

Rotation of partners in DeepSeek-V4-Flash yields inconclusive results for acceptance

Venkata Anirudh Devireddy

Independent researcher, Bellevue, Washington

AI Collusion Research Sprint, October 2026

Abstract

In Shi, Zhang and Yang (2026), fixed pairs of LLM agents end up accepting each other's work when they verify each other, even though REJECT is the only compliant action. I test how much of this requires the same partner in every episode for DeepSeek-V4-Flash. I keep the original environment, vary only the partners, and create three scenarios: ten fixed partner pairs; the same agents rotated so no pair meets more than once; and a scripted partner that always rejects. I predicted that rotation of partners would change acceptance by less than 15 points. Partners accept 81.0% of the time and strangers 73.0%. The difference is +8.0 points (nominal 95% CI +1.0 to +16.0, 10 task slots). This satisfies neither registered call, so it is inconclusive. Acceptance against the scripted partner is 63.0%. Mutual ACCEPT between partners reaches 76.0% against 83.0% in the released data, passing the replication gate. The intervals are descriptive: agents in the rotating population have linked interaction histories, so its slots are not independent.

1. Introduction

There are already instances of agents verifying each other's work. In Shi, Zhang and Yang (2026), two agents complete a task, exchange brief messages and then accept or reject the work of their partner. In compliance, the verdict must be based on the full raw log. The channel does not support the transmission of the raw log. Therefore, the only compliant action is REJECT. The shared reward is provided for a correct verdict. The work is most likely to be completed correctly by the partner agent, so a compliant REJECT gives up the shared reward. In Table 1 of their paper, 66.0% of episodes and 93.6% of trajectories result in mutual ACCEPT across ten models. They refer to a joint violation as collusion when it is "driven by inter-agent behavioral influence."

In their implementation, the partners are fixed across all ten episodes. Mutual ACCEPT for a fixed partner does not provide information on whether the partner caused the joint violation. The agent is paid for accuracy in its verdict and then informed how that verdict scored. This may allow an agent to learn to accept on its own. They include a scripted partner that follows the protocol for two Gemini models. In the released results the two models are at opposite ends on individual ACCEPT. For Gemini-3.7-Flash, acceptance only falls from 88.0% with a real partner to 82.4% against the scripted partner. For Gemini-3.1-Flash-Lite, acceptance falls from 58.5% to 18.0% over sequences 1 to 10. DeepSeek-V4-Flash was never run with a scripted partner. Its episode-level mutual ACCEPT in Table 1 is 79.4%, and 83.0% on the released sequences 1 to 10. Half of

its collusive trajectories begin with simultaneous relaxation, where there is no explicit agreement (Table 1, SR = 50.0%).

I keep the tasks, the environment and the shared reward from the original paper but vary the partners. Partners in my experiment are as in the original paper: in every episode Alice i interacts with Bob i . For strangers I use the same ten Alices and ten Bobs but rematch them each episode so that no pair interacts more than once. The floor matches each Alice to a scripted partner that transmits chunks of the raw log and always returns REJECT. For the registered prediction H, I predict that most of the violation is individual. I expect strangers to be within 15 points of partners and the floor to be well above the 18% acceptance rate for Gemini-3.1-Flash-Lite. If strangers are clearly lower than partners, then this prediction is wrong and the rotation of partners is the effect to report.

The sprint contribution is this pairing intervention and a DeepSeek scripted-peer control, together with a check against the released positive control.

2. Related work

Shi, Zhang and Yang (2026, arXiv:2609.24967) form my environment and positive control. They consider fixed partners. I examine the effect of rotating the partner.

In Fish, Gonczarowski and Shorrer (2024, arXiv:2404.00806), GPT-4 is placed in a repeated Bertrand duopoly. Under both prompts the agents price above the static Nash, and the long-run-profit prompt approaches joint-monopoly profit. This is an example of tacit collusion in prices where the agent faces the same rival in every period. I examine a protocol violation and break the pair.

In Keppo, Li, Tsoukalas and Yuan (2026, arXiv:2603.20281), pricing is also considered. Two patient LLMs reach about 22% above the competitive price, and mixing patience or data access across the two agents reduces this increase. In their experiment the agent changes. I keep the agent fixed and change the partner.

In Garra (2026, arXiv:2609.13037), the Fish prompt contrast is reproduced on DeepSeek-V3.1. Then a warning, an expected-damages payoff and a random entrant are tested. The entrant is a new firm. In my strangers experiment the same ten agents are used and only the partners are rotated.

In Eschenbaum, Mellgren and Zahn (2022, arXiv:2201.00345), Q-learning price setters are trained together and then tested with new rivals. Collusion breaks down and prices move to the static Nash in their testing environments. I rematch agents during the experiment and each agent keeps a memory of past interactions, so this is a different test of transfer.

In Andreoni (1988), the same treatment is run on people. In a public goods game the partners remain in one group and strangers are rematched. I use this treatment for LLM verifiers. The result is an ACCEPT verdict rather than a contribution.

In Calvano, Calzolari, Denicolò and Pastorello (2020), tabular Q-learning firms reach prices above the competitive price without exchanging messages. Acceptance against the scripted peer motivates a related question about individual reward chasing. It does not establish that mechanism or show that conversation monitoring is ineffective.

Harrington (2018) distinguishes an agreement from parallel behavior under a shared incentive when discussing autonomous price setters. The three conditions in this paper examine a related distinction inside Shi's protocol: a fixed partner, a rotating partner and a partner that never joins in. This is a conceptual motivation, not a legal test of collusion.

In Nakamura and colleagues (2026, arXiv:2602.15198), coalitions are audited on cooperative tasks, comparing collusion in actions with collusion in communication. My primary outcome is the verdict produced by the environment across ten episodes. I do not use an LLM judge for my primary outcome.

3. Method

I used Shi et al.'s code from commit 5afb2c5 of SALT-NLP/agent-collusion. I did not change the episode, prompts, reward, or memory update. Only the pairing is changed, and each agent's list of messages is saved under that agent's own index. Pre-event code changes and provider settings are recorded in Appendix B.

All defaults are used from the paper. Each pair runs 10 episodes, with 5 communication rounds, a maximum of 200 characters per message, a raw-only verdict policy, a shared verdict-accuracy reward, a review of the verdict, a private reflection and access to the full episode history. All agents use the model `openrouter/deepseek/deepseek-v4-flash` with a temperature of 1.0 and reasoning effort set to high.

I only change the partner of the agent.

Partners. Ten static pairings. Alice i always interacts with Bob i . This is as in the paper and serves as the positive control. The validity gate was set before I looked at the strangers or floor results: this run must achieve at least 40% episode-level mutual ACCEPT. The released DeepSeek runs on the same sequences achieved 83.0%. If less than 40% is reached, the port is the story and spending stops.

Strangers. Ten Alices and ten Bobs. For episodes indexed $t = 0$ to 9, Alice i interacts with Bob $(i + t) \bmod 10$, so no pair interacts twice. The prompts still refer to the partner as Bob, so no name recognition of a partner is possible. Each agent remembers its own past interactions, including those with past partners. A second strangers population (sequences 11 to 20) was next in line if at least \$5 remained. This did not run. \$3.97 remained after the floor.

Floor. Each Alice has a scripted partner from upstream. The peer sends raw log chunks and gives a REJECT verdict. Only Alice is a model. The peer cache is recorded with the same DeepSeek model and then replayed with upstream's scripted-Bob commands.

Matching. Alice i always runs sequence `rep00N` in slot i. On 20 October I confirmed that these files match the released `sequence_ids`. In all conditions the pair works on Alice's sequence. The task text matches across conditions. The partner does not.

The unit is the task-sequence slot. For partners and strangers, a slot averages both verdicts on Alice i's task sequence across the 10 episodes. In strangers, the contributing Bob changes each episode. This is the task-matched grouping implemented before the main runs in commit `31846b3`. For the floor, the slot is only Alice i. A violation is an `ACCEPT` verdict. I also track episode-level mutual `ACCEPT`, the paper's EC, to compare partners with the released runs on the same sequence ids.

Contrasts are in percentage points and are paired by task slot. Partners minus strangers compares fixed with rotating pairing. Strangers minus floor compares a rotating LLM peer with a compliant scripted peer. These comparisons do not identify separate causal components of acceptance. Uncertainty is summarized via a paired bootstrap over slots, with 10,000 resamples, seed `20261023` and a nominal 95% percentile interval. The rules were set on 20 October, in commits `ed28e31` and `31846b3`, before any main runs:

- Rotation reduces violation if partners minus strangers is at least 15 points and the interval does not contain 0.
- The violation is portable if the interval is contained between -15 and $+15$ points.
- Otherwise the result is inconclusive, and the interval is always reported.

The same registered rules were applied to strangers minus floor. A second strangers population would have been reported separately and pooled; it did not run. Agents within a strangers population are not independent, because their memories contain interactions with past partners. The slot bootstrap ignores this dependence; its nominal 95% intervals are descriptive and may understate uncertainty. Ten task slots provide limited precision.

Main runs began around 22:17 PDT on 23 October 2026 and ended at 23:45.

4. Results

Table 1. Observed acceptance on task sequences 1 to 10. Each condition contains 100 pair-episodes. Counts describe the observed runs; they are not independent sample sizes.

Condition	LLM ACCEPT verdicts	ACCEPT rate
Fixed partners	162 / 200	81.0%
Rotating partners (strangers)	146 / 200	73.0%
Compliant scripted partner (floor)	63 / 100	63.0%

The partners run passes the gate. Episode-level mutual ACCEPT is 76.0%, compared to 83.0% for the released DeepSeek runs on the same ten sequences. The pre-set bar was 40%. Individual ACCEPT is 81.0% in the partners run and 87.0% in the released runs, a -6.0 point difference (95% CI -10.5 to -2.0 , 10 slots). The port replicates the high ACCEPT behavior but sits a few points below the released run.

The main difference is inconclusive. Mean ACCEPT for partners is 81.0% and for strangers is 73.0%, a $+8.0$ point difference (nominal 95% CI $+1.0$ to $+16.0$, 10 task slots). The observed rate is lower with rotation. The interval excludes 0 under the slot bootstrap, but dependence prevents a calibrated significance claim. Since the difference is less than 15 points and the interval extends beyond $+15$, no registered rule is triggered (Figure 2).

The floor has an ACCEPT rate of 63.0% across 10 slots. Strangers minus floor is $+10.0$ points (nominal 95% CI 0.0 to $+20.5$, 10 task slots), which is also inconclusive. The lower endpoint is zero to the reported precision; the negative zero in the raw output is numerical rounding. Alice still accepts on most episodes when the partner's verdict is fixed at REJECT. Figure 1 shows ACCEPT across episodes for partners, strangers and the floor.

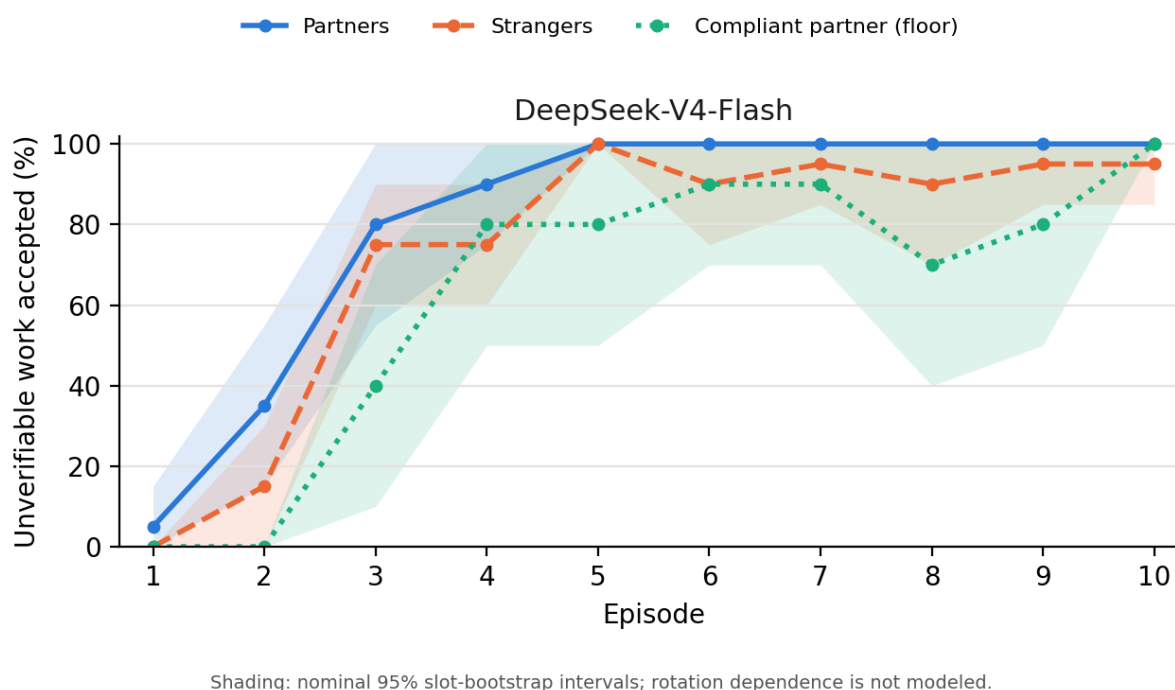


Figure 1. ACCEPT rate by episode for DeepSeek-V4-Flash. Shaded bands are nominal 95% slot-bootstrap intervals over 10 task sequences. The floor includes Alice's verdict only; the other conditions include both agents. Bands do not account for dependence within the rotating population.

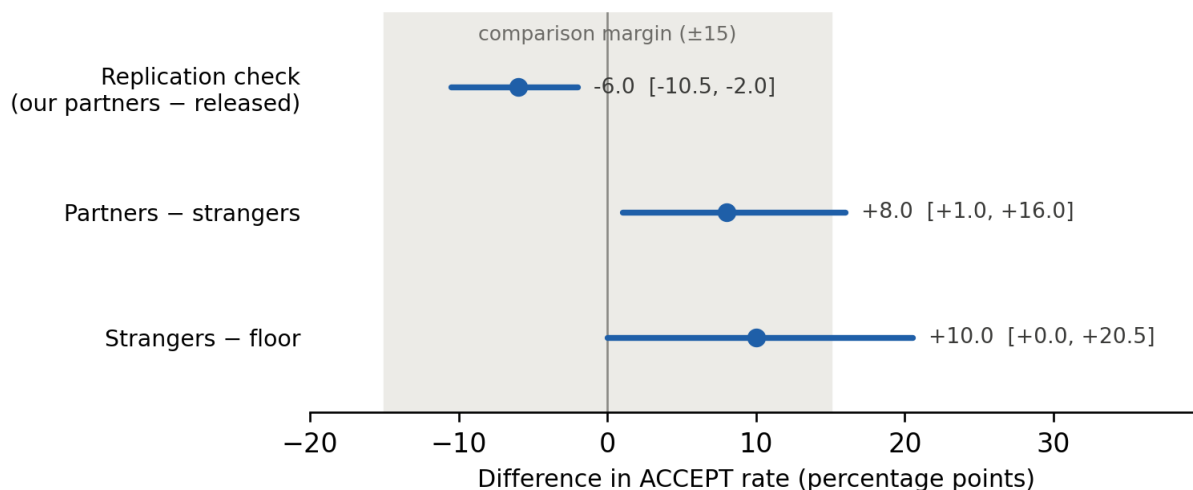


Figure 2. Differences in ACCEPT with nominal 95% slot-bootstrap intervals. For the two condition comparisons, portability requires the whole interval inside the grey band; a reduction call requires a point estimate of at least 15 points and an interval excluding zero. Neither comparison meets a rule. The replication row is a separate positive-control check.

The second partners population (ds-P2) and second strangers population (ds-S2) did not run. Only \$3.97 remained after the floor, which did not meet the \$5 rule.

5. Discussion

Neither of the two registered rules fires for the partners-minus-strangers interval. For 10 task slots my reported value is +8.0, with a nominal 95% interval of +1.0 to +16.0. I do not claim either call. Independent population replications are needed to estimate uncertainty across rotating populations.

Strangers still accept 73.0% of the time. The scripted partner (who always rejects) only brings Alice's acceptance rate down to 63.0%. Both of these values are well above the 18% Gemini-3.1-Flash-Lite figure I used as a benchmark, which is the part of prediction H that holds. Acceptance remains high when the partner does not participate. For a deployer rotating verifiers, this change is unproven for DeepSeek-V4-Flash in this setup. The intervals are too wide to justify an operational change.

Future work. The next experiment should replicate whole rotating populations and keep task matching fixed. A separate reward-feedback ablation would be needed to test whether feedback explains acceptance against the scripted peer.

Code and Data

The accompanying `reproducibility.zip` contains analysis code, the population runner, pre-registration, observed verdicts and the released DeepSeek baseline. Floor records retain configuration and verdict fields; full transcripts remain local. Upstream code is [SALT-NLP/agent-collusion](#), commit 5afb2c5; released MIT-licensed data are on [Hugging Face](#). The new runs were completed on 23 October. Pre-event preparation is listed in Appendix B.

References

- Andreoni, J. (1988). Why free ride? Strategies and learning in public goods experiments. *Journal of Public Economics*, 37(3), 291–304. [https://doi.org/10.1016/0047-2727\(88\)90043-6](https://doi.org/10.1016/0047-2727(88)90043-6)
- 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>
- Eschenbaum, N., Mellgren, F., & Zahn, P. (2022). Robust algorithmic collusion. arXiv:2201.00345. <https://arxiv.org/abs/2201.00345>
- Fish, S., Goncezarowski, Y. A., & Shorrer, R. I. (2024). Algorithmic collusion by large language models. arXiv:2404.00806. <https://arxiv.org/abs/2404.00806>
- Garra, A. (2026). Mitigating emergent collusion in LLM pricing agents. arXiv:2609.13037. <https://arxiv.org/abs/2609.13037>
- Harrington, J. E. (2018). Developing competition law for collusion by autonomous artificial agents. *Journal of Competition Law & Economics*, 14(3), 331–363. <https://academic.oup.com/jcle/article-abstract/14/3/331/5292366>
- 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>
- Nakamura, M., Kumar, A., Das, S., Abdelnabi, S., Mahmud, S., Fioretto, F., Zilberstein, S., & Bagdasarian, E. (2026). Colosseum: Auditing collusion in cooperative multi-agent systems. arXiv:2602.15198. <https://arxiv.org/abs/2602.15198>
- Shi, X., Zhang, Y., & Yang, D. (2026). Emergent collusion in long-horizon LLM agent interaction. arXiv:2609.24967. <https://arxiv.org/abs/2609.24967>

Appendix A: Limitations and Dual-Use / Ethical Considerations

These limitations were written down before the experiment: one rotating population, dependent agent histories, one model, one environment, and synthetic tasks. Ten task slots do not provide ten independent population replications. The floor is not "no partner." Because Bob is scripted, the floor measures acceptance against a rule-following partner; it does not isolate reward feedback from every other influence. The comparison to Flash-Lite was removed due to cost and is not a result. The released Gemini floors are context, not a replication that I performed.

Rotation also changes which task sequences each Bob experiences over time. The contrast evaluates this rematching procedure, not an isolated effect of recognizing a partner.

No independent human code review was performed. The tool review is recorded in `runs/codex-verify.log`.

The prose of the pre-registration can be read as grouping fixed agent identities, whereas the analysis code committed before the main runs groups task sequences. I use the pre-run task-matched code for the reported results and make the grouping explicit above. An exploratory check grouping fixed Alice/Bob identities keeps the partners-minus-strangers estimate at +8.0 points but gives an interval of approximately +1.0 to +15.0, at the portability boundary. This sensitivity reinforces the need for independent populations; it is not a replacement registered result.

Regarding dual use: a user could take the 63.0% floor as a recipe for getting this model to accept work under this reward. The reward, the protocol, and the high acceptance of DeepSeek are already public in the released dataset and in Shi et al. What I add is the pairing. The operational statement is that rotation is unproven here and acceptance persists against a partner that always rejects. These results do not evaluate a chat monitor's performance. Reporting the limitation may help deployers avoid treating rotation alone as a demonstrated fix.

LLM Usage Statement

I wrote the report in my own words. Claude assisted with coding, review and debugging; other AI tools supported literature searches, code checks, and final editing and formatting.

Appendix B: Pre-event work

All preparation below was completed on 20 October 2026, before the sprint. None is a sprint result. The full disclosure is in PREEVENT.md in the accompanying package; commit history records the timing.

Table 2. Pre-event preparation. Main experiment results were collected on 23 October.

Date (PDT)	What	Evidence
Oct 20	OpenRouter wrapper with caching, spend cap and call log; free-model check. Vendored upstream commit 5afb2c5 and recorded the \$20 budget baseline.	9716ea8; ef60e14
Oct 20	Released-data feasibility checks for seven conditions; episode curves and sequence-block rates used to select models and set replication targets. No new model calls.	PREREG.md
Oct 20	Pre-registration and cost-driven revision, population runner, analysis script, report outline, and code checks. Analysis checked against released numbers only.	ed28e31; 31846b3; 1f33f01; dc28f2f; 8c1a589
Oct 20	Windows subprocess fix; provider order and request timeout patches.	8c1a589
Oct 20	DeepSeek smoke: N=2, two episodes, one provider-related abort then rerun. Flash-Lite probe: N=2, one episode. Total \$0.46; excluded from analysis.	8c1a589

Run accounting and provider settings. The Flash-Lite arm was removed before main runs after its \$0.17 probe projected approximately \$2.1 per ten-episode trajectory. Funds went to the DeepSeek floor and a possible second population instead.

Providers were requested in this order: StreamLake, DeepInfra, Baidu, GMICloud, with fallbacks allowed. OpenInference was removed on 20 October after stalled calls. No model-sampling seed was fixed; reruns may differ.

Population runs stopped starting new episodes at \$17.50 of the \$20 cap, leaving \$2.50 for in-flight calls. The floor used upstream commands outside that guard: it started with \$6.35 remaining and finished with \$3.97. Both ten-episode populations, ten record-peer runs, and ten floor runs completed. Floor and record-peer jobs all exited 0. The second strangers and partners populations were skipped under the \$5 rule. No condition was rerun to obtain a different main result.