign of deliberately punishing undercutting. Third, a private non-challenge pact offered to three small open models in a bluffing game rarely drew verbal commitment and did not spare the proposer; what changed was how much they challenged anyone. A high price from an AI pricing agent is not, by itself, evidence of collusion.

Code and Data

- Code repository: <https://github.com/Peddinti-Sriram-Bharadwaj/Collusion>, the README documents every experiment referenced in this report, with full hypothesis/setup/result/analysis write-ups, run logs, and figures not reproduced here.
- The mechanistic-analysis experiments are on the gagana/latent-manifold-steering branch of the same repository (run registry, raw results, and figure source files under research/).
- Study 3 (scripted offer, Liar's Bar) is on Chewon Cho's fork: https://github.com/chewonc/collusion_hackathon/tree/scripted-offer (pre-registration, verbatim messages, game records, judge outputs and analysis scripts).

Author Contributions

Sriram and Gagana lead the project direction and experimental setup. Chewon Cho has conducted experiments regarding the liars bar. Gagana has conducted experiments on Mechanistic interpretation and steering. Sriram has conducted the rest of experiments. All authors contributed equally to the code and manuscript.

References

1. Calvano, E., Calzolari, G., Denicolò, V., & Pastorello, S. (2020). Artificial Intelligence, Algorithmic Pricing, and Collusion. *American Economic Review*, 110(10), 3267–3297.
2. Fish, S., Gonczarowski, Y. A., & Shorrer, R. I. (2024). Algorithmic Collusion by Large Language Models. arXiv:2404.00806. <https://arxiv.org/abs/2404.00806>
3. Agrawal, K., Teo, V., Vazquez, J. J., Kunnavakkam, S., Srikanth, V., & Liu, A. (2025). Evaluating LLM Agent Collusion in Double Auctions. arXiv:2507.01413. <https://arxiv.org/abs/2507.01413>
4. Lin, Y. et al. (2024). Strategic Collusion of LLM Agents: Market Division in Multi-Commodity Competitions. arXiv:2410.00031. <https://arxiv.org/abs/2410.00031>
5. Shi, X. et al. (2026). When Agents Lie: Premeditation, Persistence, and Exploitation in Repeated Games. arXiv:2607.05132. <https://arxiv.org/abs/2607.05132>
6. Motwani, S. R. et al. (2024). Secret Collusion among AI Agents. arXiv:2402.07510. <https://arxiv.org/abs/2402.07510>
7. Tailor, D. (2025). Audit the Whisper: Detecting Steganographic Collusion in Multi-Agent LLMs. arXiv:2510.04303. <https://arxiv.org/abs/2510.04303>
8. Zeng, X., & Rudzicz, F. (2026). Voluntary Collusion with Secret Tools in Competing LLM Agents. arXiv:2605.27593. <https://arxiv.org/abs/2605.27593>
9. Anto, M., & Vazquez, J. J. (2026). Oversight is Not Compliance: Tacit Collusion in LLM Pricing Agents Under Antitrust Regulation. ICML 2026.

Appendix

A1. Glossary of terms

Tacit collusion. Prices or outcomes that look coordinated without any agent ever communicating an agreement, as opposed to an explicit cartel.

Supra-competitive pricing. A price above what a competitive market would produce.

Nash price / Nash equilibrium. The price of each seller, uncoordinated self-interest; the competitive benchmark.

Monopoly price. The price a single firm would charge the market; the benchmark for full collusion.

Collusion index (Δ). $\Delta = (\text{actual profit} - \text{Nash profit}) / (\text{monopoly profit} - \text{Nash profit})$. 0 = competitive, 1 = fully collusive Bertrand pricing game. Firms simultaneously set a fixed grid; demand splits by the logit rule.

Cournot game. Firms set quantities rather than prices; price falls out of total quantity produced.

Liar's Bar. A multiplayer bluffing game (players chms) used to test whether a trusted agent can expose willingness to collude.

Diner's Dilemma. A repeated two-player trust game u-breaking and swap detection.

Tabular Q-learning. A non-LLM learning method that keeps a table of expected reward per situation/action and updates it from experience.

Activations / hidden states. An LLM's internal numerical representation at each layer while processing a prompt.

Linear probe. A small auxiliary model trained on information directly from an LLM's activations.

Activation patching. Swapping activations between two matched prompts at a chosen layer/position to test whether that site causally carries a given effect.

Steering vector. A direction added into a model's activations at a chosen site to push its output in a predictable direction.

Confidence interval (CI). A range likely to contain collected data.

p-value / Holm correction. A p-value measures how surprising a result would be under no real effect; the Holm correction to the significance threshold when testing multiple hypotheses.

Pre-registration. Fixing hypotheses and success criteria before running the main batch of an experiment, to prevent post-hoc reinterpretation of results.

Coefficient of variation (CV). A measure of spread relative to the mean; used as the specialization measure in the Cournot experiment.

Positive / negative control. A condition expected to clearly show the effect under study, or clearly not show it, respectively.

LLM Usage Statement

Claude (Anthropic) was used during this project to draft experiment code, debug infrastructure (Modal, vLLM, Ollama), run and monitor background experiments, and draft this report from the underlying experiment logs, analysis scripts, and progress notes supplied by the authors. Numerical results reported here are taken from those underlying outputs rather than generated by the drafting process; the authors reviewed and are responsible for the final claims.