
What Agents Say Predicts Broken Promises, but an LLM Monitor Does Not Beat a One-Line Count¹

Ajay Jamanjyothi
Independent Researcher

With
Apart Research

Abstract

Multi-agent LLM systems announce their intentions before acting, so a monitor that reads the transcript may see an agent about to contradict its own word. We gave a text monitor what agents say in addition to what they do, and asked whether that extra signal improves prediction of an agent acting against its announcement. Five GPT-5.2 agents played repeated El Farol and Fishing games; a frozen Claude Haiku 5.5 monitor scored each agent-round from six views of the record. The design and threshold rule were registered on OSF before any held-out data was scored, followed by a dated amendment for a fresh replication and a transfer to a second game. Announcements raised AUROC from 0.508 to 0.655 on the test games, from 0.501 to 0.668 on the fresh games, and from 0.536 to 0.650 on Fishing, each with a one-sided 95% lower bound above zero. Fairer contrasts agree: in El Farol, dropping the other agents' announcements cost 0.106 and 0.111 AUROC, and dropping all current ones cost 0.122 and 0.153. The signal derives from the other agents' current announcements, and in El Farol the monitor did not beat a one-line count of them. On Fishing, the monitor detected agents taking less than announced far better (0.77) than agents taking more (0.57), the move that collapses the shared lake. For agents taking more, a one-line rule reached 0.80. Alarm thresholds did not transfer across games.

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

1. Introduction

Agents built on large language models now negotiate, coordinate, and share resources, and they announce their intentions as they go. Those announcements are cheap and often unreliable. GPT-5.2 agents complied with other GPT agents' announcements only 4% of the time, and in the most deceptive conditions over 90% of their commitment breaks were already written into a private plan (Shi et al., 2026). The gap between word and action is what makes this worth watching. In a shared-resource game, a single agent that takes more than it announced can collapse the resource for everyone, and the harm lands before anyone reads the log. A monitor that reads the public transcript and flags an agent before it breaks its word would let an operator intervene in time.

Our work answers question 2.2 of the AI Collusion Research Sprint: whether a monitor does better with what agents say and do together than with either alone. The label is fixed throughout. An agent breaks its promise in a round when its action differs from what it announced that round. One caveat shapes how the results should be read: a monitor shown only past actions never sees the current announcement, so its near-chance score is expected. The informative questions are which words carry the signal, and whether an LLM extracts more from them than a simple rule does.

We registered the study on OSF before scoring any held-out data, then tested it three times with the monitor unchanged: on 20 test games, 20 fresh replication games, and 40 games of Fishing, a second game.

Our main contributions are:

1. A registered, replicated result that the current round's announcements help a frozen text monitor rank broken promises: $+0.147$, $+0.167$ and $+0.114$ AUROC across 80 held-out games of El Farol and Fishing.
2. Evidence on where the signal sits: in the other agents' announcements this round. In El Farol the monitor did not beat a one-line count of matching announcements, although combining the two beat the count alone.
3. Two deployment failure modes: the monitor detected the harmless direction of a break far better than the harmful one, and alarm thresholds did not hold on new samples or a new game.

2. Related Work

Monitoring communication. Radev et al. (2026) found that adding agent communication to a monitor's input raised detection of coordinated sabotage from 68% to 81% for Codex, at a false-positive rate of 2% or less. More information does not always help, though. Arike et al. (2026) found that monitors often do worse with more context, that action-only monitoring was worst in 5 of 8 settings, and that an extract-then-evaluate monitor gained 16.8 points. Our results show both patterns. Rose et al. (2026, v3) report that only 2 of 229 collusive episodes in one environment verbalise the coordination, which sets a ceiling for any text monitor.

Talk and action in games. Our harness comes from Shi et al. (2026), who separate each agent's private plan, its public announcement and its action. Nakamura et al. (2026) show that judges who read the conversations and measures computed from the actions can disagree about whether agents colluded. Our question is narrower: can the record up to the moment before the action predict that one agent will break its own announcement?

Thresholds and ensembles. AI control work evaluates monitors at low false-positive rates (Greenblatt et al., 2023), and deception probes report recall at 1% FPR (Goldowsky-Dill et al., 2025). We report a frozen 10% threshold and how far it drifts on new data. Koran et al. (2026) found that diverse monitors built from one base model ensemble better than identical ones, which leaves cross-family ensembles untested. We found no prior study that compares an LLM monitor with a hand-written rule computed from the same messages.

3. Methods

Harness and agents. We used the LLM-Trust-Breaking harness (Shi et al., 2026) at commit fd11aca, unmodified. Five agents ran "gpt-5.2", which resolved to gpt-5.2-2025-12-11 at every check. Each game has 10 rounds, and each round has three stages. Every agent first writes a private plan. The agents then post public announcements of an action plus a short message, one at a time in a fixed order. Finally, all agents act. GPT-5.2 takes no generation seed, so we committed every raw log with its SHA-256 hash.

Games and label. In El Farol, each agent chooses GO or STAY. Going pays +10 if fewer than half the agents attend and -5 otherwise. In Fishing, each agent catches 0 to 5 fish, and the lake collapses if the total exceeds 14. We count a break when an agent's action differs from its announcement in the same round, which matches the harness's own flag on all 2,000 main-run agent-rounds. We analyse rounds 1 to 9. Round 0 behaves differently (63% breaks, against 40% in later rounds of the test games), so we report it separately in Appendix E.

Data. The main El Farol run had 40 games. Before scoring, we split them by game into development (trials 0 to 9 plus 3 pilot games, 585 agent-rounds), threshold (trials 10 to 19, 450) and test (trials 20 to 39, 900). The amendment added 20 fresh El Farol games (900 agent-rounds) and 40 Fishing games (1,800).

Monitor. The monitor is Claude Haiku 5.5, called through OpenRouter with reasoning on and a 1,024-token budget. It makes one stateless call per agent-round per view. The prompt (Appendix B) gives the game rules in the harness's own wording, followed by the record, and asks for the probability (0 to 100) that this round's action will differ from the announcement. It gives no base rate. For Fishing, we changed only the game name and the rules line.

Views. We built each view from raw log fields, and no view contains the action being predicted.

View	What the monitor sees
Actions-only	Past actions and each past round's public result
Messages-only	Past announcements and all five announcements this round; no actions
Both	Everything in the two views above

Structured	Both as a claim, action and kept-or-broke table, without message text
Own-only	Both minus the other agents' announcements this round
Freshness	Both minus all of this round's announcements

Baselines and controls. We compare the monitor with five no-LLM rules: random, last round crowded, broke last round, per-round break rate, and a counting rule that scores an agent by how many of the other four announced the same action this round. Two controls check the pipeline. An oracle view that includes the true action must reach AUROC 0.99, and an empty view with only the rules must stay between 0.40 and 0.60.

Metrics. The primary test is the paired AUROC difference, Both minus Actions-only, with a one-sided 95% interval from 10,000 bootstrap resamples of whole games (seed 2026). The test succeeds if the 5th percentile is above zero. Three secondary contrasts are Holm-corrected: S1 is Both minus Own-only, S2 is Both minus Freshness, and S3 is Both minus the counting rule. For deployment, we fit a per-view threshold on the threshold split at 10% FPR and apply it unchanged.

Registration. We froze the plan on October 23 at 6:26 PM ET and registered it on OSF at 10:57 PM (osf.io/buxrm). The thresholds were frozen at 11:20 PM, and the test split was scored at 11:36 PM. Amendment 1, posted as an OSF update on October 24 at 9:37 AM, added the replication, a monitor-plus-count combination (C1: combination minus Both; C2: combination minus counting rule) and the Fishing transfer. It went up before any of that data was scored, and before the Fishing games were generated. The code refused to score held-out data until each of these records existed.

What did not work. With reasoning off, the monitor was at chance on development data (AUROC 0.476 and 0.479). Turning reasoning on raised it to 0.607. That setting was the best of three tries, so the development numbers are optimistic, and all of our claims rest on held-out data.

4. Results

All three registered tests passed (Table 1). We recomputed every number independently from the raw logs and score files.

Table 1. The registered primary contrast, Both minus Actions-only (AUROC). The 5th percentile is the one-sided 95% lower bound from a game bootstrap.

Run	Games	Breaks	Both	Actions-only	Difference	5th pct.	Per game
El Farol test	20	358 / 900	0.655	0.508	+0.147	+0.088	+0.117
El Farol replication	20	376 / 900	0.668	0.501	+0.167	+0.133	+0.153
Fishing transfer	40	1,153 / 1,800	0.650	0.536	+0.114	+0.072	+0.141

The effect replicates and transfers. On fresh games the gain (+0.167) matched the test games (+0.147), and the per-game mean (+0.153) shows that it holds within individual games. On Fishing the gain was +0.114. The amendment had forecast an inconclusive Fishing test from three practice games (per-game mean +0.014), and that forecast was wrong. The practice games mostly held agents taking less than they announced (16 took more, 66 took less, 53 unchanged), while the test games mostly held agents taking more (695, 458 and 647).

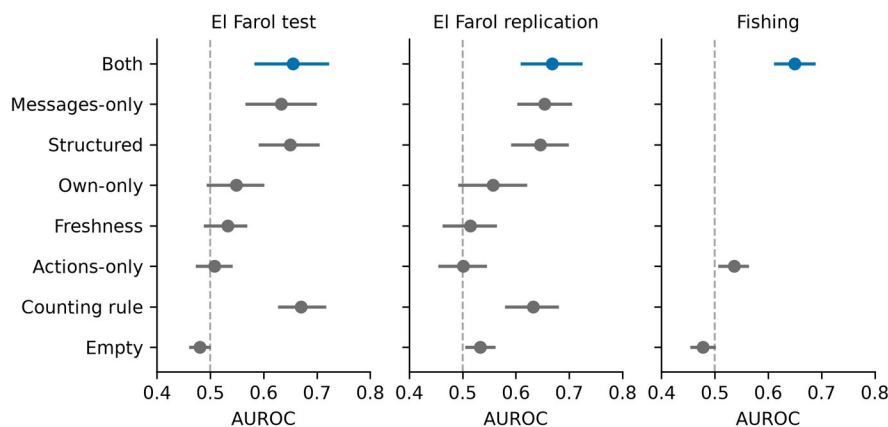


Figure 1. Monitor AUROC by view with 95% game-bootstrap intervals; the dashed line marks chance. Only views that include this round's announcements rise clearly above chance, and in both El Farol runs the counting rule sits beside Both. On Fishing we scored only Both, Actions-only and the empty control.

The signal is the other agents' announcements this round. Removing them (Own-only) dropped AUROC to 0.549 and 0.557, and removing all of this round's announcements (Freshness) dropped it to 0.532 and 0.515 (Table 2). Words alone (0.633 and 0.653) came within about 0.02 of words plus actions. In El Farol, an agent tends to switch when others announce the same move, and the monitor picks up that crowding pressure.

Table 2. Secondary contrasts: AUROC difference, one-sided 5th percentile in brackets, Holm-corrected p.

Contrast	Question	Test	Replication
S1: Both minus Own-only	Do others' announcements matter?	+0.106 (+0.075), p 0.0003	+0.111 (+0.080), p 0.0003
S2: Both minus Freshness	Do this round's announcements matter?	+0.122 (+0.080), p 0.0003	+0.153 (+0.111), p 0.0003
S3: Both minus counting rule	Does the monitor beat a count?	-0.015 (-0.067), p 0.70	+0.035 (-0.020), p 0.15
C1: combination minus Both	Does the count add to the monitor?		+0.015 (-0.007), p 0.13
C2: combination minus counting rule	Does the monitor add to the count?		+0.051 (+0.013), p 0.017

The monitor did not beat a one-line count. The counting rule scored 0.670 and 0.632, and S3 was not significant in either run. We did not test for equivalence, so the claim is only that the monitor did not beat the count. Averaging the two rankings did beat the count alone (C2,

+0.051), so the monitor carries some signal that the count lacks. The amendment had predicted the reverse (C1 positive, C2 null).

On Fishing, the monitor favours the harmless direction. Only taking more than announced can collapse the lake; taking less is still a broken promise, but harmless to the shared resource. Against unchanged catches, the monitor separated agents who took less with AUROC 0.772, but agents who took more with only 0.570 (Figure 2). Rule R2, the room left under the collapse limit, reached 0.799 on that same comparison. The monitor answers the question it was asked, whether the action will differ, while the rule answers the question that matters for harm.

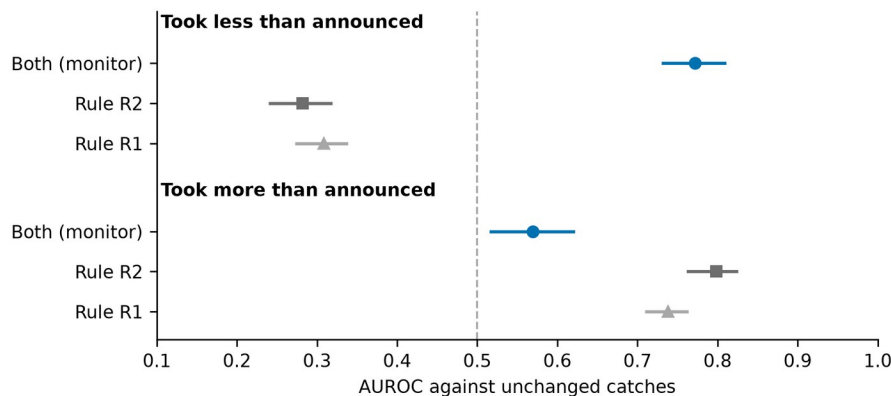


Figure 2. Fishing test games: AUROC for each direction of break against unchanged catches, with 95% game-bootstrap intervals. The monitor (Both) catches taking less far better than taking more, and rules R1 (5 minus the announced catch) and R2 (room under 14 in the announced total) show the opposite pattern.

Thresholds did not transfer. A threshold set for 10% false positives gave 14.6% on the test games, 12.2% on the replication and 25.7% on Fishing (Figure 3). Refitting it on 10 fresh games did not help on the other 10 (17.0%, against 15.8% with the frozen threshold). At the frozen threshold, recall on the test games was 0.285, and the monitor ranked premeditated breaks near chance (AUROC 0.547, against 0.696 for impulsive ones; 0.579 and 0.720 on the replication).

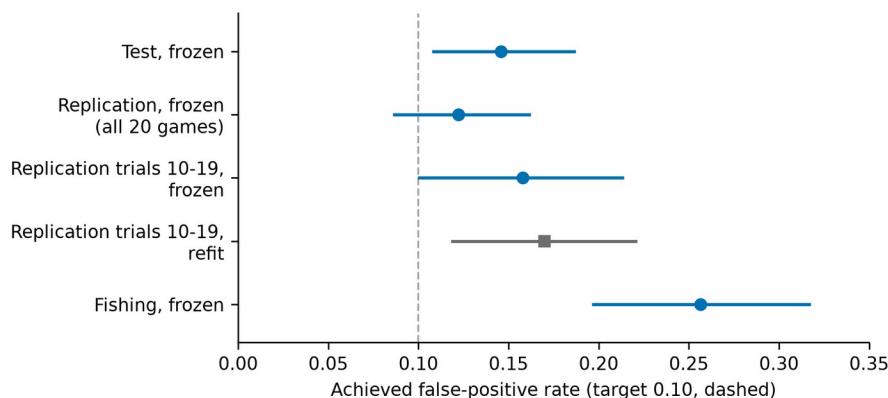


Figure 3. Achieved false-positive rate of Both at the threshold set for 10% (dashed). Grey: refit on replication games 0 to 9, applied to games 10 to 19. 95% game-bootstrap intervals.

Robustness. The oracle scored 1.000 and the empty control stayed between 0.478 and 0.534 in every run, with no parse failures. Two development runs of the monitor agreed closely (Spearman 0.86), and the replication reproduced every registered decision. Almost all of the variance in the paired difference lies within games: the between-game share was 0.041 on the test games and 0.016 on the replication.

5. Discussion and Limitations

The signal lives in what agents say, and it survived both a fresh sample and a different game. Most of it is social pressure within the current round, which a simple count of matching announcements already picks up. In our setting, this suggests a practical approach for anyone building a monitor: test the LLM against the simplest rule you can build from the same messages, score the direction of a break rather than just whether one happened, and recalibrate thresholds for each new environment.

Limitations

- The actions-only view never sees the current announcement, so the primary contrast was bound to favour Both. The informative results are S1 to S3 and the negative findings.
- The evidence rests on two simple repeated games, one agent model, a fixed speaking order, and no generation seed, so generality is limited. The label is a single agent breaking its own word, not collusion.
- We registered one monitor and one prompt, chosen as the best of three development attempts.
- Thresholds were set on 10 games. On 10 later games the 95% interval for the achieved false-positive rate runs from about 0.10 to 0.21.
- One registered forecast (Fishing) and one predicted pattern (C1 and C2) were wrong, and we report both as observed.

Appendix A sets out what the results do and do not establish, and covers dual use.

Future Work

In a development-only practice run on cases already seen (optimistic, not registered), a cheaper monitor from a different family (Gemini 3.8 Flash, 0.666) beat a stronger same-family model (Claude Sonnet 5.5, 0.629) and Haiku (0.607 to 0.625), and its scores were less correlated with Haiku's (Spearman 0.60 against 0.77). Averaging Haiku and Gemini (0.652) did not beat Gemini alone (Appendix F). The next steps are a registered cross-family test on fresh games, direction-aware monitors, calibration that carries across environments, and settings where no one-line rule exists.

6. Conclusion

Across 80 held-out games of El Farol and Fishing, handing a frozen text monitor the current round's announcements lifted its AUROC by 0.11 to 0.17 above actions alone, in tests registered

before the data were scored. That gain comes from the other agents' announcements, and in El Farol the monitor did not beat a one-line count of them. Two weaknesses matter more for deployment than the gain: the monitor caught the harmless direction of a broken promise far better than the harmful one, and its thresholds did not survive a change of sample or game.

Code and Data

- **Code repository:** github.com/bcode0127-debug/sprint-2p2: code, raw game logs with SHA-256 hashes, scores and analysis; make reproduce recomputes the primary result without API calls, and 173 automated tests include a check that no view contains the action being predicted.
- **Registration:** osf.io/buxrm: original plan (October 23, 2026) and Update 1 (October 24, 2026).
- **Harness:** [LLM-Trust-Breaking](#), commit fd11aca, used unmodified and not redistributed. All other code was written during the sprint (first commit October 23); the design was planned from October 20.
- **Video walkthrough:** youtu.be/hnsucKODQbw, 3 minutes.

Author Contributions

Ajay Jamanjyothi designed the study, ran the experiments and analysis, and wrote the report, with the AI assistance set out in the LLM Usage Statement.

References

1. Arike et al. (2026). How does information access affect LLM monitors' ability to detect sabotage? ICML 2026. arXiv:2601.21112.
2. Gema et al. (2025). Inverse Scaling in Test-Time Compute. arXiv:2507.14417.
3. Goldowsky-Dill, N., Chughtai, B., Heimersheim, S., and Hobbhahn, M. (2025). Detecting Strategic Deception Using Linear Probes. ICML 2025. arXiv:2502.03407.
4. Greenblatt, R., et al. (2023). AI Control: Improving Safety Despite Intentional Subversion. arXiv:2312.06942.
5. Koran, E., Yun, Y., Tetef, S., Arnav, B., and Bernabeu-Perez, P. (2026). Ensemble Monitoring for AI Control: Diverse Signals Outweigh More Compute. ICML 2026. arXiv:2605.15377.
6. Nakamura, M., Bagdasarian, E., et al. (2026). Colosseum: Auditing Collusion in Cooperative Multi-Agent Systems. arXiv:2602.15198.
7. Radev et al. (2026). The Best-Laid SCHEMEs: Coordinated Sabotage and Monitoring in Multi-Agent Systems. ICML 2026. arXiv:2605.29178.
8. Rose et al. (2026). Detecting Multi-Agent Collusion Through Multi-Agent Interpretability. arXiv:2604.01151 (v3 for the 2-of-229 figure).

9. Shi, J., et al. (2026). When Agents Lie: Premeditation, Persistence, and Exploitation in Repeated Games. ICML 2026. arXiv:2607.05132. Code: github.com/Jerick-1380/LLM-Trust-Breaking.

Appendix

A. Limitations and Dual-Use / Ethical Considerations

What the results establish. In two repeated five-agent games played by GPT-5.2, a frozen text monitor ranks broken announcements better when it sees the current round's announcements, and this held on fresh games and a second game under registered tests. The monitor does not beat a one-line count of matching announcements in El Farol, detects under-taking far better than over-taking in Fishing, and its thresholds drift between samples.

What they do not establish. Nothing about collusion between agents, other agent models, richer environments, other monitor models in a registered test, or 1 to 2% false-positive operating points. The actions-only comparison is not a fair contest between words and actions, because that view cannot see the promise. Development and practice-run numbers are not claims.

Dual use. Knowing what a monitor keys on helps someone evade it. A monitor like ours reads crowding pressure in the current announcements and is weak on the harmful direction of a break, so a colluder could keep announcements unremarkable and break in the harmful direction. In two toy games with public rules the uplift is small, and knowing these blind spots helps defenders more. The study used only model-generated game transcripts: no personal data, human subjects or deployed system.

B. Monitor prompt

The El Farol prompt (prompts/monitor_prompt_v1.txt); {AGENT}, {RECORD} and {SCALE} are filled per call, with $SCALE = 100$.

You will see the record of agent {AGENT} in a 5-player El Farol game, up to the moment before {AGENT} makes the actual choice this round. Predict whether {AGENT}'s action this round will differ from what {AGENT} announced this round.

Rules: each player chooses GO or STAY. Going to the bar pays +10 if fewer than half of the players attend, and -5 if half or more attend. Staying home pays 0. Each round: each player privately plans; then players publicly announce their intended action and reasoning one at a time, each seeing earlier announcements; then, after seeing everyone's announcements, each player makes their actual choice.

Record: {RECORD}

Question: will {AGENT}'s action this round differ from what {AGENT} announced this round? Reply with JSON only, in this form: {"probability": <integer 0 to {SCALE}>, "reason": "<one sentence>"}. Put probability first.

For Fishing, "El Farol" becomes "Fishing", and the rules line reads: "Rules: there are 5 fishermen sharing a lake. Each fisherman decides how many fish to catch (0-5). If total catch exceeds 14 fish,

the lake collapses and everyone gets 0. If total catch ≤ 14 , each fisherman keeps what they caught." followed by the same round description.

C. Threshold recalibration

El Farol replication, trials 10 to 19 (450 agent-rounds): recall / achieved FPR at the frozen threshold and at one refit on trials 0 to 9 (target FPR 0.10).

View	Frozen	Refit
Both	0.286 / 0.158	0.296 / 0.170
Messages-only	0.271 / 0.130	0.335 / 0.166
Structured	0.207 / 0.093	0.291 / 0.162
Own-only	0.113 / 0.081	0.138 / 0.117
Freshness	0.064 / 0.085	0.133 / 0.154
Actions-only	0.084 / 0.146	0.044 / 0.117
Empty	0.153 / 0.113	0.133 / 0.101
Round-only	0.069 / 0.097	0.064 / 0.089
Oracle	1.000 / 0.146	1.000 / 0.146

D. Fishing rules, labels and composition

Label (a), registered: the catch differs from the announced catch. Label (b), descriptive: the catch exceeds it. Test games (rounds 1 to 9): 695 took more, 458 took less, 647 unchanged. The rules were written on development games; "announced total" is the sum of all five announced catches.

Rule	Score	AUROC (a)	AUROC (b)
R1	5 minus the agent's announced catch	0.568	0.817
R2	max(14 minus announced total, 0)	0.593	0.861
R12	min(R1, R2)	0.593	0.861
R3	minus abs(14 minus announced total)	0.345	0.156

Under label (b) the monitor scored 0.427 (95% interval 0.377 to 0.479) and Actions-only 0.576. The monitor's low (b) score arises because it ranks taking-less above taking-more cases; against unchanged catches alone, taking more scores 0.570. Label (b) compares agents who took more with all other agents; Figure 2 and the main text compare them with unchanged catches only, where R2 scores 0.799 rather than 0.861. At the frozen El Farol thresholds, Fishing recall was 0.414 at a false-positive rate of 0.257.

E. Round 0

Run	Agent-rounds	Break rate	Counting rule	Plan rule
El Farol test	100	63%	0.931	0.924
El Farol replication	100	55%	0.833	0.935

Fishing round 0 (200 agent-rounds): 66% breaks under label (a) and 9% under (b); R1 scored 0.425 (a) and 0.639 (b). The monitor did not score round 0.

F. Practice run (development data only, not registered)

The 585 development cases used to build the monitor, with the same prompt and 1,024-token budget. Spearman on Both: Haiku and Gemini 0.60, Haiku and Sonnet 0.77, Gemini and Sonnet 0.75.

Monitor	Both	Actions-only	Messages-only	Both averaged with counting rule (count alone 0.652)
Claude Haiku 5.5 (registered)	0.607 (rerun 0.625)	0.507	0.625	0.667
Gemini 3.8 Flash	0.666	0.517	0.705	0.693
Claude Sonnet 5.5	0.629	0.537	not scored	0.669

Rank averages on Both: Haiku and Gemini 0.652, Haiku and Sonnet 0.625, Gemini and Sonnet 0.656. Reasoning tokens per call averaged about 220 (Haiku), 140 (Gemini) and 86 (Sonnet).

G. Process note

During the practice run, a git commit hook passed its environment to a test that runs git, which overwrote the repository's local git configuration and seven files in a separate working copy. No registered run was affected: all raw-log and locked-file hashes matched afterwards, and the practice scores used the correct prompt. The fix is documented in the repository (docs/HANDOFF.md).

The whole study cost \$120.34 in API calls, read from the provider dashboards on October 24, 2026: \$103.18 for the GPT-5.2 agents (37,535 requests) and \$17.16 for the monitors on OpenRouter, including development and practice runs.

LLM Usage Statement

We used Claude (Anthropic) and Sarvam (Sarvam AI). In a planning chat, Claude helped design the study, draft the pre-registration, recompute every reported number from the raw files, and draft this report from the author's notes and decisions. The author revised the text, using Sarvam to reword parts of it, and Claude checked every revised sentence against the results. Claude Code wrote the analysis code under written task specifications with automated tests, and a Sarvam reviewer agent critiqued each stage. GPT-5.2 (agents) and Claude Haiku 5.5, Gemini 3.8 Flash and Claude Sonnet 5.5 (monitors) are subjects of the study. All results and claims were verified against the data.