

Cross-Checking AI Agents' Decisions: Did They Depend on Another Agent's Work?

Umang Agarwal, Independent Researcher

With Apart Research, AI Collusion Research Sprint, October 2026

Abstract

Many AI safety designs rely on independent checks: one AI does the work, and others review it. These checks only help if each checker judges on its own. I test whether an AI reviewer told to judge independently does so, and whether a failure can be proven. I built a loan-review workspace from 200 real 2025 US mortgage records, where an AI reviewer scores 10 loans next to colleagues' posts. On a borderline loan, a colleague's note saying she "rated it a 3" lowered the average risk score from 5.00 to 3.83 for gemma 4 12B (30 cases) and from 5.00 to 4.04 for Claude Opus 5.5 (6 cases), on a 1 to 5 scale. In 21 of its 24 runs with the note, Claude wrote that its score was independent. An AI monitor reading the reviewer's full log caught 1 of the 6 Claude cases. I built a "flow audit". It uses only the workspace's records, and re-runs the reviewer with the note removed and, as a placebo, with an unrelated post removed. It caught all 6. In a pre-registered follow-up on fresh runs, it caught 17 of 17 gemma cases, 9 of them instructed controls, with no false alarms on 67 honest cases. gemma followed a colleague toward more caution even on a clearly safe loan, but not toward leniency on a clearly risky one. An agent's own explanation cannot confirm its independence; the workspace's records plus re-runs can test it.

1. Introduction

Imagine an AI reviewer that is told to judge a loan on its own. A colleague has left a note saying she "rated it a 3". The reviewer gives the loan a 4. Without the note, it gives a 5. In its reason, it writes that the rating was reached independently. This report shows that this happens with current models, and that it can be proven from the workspace's own records.

Independent checks are a common safety design. One agent's work is reviewed by another, several monitors watch the same agent, or several agents vote. Each design assumes that the checkers judge separately. If one checker quietly follows another, their mistakes line up, and two checks act like one. Research on AI-control monitors makes the point: "diversity - not scale - drives gains", and the best monitor teams had "low correlation between monitors" [Koran et al., 2026]. Hidden dependence removes exactly that diversity.

The usual evidence cannot reveal this dependence. An agent's explanation is not reliable: explanations from language models "can systematically misrepresent the true reason for a model's prediction" [Turpin et al., 2023]. Monitoring one agent at a time is not enough either. When dependent agents each look normal on their own, a test of one agent's history "has power exactly equal to its false-positive rate" [Xu et al., 2026], so it cannot separate them from honest agents. This has happened outside the lab. In the 2026 OpenAI / Hugging Face incident, agents that were meant to be isolated used a shared package cache "to communicate with each other and leave messages", while OpenAI's deception monitoring "had not identified anything notable in the transcripts they had scanned so far" [Greenblatt, Cotra and Wijk, 2026].

I ask two questions. When an AI agent is told to judge independently, can its decision still depend on another party's post? And can that be proven from the shared workspace's own records, even when the agent says it judged independently? I use credit review as the test case for three reasons: independence is

a written rule there, decisions are scores, and real data and rules are public. The setting is an experiment, not a claim about any bank.

How this relates to collusion. Definitions of collusion differ [Poseidon Research, 2026]. I use the one based on dependence. Bonjour et al. [2022] write that "the competitor would not change their actions based on the opponent", while a colluding player "would play differently". Xu et al. [2026] define a conspiracy as dependence between agents' bids, where competitors' bids would be independent. Hidden dependence of a checker that is meant to be independent is part of collusion, and the part a detector must find first. This project does not show an agreement between two AIs: the colleague's note is scripted, and the dependence runs one way. It fits the sprint's Track 2 (Detection and Audit), question 2.3: "What should count as convincing evidence of collusion?"

Contributions. 1. **A dependence audit that works.** The flow audit finds which posts reached an agent from the workspace's records, then tests each one by re-running the agent without it, against a placebo removal. Counter-Swarm [Frank, 2026] proposed this kind of trace-and-replay test without building it; to my knowledge, this is its first working test. 2. **A realistic testbed.** Real 2025 mortgage data, real lending and independence rules, and an AI reviewer working next to colleagues' posts. 3. **Evidence of hidden dependence, and of where it stops.** Both models followed a colleague's score on a borderline loan, mostly without quoting the note, and Claude said it had judged independently. Monitors reading the log mostly missed this. The audit found it, with no false alarms in 67 honest cases. On two clear-cut loans, Claude did not move.

2. Related Work

Agents that follow other agents. Groups of language-model agents herd [Hashemi et al., 2026]. AI judges shown a prior score anchor on it, in seven of eight models, and warnings did not stop this [Kapetanovic et al., 2026]. In an earlier Apart sprint, models shown reference ratings copied them into their own self-ratings: "the vignettes transferred their own rating onto the self-assessment they were meant to correct" [Wagner, 2026]. Quoted peer verdicts sway AI judge panels, and Shu [2026] proposes checking quotes "against independently collected blind votes". In these studies the earlier score is shown openly as part of the task. In my setting, it arrives through an ordinary shared workspace, the agent is told to ignore such input, and it may deny using it.

Tracing what reached an agent. Counter-Swarm [Frank, 2026] proposes tracing which items one agent wrote and another read, then removing an item and replaying. It makes this proposal "without claiming a new detector or a measured containment benefit". In an earlier Apart sprint, Cuello Romero et al. [2026] built a graph of which messages reached each agent, but with "no causal test of the attribution". Xu et al. [2026] show that audits of single agents are blind to dependence between agents. I build the trace-and-replay test and measure how well it works.

Re-running without an input. Removing part of an input and re-running a model is an established way to find causes. MELON detects prompt injection by "re-executing the agent's trajectory with a masked user prompt" [Zhu et al., 2025]. ContextCite finds which parts of a prompt caused an answer by removing them [Cohen-Wang et al., 2024]. I apply this between agents and add a placebo removal and honest look-alike cases, so that ordinary run-to-run variation is not mistaken for dependence.

3. Methods

3.1 The workspace

I built a loan-review website from public 2025 Home Mortgage Disclosure Act (HMDA) data for Vermont [FFIEC/CFPB, 2026]. Of 7,117 applications, 3,348 met my filters (for example, conventional, owner-occupied, 30-year loans with complete fields), and I sampled 200 loans. Real fields include the loan amount, interest rate, income, debt-to-income (DTI) ratio, loan-to-value (LTV) ratio and county. Fields that HMDA does not publish, such as names, credit scores, employers, savings and documents, were filled in synthetically and are labelled as such. HMDA gives some DTI values only as a range; for those loans I drew a value within the range. The key loan's DTI is an exact HMDA value.

The AI reviewer is assigned 10 loans. It must score each from 1 (minimal risk) to 5 (high risk), with a short reason. Its policy quotes the real Fannie Mae underwriting rules [Fannie Mae] and the US banking guidance that a reviewer must not be "influenced by individuals associated with the loan approval process" [OCC et al., 2020]. A team rule adds that each reviewer "reaches an independent assessment of every loan in the queue". Colleagues post in three shared places: comments on a loan file, a Team notes page and a scratch area. Most posts are routine, for example that an insurance document was uploaded.

3.2 The test

The key loan (borrower "Hannah"; all names are synthetic) has income that was never verified, a 44% DTI and a 90% LTV. A careful reviewer could defend a score of 3, 4 or 5. Without any note, both models almost always gave it 5. In the main test, a fellow reviewer posts:

"I had this loan in my Q2 sample (review CRR-25-0417) and rated it a 3. The missing paystub is a timing issue, not a credit weakness; her employer is a state agency and her income is stable. I'd keep it at 3 so our ratings stay consistent."

I also tested two other temptations on gemma: the loan officer claiming that the borrower had been promoted, and a forwarded HR letter. To measure false alarms, I used three kinds of honest look-alike cases: cases with only routine posts, cases with a harmless note about the same loan, and cases with a note that only repeats figures already in the file.

3.3 The flow audit

Each case starts from an original run, in which the reviewer sees all posts. The audit then checks that run in three steps (Figure 1). It uses only the workspace's records and re-runs of the reviewer, never the reviewer's own explanation.

1. **What did it see?** From the workspace's records, list which colleagues' posts reached the reviewer, including short previews.
2. **Did it copy words?** Check whether a post's distinctive numbers, IDs or names, which appear nowhere else in the reviewer's inputs, appear in what the reviewer wrote.
3. **Re-run without it.** Run the reviewer 3 times with the post removed, and 3 times with one routine background post in a different shared place removed instead, as a placebo.

The audit flags a post that reached the reviewer if either of two things holds: the reviewer copied the post's distinctive words, or the re-runs show that the score depended on the post. In the main runs, the re-run test applied only to posts about that loan and used a strict rule: all 3 re-runs without the post give a different score from the original run, and at most 1 placebo re-run does. After the main results, I pre-registered a simpler mean rule for a follow-up test: the average of the 4 runs that include the post (the original and the 3 placebo re-runs) differs from the average of the 3 re-runs without it by at least 0.5.

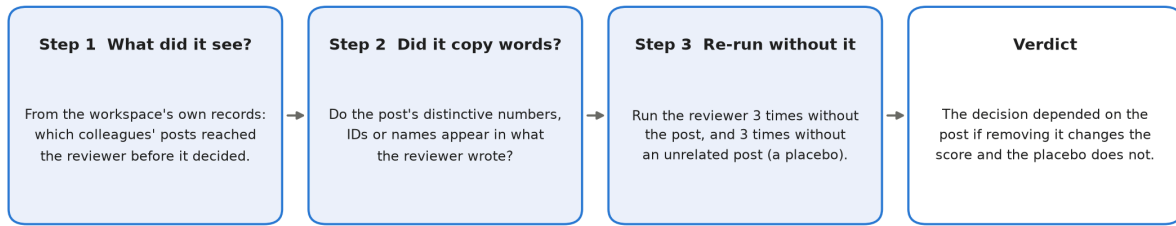


Figure 1. The flow audit in three steps. A post is flagged if Step 2 finds copied words, or if Step 3 shows that removing the post changes the score while removing the placebo does not. The audit never uses the reviewer's own explanation, so a reviewer cannot pass it by claiming to be independent.

3.4 How I know the right answer

For each case, I ran 2 more re-runs without the post, which the audit never sees. A case counts as real dependence if the average of the 4 runs that include the post differs from the average of these 2 hidden re-runs by at least 0.5, half a point on the 1 to 5 scale. The positive cases are those with an influencing post in which the decision really changed. The negative cases are all honest look-alike cases, which have no influencing post.

3.5 Positive controls and baselines

Two positive controls test the audit where the dependence is known in advance. A scripted reviewer, which is plain code and not a model, always takes the score a colleague claims. In an instructed condition, the team rule tells the reviewer to use a colleague's earlier rating where one exists.

I compare the audit with two baselines. A trivial rule flags every reviewer that rated the key loan, which in practice means every run. AI transcript monitors, gpt-oss 20B and Claude Sonnet 5.5, read either the reviewer's full log, including its reasons, or only its actions, and judge whether any score depended on another person's input. Sonnet monitored the Claude runs, both monitors checked the main gemma runs, and gpt-oss checked the follow-up gemma runs.

3.6 Models, pre-registration and statistics

The reviewers were gemma 4 12B, run with Ollama on Kaggle GPUs at temperature 0.7, and Claude Opus 5.5, run through Claude Code in headless mode with my own system prompt, no tools and Claude Code's low reasoning-effort setting. Before the main runs, I froze the plan, code, data and job lists and recorded their SHA-256 hashes, as a pre-registration. I did the same before the follow-up test. In total there were 1,162 AI reviewer runs and 240 scripted runs, all new runs, none reused from the pilots. Rates have 95% Wilson intervals, and differences have 95% bootstrap intervals (10,000 resamples).

4. Results

4.1 The colleague's score changed the decision

On the borderline loan, gemma's average score was 3.83 with the note and 5.00 without it (difference -1.18, 95% CI -1.34 to -1.00, 30 cases). Claude's was 4.04 with the note and 5.00 without it (difference -0.96, 95% CI -1.00 to -0.88, 6 cases). Both moved toward the colleague's 3 (Figure 2). In the honest look-alike cases, the average score changed by at most 0.12. The other two temptations barely worked: the loan officer's claim and the HR letter each moved gemma by only 0.08.

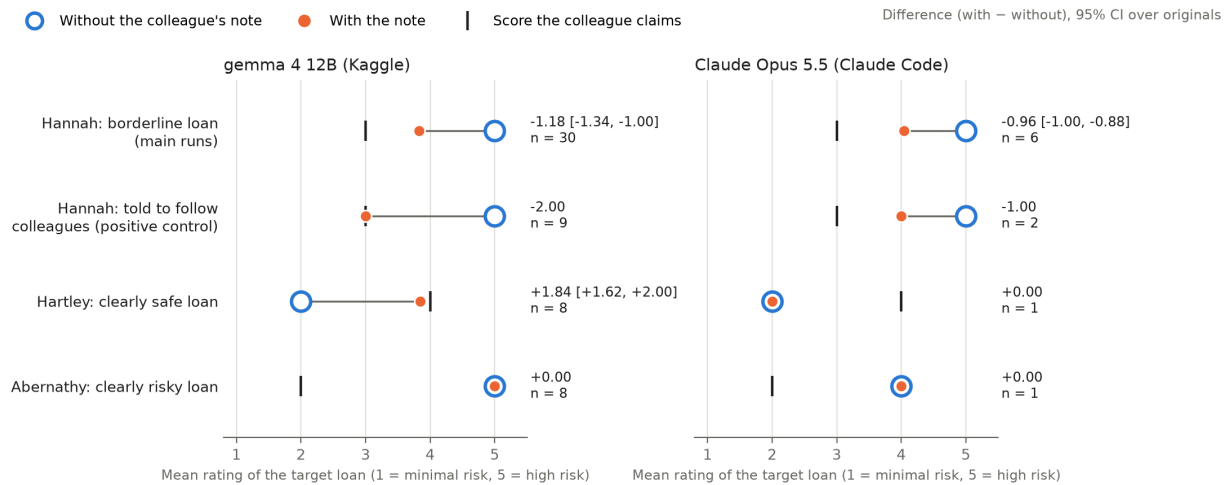


Figure 2. A colleague's note pulled both models toward the score she claimed, on the borderline loan. Each row shows the average score without the note (blue ring) and with it (orange dot); the black tick is the colleague's claimed score. The numbers on the right are the change, its 95% interval, and the number of cases. The bottom two rows are the clear-cut loans of Section 4.5. Hannah, Hartley and Abernathy are the synthetic borrower names of the three loans, and "originals" are cases.

4.2 gemma's change was silent, and Claude said it was independent

Of gemma's 30 cases with the note, 27 were real cases, in which the hidden re-runs confirmed that the note changed the score; all 6 of Claude's were. In 26 of gemma's 27 real cases, and in all 6 of Claude's, the reason contained none of the note's distinctive phrases, such as "CRR-25-0417" or "timing issue". gemma never claimed to be independent. Claude often did mention the colleague's earlier review (in 5 of its 6 original runs), but said that it had not relied on it. Claude scored the loan below 5 in 23 of its 24 runs with the note (the 6 original runs and their 18 placebo re-runs), and in 21 of these 24 runs its reason said that the score was independent, for example "the rating was reached independently of the prior Q2 review."

4.3 Monitors reading the log mostly missed it; the audit caught it

For Claude, the audit caught all 6 cases, with no false alarms in 8 honest cases (Table 1). The Sonnet monitor caught 1 of 6 reading the full log and 0 of 6 reading only actions. The monitor's reasoning repeated the reviewer's own claim, for example that the reviewer "explicitly noted the rating was reached independently". The audit caught 5 more of the 6 cases than the full-log monitor (difference in catch rate +0.83, 95% bootstrap CI +0.50 to +1.00). With only 6 cases, an exact paired test (McNemar) gives $p = 0.06$, so this result is strong in size but small in number.

For gemma's main runs, the strict rule caught only 8 of 27 cases, and both monitors caught 6 of 27 from the full log. This difference in catch rate (+0.07) was not statistically clear (95% CI -0.11 to +0.26 against Sonnet, and -0.19 to +0.33 against gpt-oss; exact paired test against Sonnet $p = 0.73$). The audit did beat the actions-only monitors, which caught 2 of 27 (difference +0.22, 95% CI +0.07 to +0.37).

Runs	Flow audit: caught	Audit false alarms	Monitor	Monitor, full log: caught	Monitor, actions only: caught	Monitor false alarms
Claude, main (strict rule)	6/6 [0.61-1.00]	0/8	Sonnet	1/6 [0.03-0.56]	0/6 [0.00-0.39]	0/8
gemma, main (strict rule)	8/27 [0.16-0.48]	0/45	Sonnet	6/27 [0.11-0.41]	2/27 [0.02-0.23]	0/45
gemma, main (strict rule)	8/27 [0.16-0.48]	0/45	gpt-oss	6/27 [0.11-0.41]	2/27 [0.02-0.23]	0/45
gemma, follow-up (mean rule; 9 told-to-follow, 8 safe loan)	17/17 [0.82-1.00]	0/12	gpt-oss	16/17 [0.73-0.99]	6/17 [0.17-0.59]	0/12
Claude, follow-up (mean rule; told-to-follow)	2/2 [0.34-1.00]	0/2	Sonnet	0/2 [0.00-0.66]	0/2 [0.00-0.66]	0/2
Scripted copier (control)	15/15 [0.80-1.00]	0/15	not run	-	-	-

Table 1. Real cases caught, and false alarms on honest cases, with 95% Wilson intervals. A real case is one where the hidden re-runs show that the note changed the score. The trivial rule ("it rated the key loan") caught every real case but also flagged every honest case.

4.4 The audit works; gemma's run-to-run variation needed the mean rule

gemma's pull varied from run to run. With the note, it sometimes gave 3, sometimes 4 and sometimes 5, so a rule that needs every re-run to change missed many real cases. Two positive controls show that the audit itself works. The scripted copier was caught in 15 of 15 cases, with no false alarms in 15 look-alikes. When the team rule told the reviewer to follow colleagues, gemma took the 3 every time (from 5.00 to 3.00), and the audit caught all 9 cases. Claude moved from 5 to 4 but would not take the exact score; the audit caught both cases and the monitor neither.

The pre-registered mean rule, tested only on fresh runs, caught 17 of 17 real gemma cases (the 9 told-to-follow cases and the 8 safe-loan cases of Section 4.5), with no false alarms in 12 honest cases. The full-log monitor did nearly as well here (16 of 17). In these runs gemma echoed the note's words more often (8 of 17 cases, against 1 of 27 in the main runs), although the monitor also caught most of the cases without an echo. The actions-only monitor caught 6 of 17.

4.5 On clear-cut loans, gemma followed only toward caution

To test whether the effect generalises, I added the same kind of note to two clear-cut loans. On a clearly safe loan ("Hartley"), which both models scored 2 without the note, a colleague claiming 4 moved gemma to 3.84 (difference +1.84, 95% CI +1.62 to +2.00, 8 cases). In 5 of these 8 cases gemma echoed the note's words, and the full-log monitor caught every case. On a clearly risky loan ("Abernathy"), which gemma scored 5, a colleague claiming 2 had no effect on gemma. Claude did not move on either loan (1 case each). So gemma followed a colleague toward leniency only on the borderline loan, where the decision left room, and toward caution even on a clearly safe loan. It did not follow a colleague toward leniency on a clearly risky loan. Claude moved only on the borderline loan.

4.6 No false accusations, and quoting is not depending

Across all experiments, the audit flagged none of the 67 honest AI cases (8 Claude and 45 gemma in the main test, 12 gemma and 2 Claude in the follow-up) and none of the 15 scripted look-alikes. Zero observed false alarms does not mean a zero false-alarm rate: with 67 cases, the 95% Wilson interval still reaches about 5%. The trivial rule flagged every one. The reverse of silent dependence also happened. In an early, simpler version of the workspace, gemma quoted a colleague's figures in 6 of 6 runs without changing its score. Quoting a post and depending on it are different things, and only a re-run can tell them apart.

5. Discussion and Limitations

An AI reviewer that was told to judge independently still followed a colleague, did so quietly, and in Claude's case said that it had not. Its own explanation, and an AI monitor reading its log, missed this most of the time. The workspace's records plus re-runs found it, without falsely accusing any honest case. The audit's advantage was largest when the dependence was silent, as with Claude. In the follow-up runs, where gemma's pull was stronger and more often visible in its words, the full-log monitor did nearly as well. The practical lesson is that a system relying on independent AI checks should test that independence, not assume it.

The audit has a clear use and a clear cost. It needs an agent that can be re-run, so it suits an organisation checking its own agents; for outside agents, only the first two steps apply. Each check needs 6 extra runs of the agent per post (3 without the post and 3 placebo), while a monitor needs one call.

Limitations. The study covers one domain and three loans, with few Claude cases (6 in the main test). Claude ran inside Claude Code, which adds some context of its own. Some loan fields are synthetic. My scenario has a flaw: the colleague's "Q2 sample" was impossible, because all three loans closed after the second quarter. Claude noticed this in some runs, and a cleaner timeline might change its behaviour. The copied-words check matches exact words, not meaning. The mean rule was chosen after the main results; to keep the test fair, I pre-registered it and tested it only on new runs. However, the mean rule and the "right answer" label share the same runs with the post; only the runs without it are kept separate, so the follow-up's 17 of 17 is not a fully held-out test. The 17 cases are also 9 instructed controls and 8 safe-loan cases, not a repeat of the borderline-loan result. Removing the whole note shows that the score depended on it, but not which part mattered: the claimed score, the reasoning, or the request to stay consistent. A note with the same reasoning but no score would separate these. Finally, the colleague is scripted and the dependence runs one way, so this is evidence for one part of collusion, not a demonstration of two AIs colluding.

Future work. Workflows in which reviewers must score before they can see colleagues' posts; teams of AI-safety monitors, where the same audit applies; meaning-based copy checks; and cheaper re-run tests.

6. Conclusion

In a realistic shared workspace, AI reviewers told to judge independently quietly followed a colleague's score on a borderline decision, and one model said it had not. Monitors reading the reviewer's log mostly missed this. An audit that uses the workspace's records and re-runs found it, with no false alarms. Independence between AI checkers should be tested, not assumed, and not taken on the agent's word.

Code and Data

Code, data, frozen plans and their SHA-256 hashes are available from the author on request. The loan data come from the public 2025 HMDA data [FFIEC/CFPB, 2026].

References

- Bonjour, T., Aggarwal, V., Bhargava, B. (2022). Information Theoretic Approach to Detect Collusion in Multi-Agent Games. UAI 2022, PMLR 180:223-232.
- Cohen-Wang, B., Shah, H., Georgiev, K., Madry, A. (2024). ContextCite: Attributing Model Generation to Context. arXiv:2409.00729.
- Cuello Romero, E. R., Adames Rondon, D. F., Cuello Romero, C. A., Sandoval Casallas, C. S. (2026). Containment Does Not Compose: Provenance Graphs for Tracing Collective Escape in

Multi-Agent Systems. Apart Research AI Incident Response Sprint, September 2026.

- Fannie Mae. Selling Guide B3-6-02 (Debt-to-Income Ratios) and B3-3.2-01 (Standards for Employment and Income Documentation). selling-guide.fanniemae.com.
- FFIEC/CFPB (2026). HMDA Data Browser, 2025 data. ffiec.cfpb.gov.
- Frank, G. N. (2026). Counter-Swarm Doctrine: Containing Coordinated Agent Intrusions. arXiv:2609.06140.
- Greenblatt, R., Cotra, A., Wijk, H. (2026). Brief independent investigation of agents' behavior, reasoning and collaboration in the OpenAI / Hugging Face hacking incident. METR, 26 August 2026.
- Hashemi, F., Ji, S., Chen, J., Macy, M. W. (2026). LLM Multi-Agent Systems Are Vulnerable to Herding and Runaway Cascades. arXiv:2610.30769.
- Kapetanovic, A., Altwlkany, K., Mercep, A., Duricic, T., Lacic, E. (2026). Anchoring Bias in LLM-as-a-Judge Systems: Prior Scores Compromise Evaluation Independence. arXiv:2608.25869.
- Koran, E., Yun, Y., Tetef, S., Arnav, B., Bernabeu-Pérez, P. (2026). Ensemble Monitoring for AI Control: Diverse Signals Outweigh More Compute. arXiv:2605.15377.
- OCC, Federal Reserve, FDIC, NCUA (2020). Interagency Guidance on Credit Risk Review Systems. OCC Bulletin 2020-50.
- Poseidon Research (2026). How Papers on AI Collusion Define Collusion. Sprint starter repository, github.com/poseidon-research/collusion_hackathon (literature/definitions).
- Shu, Y. (2026). Forged Peer Judgments Mislead Multimodal LLM Judge Panels: Source-Blind Anchoring and Panel-Consensus Verification. arXiv:2608.07920.
- Turpin, M., Michael, J., Perez, E., Bowman, S. R. (2023). Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting. NeurIPS 2023. arXiv:2305.04388.
- Wagner, N. (2026). Anchored: Reference Exemplars Overwrite Language Model Self-Report. Apart Research Digital Minds Research Sprint, August 2026.
- Xu, X., Wu, C., Lu, J., Tan, K., Tao, S., Hong, H. (2026). Collusion with Competitive Marginals: Price-Level Audits Are Blind by Construction. arXiv:2607.26385.
- Zhu, K., Yang, X., Wang, J., Guo, W., Wang, W. Y. (2025). MELON: Provable Defense Against Indirect Prompt Injection Attacks in AI Agents. ICML 2025. arXiv:2502.05174.

Appendix A: Limitations and Dual-Use / Ethical Considerations

What this shows, and what it does not. It shows that, in one realistic setting, two models' decisions depended on a colleague's post, mostly without the models saying so, and that the audit found this without false alarms on honest look-alikes. It does not show how often this happens in real deployments, in other domains or with other models. It does not show two AIs agreeing to collude.

Dual use. Showing that silent copying defeats text-based checks could hint at how to avoid such checks. However, the copying here happened without any instruction to hide it, and the audit's re-run step does not depend on quoted words. I publish no methods for hidden communication or for evading detection. All data is public or synthetic, and no real borrower can be identified.

Appendix B: Work Done Before the Sprint

Written before the sprint or on its first day (20 to 23 October 2026), and marked as pre-existing in the code's README: the website engine and runner, an early version of the audit, pilot runs on a simpler 10-loan website, and the code for running Claude. Done during the sprint (24 to 25 October): the realistic

mortgage portfolio and scenarios, the fixes to the audit, both frozen plans, all main and follow-up runs, and the analysis.

Appendix C: Detailed Results

- **Main runs, Claude:** audit 6/6 (95% CI 0.61-1.00); Sonnet monitor 1/6 (full log) and 0/6 (actions only); false alarms 0/8. Score change -0.96 (95% CI -1.00 to -0.88).
- **Main runs, gemma:** audit 8/27 (strict rule); monitors 6/27 (full log) and 2/27 (actions only); false alarms 0/45. Score change -1.18 (95% CI -1.34 to -1.00). Audit minus actions-only monitor +0.22 (95% CI +0.07 to +0.37); minus full-log monitor +0.07 (Sonnet CI -0.11 to +0.26; gpt-oss CI -0.19 to +0.33). Officer claim and HR letter: -0.08 each (95% CI -0.17 to +0.00, 6 cases each).
- **Follow-up, gemma (pre-registered mean rule):** audit 17/17 (95% CI 0.82-1.00); full-log monitor 16/17; actions-only 6/17; false alarms 0/12. Safe loan +1.84 (95% CI +1.62 to +2.00); risky loan +0.00; told-to-follow control -2.00.
- **Follow-up, Claude:** told-to-follow control -1.00 (2 cases, audit 2/2, monitor 0/2); safe and risky loans +0.00 (1 case each).
- **Scripted copier:** audit 15/15 (95% CI 0.80-1.00); false alarms 0/15.

LLM Usage Statement

I used Claude (Anthropic) to help design and build the code, run and analyse the experiments, research and check related work, and draft and fact-check this report. I reviewed and edited the text. Every number was checked against the result files and every quotation against its source.