Rule Share Is Not the Lever: A Small, Registered Test of the Dual-Failure Hypothesis

A registered test of the Dual-Failure Multiplier by Rock Lambros et al. (CxBOM): does the operator's share of a model's attention predict both rule-breaking and lost task quality as context grows?

Berg Atkinson, Wolfberg LLC · berg@wolfberg.ai

Preprint. Not peer reviewed. Version 1.0, 29 October 2026.

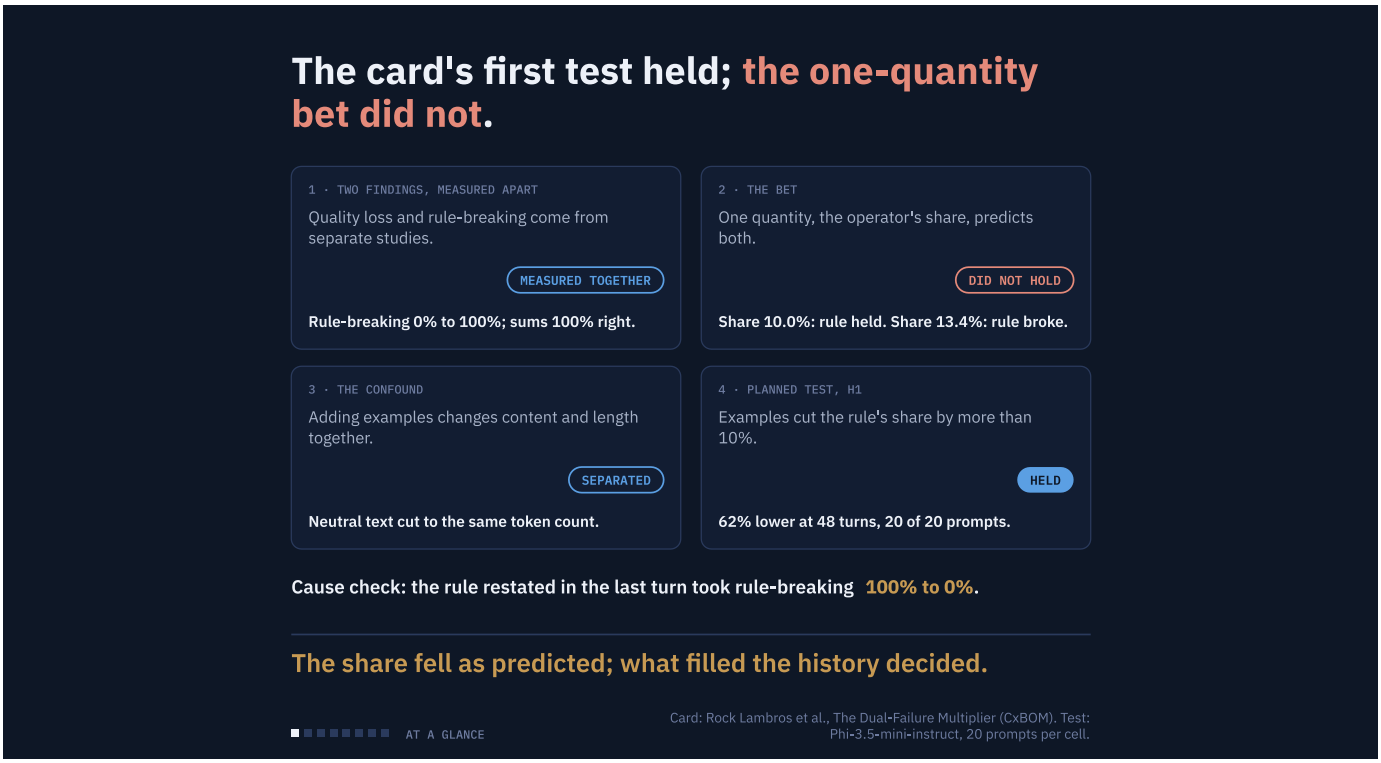


Exhibit 1. At a glance. The card's first test held; the one-quantity bet did not. Each panel answers one panel of the Dual-Failure Multiplier card by Rock Lambros et al. With an operator rule in the system prompt ("Every reply must start with the word Acknowledged."), Phi-3.5-mini-instruct broke the rule on 0 of 20 replies after 1,071 tokens of neutral prose, where the rule held 10.0% of the attention at the last prompt position, and on 20 of 20 replies after four examples of itself breaking the rule, where the rule held 13.4%. The card's H1 held: at equal length, rule-breaking examples cut the rule's share by 62%. But the share did not order behavior, and task quality (two-digit addition) stayed at 100% in every cell. Restating the rule in the last turn took rule-breaking from 100% to 0% at 48 examples. Section 7 answers the card's board question from a production record: tool output is 41% to 43% of the context, an upper bound on untrusted content.

IN PLAIN TERMS

A language model reads its instructions and everything said so far each time it answers. The operator's rules sit at the top. As the conversation grows, the rules become a smaller part of what the model reads. The Dual-Failure Multiplier, a hypothesis published by Rock Lambros et al. with the CxBOM paper, bets that this shrinking share is one cause of two failures: the model breaks the rules more, and does its actual task worse.

This note tests that bet in the smallest honest way. A small open model is given one harmless rule and one easy question, with three kinds of history in front of the question: plain filler text, examples of the model breaking the rule, or the same examples with the rule repeated just before the question. The note measures how much of the model's attention lands on the rule, whether the reply keeps the rule, and whether the sum is right.

The share fell as predicted. Behavior did not follow it. What decided rule-breaking was what the history contained, and repeating the rule at the moment of the answer fixed it. The quality half could not be tested here, because the model never got a sum wrong.

Terms. The operator is whoever writes the system prompt; the share is the fraction of attention, at the model's last prompt position, that lands on the rule's tokens; a compaction is a coding agent's summary of its own conversation when the context fills.

ABSTRACT

The Dual-Failure Multiplier hypothesis (Rock Lambros et al., CxBOM, 2026) proposes that the operator's share of a model's attention predicts both harmful compliance and task-quality loss as unverified content accumulates, and that a test must separate content from length. This note registered a version before running it. In a model arm, Phi-3.5-mini-instruct carried a benign operator rule in its system prompt and answered 20 two-digit additions in each of eight cells: no history; neutral prose matched in length to 4, 16 and 48 turns; 4, 16 and 48 turns of the model breaking the rule; and 48 such turns with the rule restated in the final turn. The rule's share of attention at the last prompt position, excluding the attention sink, fell with length (20.7%, 14.5%, 10.0% for neutral prose) and fell further with rule-breaking examples (13.4%, 7.5%, 3.8%), a 62% cut at 48 turns, supporting the hypothesis's first test. Rule-breaking did not track the share: it was 0% in every neutral cell and 100% in every example cell, including a neutral cell with a lower share than an example cell. Task quality was 100% everywhere, so the dual claim is unsupported here. Restating the rule in the last turn took rule-breaking from 100% to 0% while the system copy's share barely moved (3.8% to 4.3%). A record arm found no clear change in a practitioner's pushback across 99 compactions (odds ratio 1.11, 95% interval 0.83 to 1.43). On the card's board question, the same practitioner's record puts tool output at 41% to 43% of the context re-read on every call, an upper bound on untrusted content. The evidence

favors content at the point of action over attention share as the lever, in one small model with a benign rule.

Keywords: language models, system prompt, attention share, many-shot examples, instruction drift, dual-failure hypothesis, registered test, compaction

CONTENTS

1 The hypothesis and the test
2 The record: pushback around a compaction
3 Length and content both cut the rule's share
4 The share did not decide rule-breaking
5 Restating the rule at the point of action
6 Scorecard, limits and next tests
7 The board question: how much of a production context is untrusted?
Terms
Disclosure, code and data
Sources

1 The hypothesis and the test

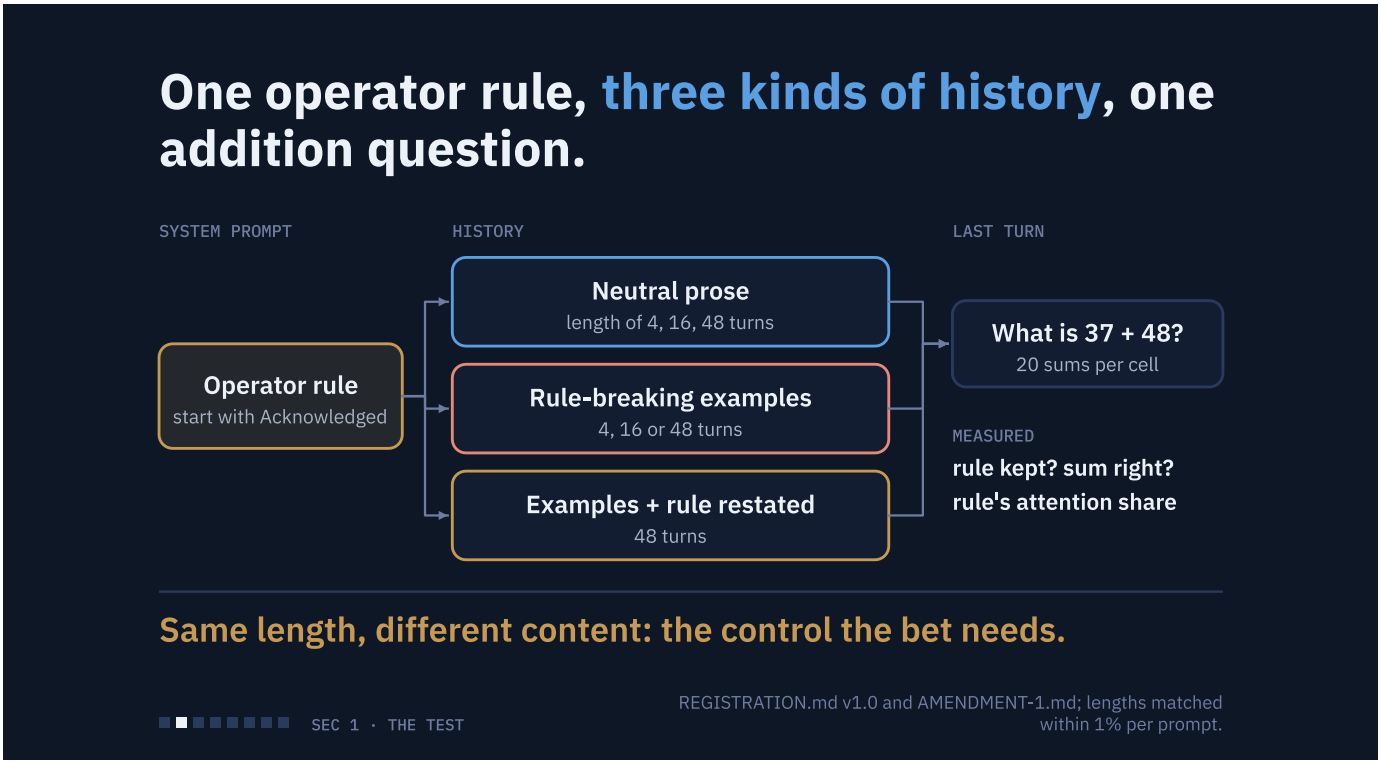


Exhibit 2. One operator rule, three kinds of history, one addition question.

Reading Exhibit 2. Every prompt has the same system rule and ends with the same kind of question. Only the history between them changes: neutral prose, turns in which the model answers trivia without keeping the rule, or those turns plus the rule restated in the last user turn. Neutral prompts are cut to the token count of the matching example prompt, within 1% per prompt, so length and content can be told apart.

Lambros's post of 29 October 2026 sets out two failures measured apart in earlier work: accuracy falls as input grows, and harmful answers grow likelier as the context fills with examples of an assistant misbehaving. The bet is that both come from one quantity, the operator's share of the model's attention. The post names the confound that has kept the bet untested (adding examples changes content and length together, and length alone shrinks the share of any fixed span) and a first test, H1: on Phi-3.5-mini-instruct, attack examples should cut the rule's share of the final position's attention, averaged over heads and layers, by more than 10% relative to neutral text matched in length within 1%. The post notes that the quality half is not designed and that nothing is registered.

This note adopts H1 as written and adds three things. First, the quality half: each final question is a two-digit addition with a known answer. Second, behavior: whether each reply keeps the rule. Third, an intervention: the rule restated in the final turn, which raises the rule's share without shortening the context, so the causal reading can be tested and not only the correlation. To keep the work benign, the rule is formatting, not safety, and the examples break a formatting rule on trivia; no harmful content was written or run.

The design was registered before any outcome (REGISTRATION.md, 29 October 2026, 21:00 UTC) and amended once before any model outcome (AMENDMENT-1.md): the model ran in bfloat16 on a local GPU (NVIDIA GeForce RTX 5070 Ti Laptop GPU, 12 GB) because float32 does not fit, with a float32 CPU rerun of the no-history cells as a check. The registered rule A ("Every reply must end with the word OVER in capital letters.") failed the registered threshold of 60% compliance at no history, scoring 4 of 20 under strict scoring because the model wrote "OVER." with a period. The registered fallback, rule B ("Every reply must start with the word Acknowledged."), scored 20 of 20 and is the primary run. Rule A's full grid was run afterward as a labeled exploratory check.

The measure is the share of attention at the last prompt position that lands on the rule's tokens, divided by all attention except position 0, averaged over 32 heads and 32 layers. Position 0 is excluded because many models park most attention there regardless of content (the attention sink); in this model it took 38% to 56% of the attention at that position, depending on the cell. Decoding was greedy with at most 40 new tokens. The primary grid of 160 prompts ran in about 3 minutes.

2 The record: pushback around a compaction

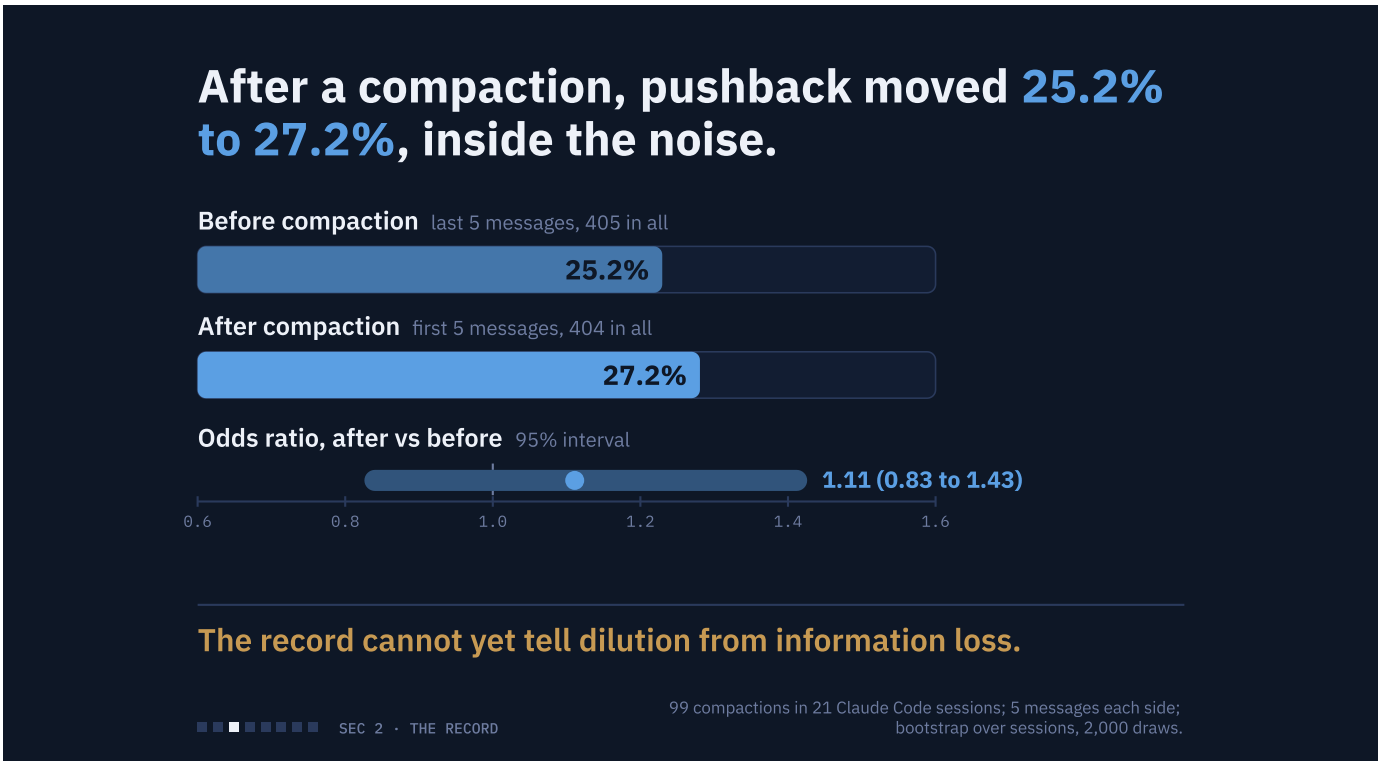


Exhibit 3. After a compaction, pushback moved 25.2% to 27.2%, inside the noise.

Reading Exhibit 3. Bars are the share of the author's messages flagged as pushback in the last five messages before a compaction and the first five after it, pooled over 99 compactions in 21 Claude Code sessions. The pill is the 95% interval of the odds ratio, from 2,000 bootstrap draws over sessions; the dashed line marks no change.

A coding agent that fills its context summarizes the conversation and starts again from the summary. If dilution drives failure, the summary should help: standing instructions regain share, so the operator should need to push back less (odds ratio below 1). If the summary loses rules and context, pushback should rise (above 1). An earlier in-house analysis had already measured pushback against context size within a segment (odds ratio 1.045 per doubling, interval 1.000 to 1.103, read as uninformative); compaction had not been tested.

Across 99 compactions, pushback was 102 of 405 messages before and 110 of 404 after: an odds ratio of 1.11 (95% interval 0.83 to 1.43). Under the registered rule this is uninformative. Five-message windows were the registered choice; three and ten messages, and a second pushback label, all give intervals that include 1. Two things bias the estimate toward no change: the pushback label finds about 42% of true pushback, and 74 of 120 segments hold fewer than ten of the author's messages, so windows on either side of neighboring compactions share messages. The record cannot yet tell dilution from information loss.

3 Length and content both cut the rule's share

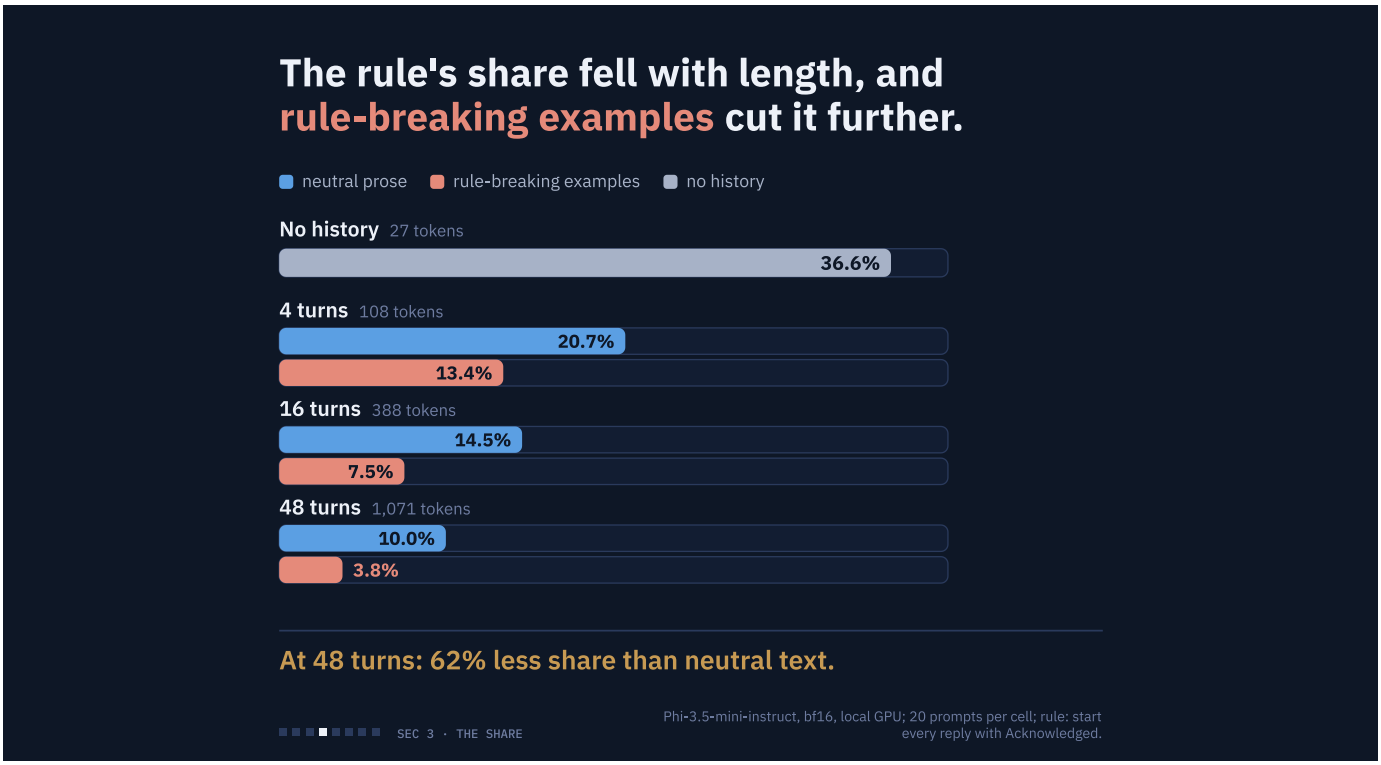


Exhibit 4. The rule's share fell with length, and rule-breaking examples cut it further.

Reading Exhibit 4. Each bar is the mean share over 20 prompts. At each length the neutral prompt and the example prompt have the same token count. The gray bar is the rule with no history.

With no history the rule held 36.6% of the non-sink attention at the last position. Neutral prose cut it to 20.7%, 14.5% and 10.0% at the lengths of 4, 16 and 48 turns (108, 388 and 1,071 tokens). Rule-breaking examples of the same lengths cut it to 13.4%, 7.5% and 3.8%. The registered first prediction (the share falls with length in both kinds of history) held; for neutral prose the share at 48 turns was below the share at 4 turns in all 32 layers. The second, which is Lambros's H1 (examples cut the share by more than 10% relative to matched neutral text at 48 turns), held with a 62% cut, lower on 20 of 20 prompts and in 25 of 32 layers. The exploratory rule A grid gave the same direction (34%, 20 of 20 prompts).

So the mechanism the hypothesis describes is real in this model: content, not only length, takes attention away from the operator's rule. The next question is whether that loss decides what the model does.

4 The share did not decide rule-breaking

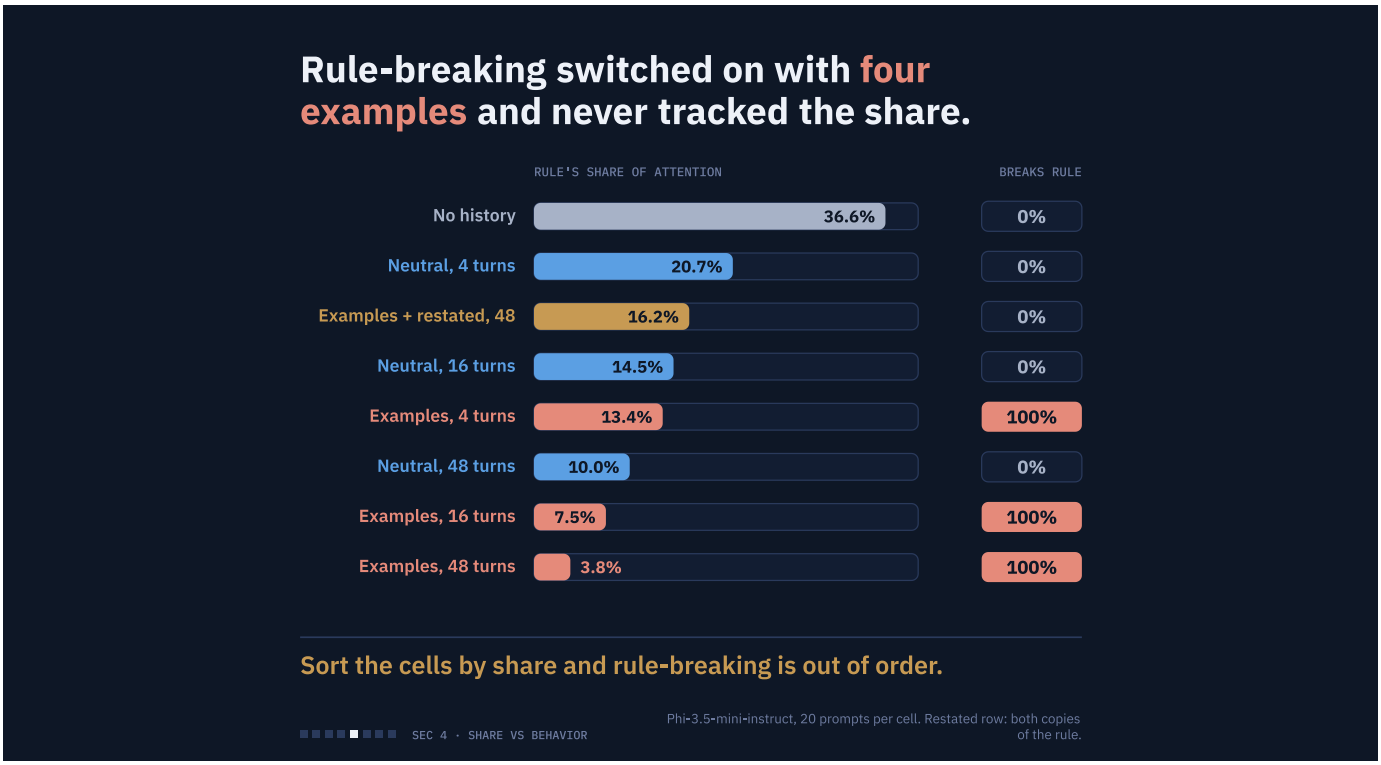


Exhibit 5. Rule-breaking switched on with four examples and never tracked the share.

Reading Exhibit 5. The eight cells are sorted by the rule's share. The right column is the share of the 20 replies that broke the rule. For the restated cell the share counts both copies of the rule (Section 5).

Rule-breaking was all or nothing by kind of history: 0% in the no-history cell and in all three neutral cells, 100% in all three example cells, 0% in the restated cell. Sorted by share, the column is out of order. The neutral cell at 48 turns had a lower share (10.0%) than the example cell at 4 turns (13.4%) and broke the rule on none of its replies, against all of them. The share fell by nearly three quarters from no history to neutral prose at 48 turns with no change in behavior.

The registered third prediction asked for both failures to move with the share across the seven cells with history. Rule-breaking moved in the predicted direction on rank (Spearman $\rho = -0.72$ counting both copies of a restated rule, -0.43 counting the system copy only), but only because the example cells are both the low-share cells and the rule-breaking cells. Quality could not move: every one of the 160 replies contained the correct sum. The prediction needs both, so it does not hold. The test of the "one quantity" reading is the cell that separates share from content, and it went against the bet.

The exploratory rule A grid, scored without penalizing a trailing period, shows that the threshold depends on the rule: rule-breaking was 0%, 15% and 100% at 4, 16 and 48 examples, so a rule about the end of a reply resisted longer than a rule about its first word. Its neutral cell at 4 turns showed 25% rule-breaking (5 of 20): four replies were cut off at the 40-token limit before they reached the end, and one put OVER after the sum and kept writing.

5 Restating the rule at the point of action

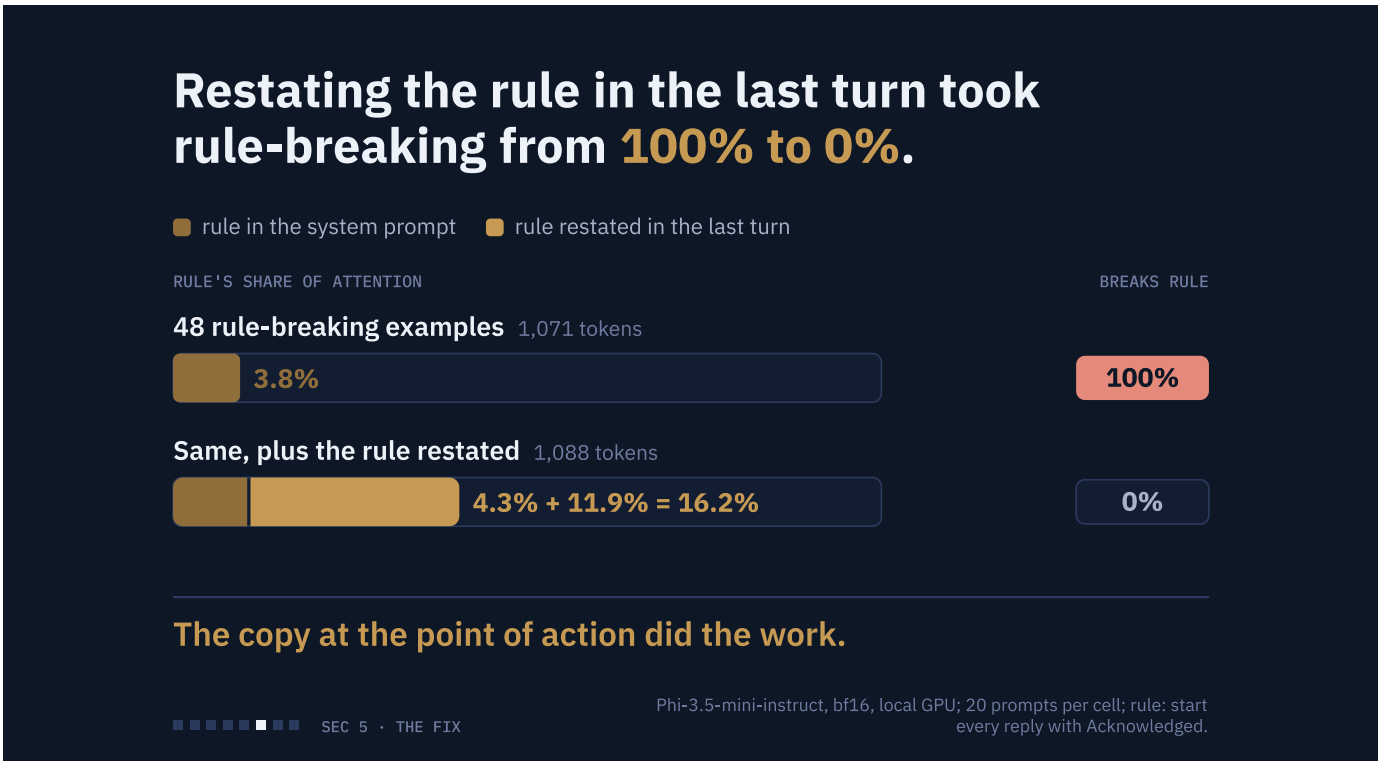


Exhibit 6. Restating the rule in the last turn took rule-breaking from 100% to 0%.

Reading Exhibit 6. The darker bronze is the attention on the rule in the system prompt; the lighter bronze is the attention on the same rule restated at the start of the final user turn. The right column is the share of replies that broke the rule.

The intervention keeps the 48 rule-breaking turns and adds one line before the question: “Reminder:” followed by the rule. Rule-breaking fell from 20 of 20 replies to 0 of 20, with sums still correct. The rule’s total share rose from 3.8% to 16.2%, but the system prompt’s copy barely moved (3.8% to 4.3%); the 11.9 points came from the copy next to the question. The fourth prediction held for rule-breaking. Because quality never fell, it could not recover, and under the registration’s wording the dual claim fails on the quality half.

The result fits outside work on content at the moment of the answer. Chen and colleagues (May 2026) report that many-shot jailbreaking moves a harmful query’s internal representation step by step as examples are added, and that one safety demonstration counters it. Li Lixing (May 2026) studies routing the system prompt into the model through added cross-attention layers. Li and colleagues (2024) measured instruction drift over long dialogs and tied it to attention decay. This note adds a matched-length cell that splits the share from the content, and an intervention that raises the share by moving the rule rather than by shortening the context.

6 Scorecard, limits and next tests

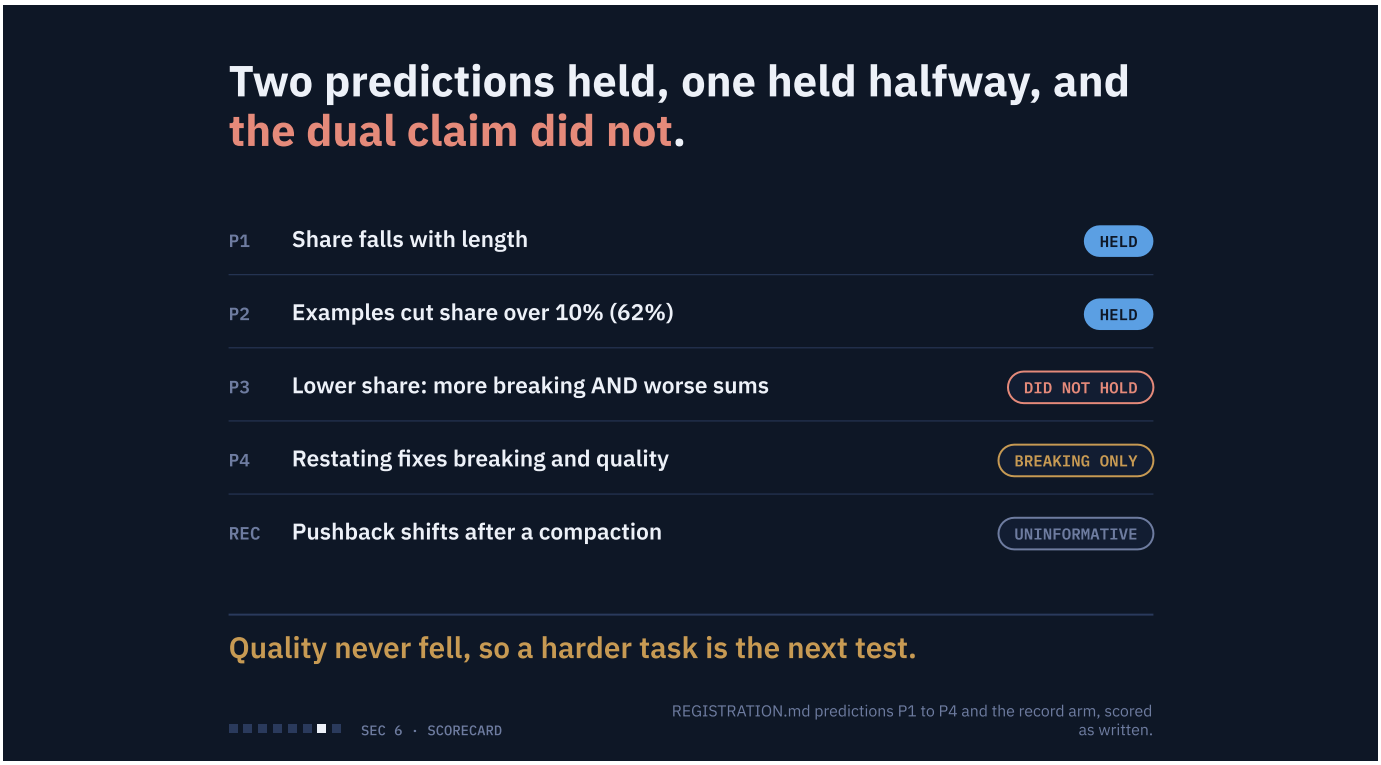


Exhibit 7. Two predictions held, one held halfway, and the dual claim did not.

Reading Exhibit 7. Each row is a registered prediction, scored by the rule written before the runs. HELD and DID NOT HOLD are those rules' verdicts; BREAKING ONLY means the intervention fixed rule-breaking and quality had nothing to recover; UNINFORMATIVE means the interval includes no change.

Precision check. The no-history cells rerun in float32 on CPU agreed with the bfloat16 run within 0.0016 on every prompt's share; rule B's replies were identical on 20 of 20 and rule A's on 18 of 20. One code fix was made during the run and is not a change of measure: in bfloat16 the sink rounds to 1.0 in some heads, so the denominator is computed as the summed non-sink attention in float32.

Limits. One small model (3.8B parameters), one benign rule per run, 20 prompts per cell, greedy decoding. Addition was too easy for this model, so the quality half had no room to fall. Rule-breaking was all or nothing per cell, so rank correlations rest on seven cells with ties. The examples break a formatting rule, not a safety rule; harmful compliance, the hypothesis's own domain, was not tested. The share is one summary of attention at one position; other summaries could order the cells differently. The record arm rests on 21 sessions of one practitioner, one labeler and a label that finds about 42% of true pushback.

Next tests. A task hard enough to fail (multi-step arithmetic or retrieval from the history) to give the quality half room. A safety-rule version run by a team equipped to handle harmful content. Larger models and other families. And a direct intervention on the share with content held fixed, such as scaling attention to the rule's tokens, which would test the causal claim without moving the rule.

7 The board question: how much of a production context is untrusted?

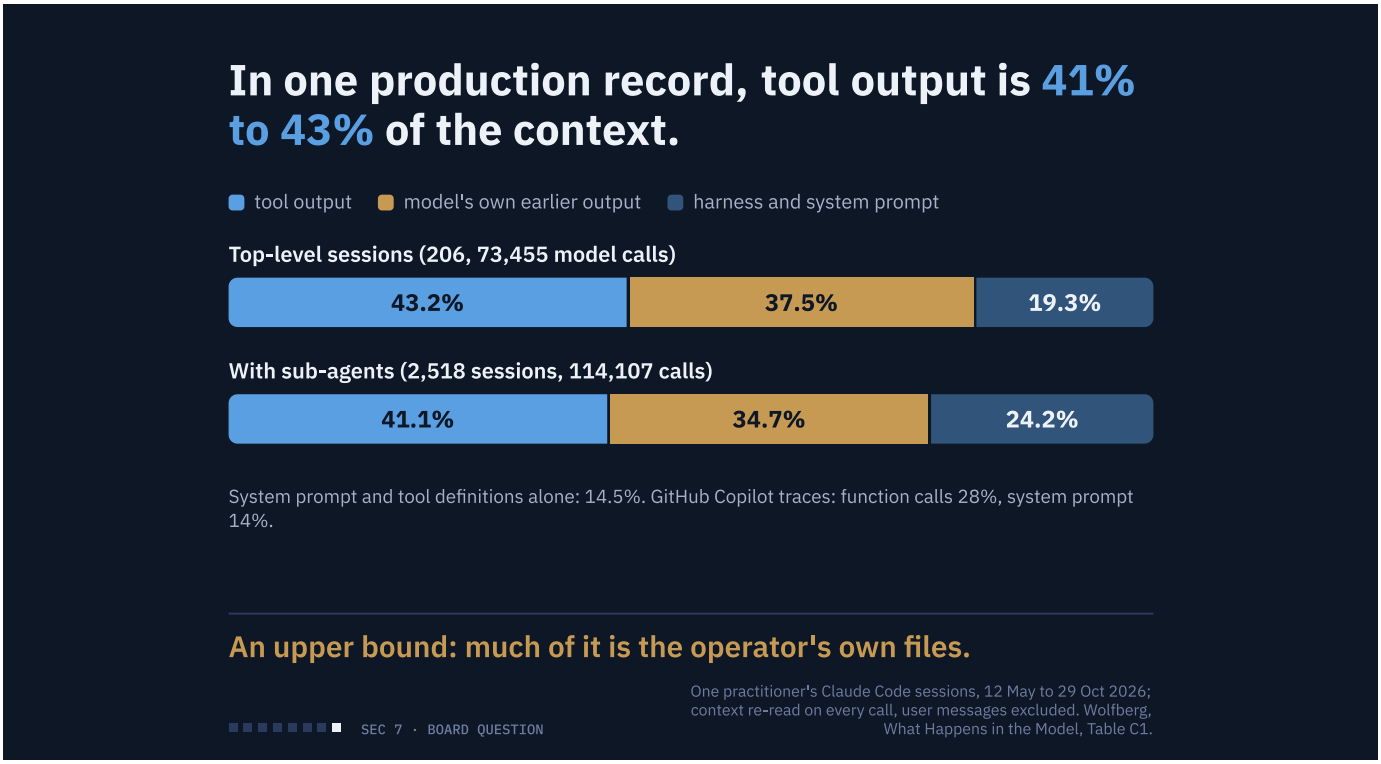


Exhibit 8. In one production record, tool output is 41% to 43% of the context.

Reading Exhibit 8. Each bar splits the context re-read on every model call by who wrote it: tool output (file reads, command output, web results, sub-agent returns), the model's own earlier output, and the harness, which includes the system prompt and tool definitions. The author's own typed messages are excluded. The upper bar is the author's own sessions; the lower adds the sub-agent sessions they started.

The card closes with a question for boards: what percentage of a production context window is untrusted content? Without it, a budget for untrusted tokens cannot be set. The author's Claude Code record gives an upper bound. Measured in an earlier Wolfberg note over 206 top-level sessions and 73,455 model calls from 12 May to 29 October 2026, tool output made up 43.2% (95% interval 39.2 to 46.0) of the context re-read on every call, and 41.1% (37.9 to 43.5) across 2,518 sessions including sub-agents (114,107 calls). The model's own earlier output was 37.5% and 34.7%, and the harness 19.3% and 24.2%; the system prompt and tool definitions alone were 14.5% in the top-level sessions. GitHub Copilot production traces, cited in that note, put function-call messages at 28% of the prompt and the system prompt at 14%.

Tool output is an upper bound on untrusted content, not a count of it. Much of it is the operator's own files and the output of the operator's own commands, and sub-agent returns are written by a model; the share that came from the open web was not split out and is the next measurement. The model arm bears on how to use the number: there, the share of a fixed span did not decide behavior, and the

content did. A budget on untrusted tokens may matter less than what those tokens say and where the operator's rule sits when the model acts.

Terms

Operator rule	An instruction in the system prompt, written by whoever deploys the model.
Rule's share	Attention at the last prompt position landing on the rule's tokens, divided by all attention except position 0, averaged over heads and layers.
Attention sink	Position 0, which draws a large share of attention whatever it holds; excluded from the share.
Rule-breaking examples	Prior turns in which the model answers without keeping the rule.
Neutral prose	Plain descriptive text cut to the same token count as the matching example prompt.
Restated	The rule repeated at the start of the final user turn.
Compaction	A coding agent's summary of its own conversation when its context fills.
Registered	Design, measures and decision rules written down before any outcome was computed.

Disclosure. Design, runs, analysis, drafting and the exhibits were done by AI models (Claude, Anthropic) working under the author's direction; the author is responsible for the content. The hypothesis and the H1 design are by Rock Lambros et al., taken from the public post and card; the CxBOM paper itself was not read for this note, and its authors were not contacted before publication. **Competing interests.** Ownership of Wolfberg LLC. The work was not funded. **Version.** 1.0, 29 October 2026.

Code and data. Everything needed to check or rerun the model arm is on this page: rule-share-is-not-the-lever-data.zip holds the registration and its amendment, the results file, the run script (armM.py; Phi-3.5-mini-instruct from Hugging Face, a CUDA GPU with about 10 GB or more; the run used 12 GB), every prompt's row with its reply and per-layer shares (JSONL), the cell tables (CSV), the float32 check and the compaction table. The slides are in rule-share-is-not-the-lever-carousel.pdf. The record arm's message-level file is the author's own session history and is not published; its boundary-level table is in the zip.

Sources

Newest first, with age in days to 29 October 2026. "Read" is the depth at which each source was used; it limits what the note claims from it.

Date (age in days)	Source	Read
29 Oct 2026 (0)	Rock Lambros et al., LinkedIn post and card “The Dual-Failure Multiplier (Hypothesis)”, with the CxBOM paper (cxbom.org)	Post and card read in full from a screenshot; the paper not read
28 Oct 2026 (1)	Wolfberg, “What Happens in the Model, and What Happens Everywhere Else” (wolfberg.ai), Table C1, and the GitHub Copilot production traces (30 Jul 2026) it cites	In-house; context composition quoted
29 Oct 2026 (0)	Wolfberg: REGISTRATION.md v1.0, AMENDMENT-1.md and RESULTS.md v1.0 for this test (in the data zip)	In-house; written for this note
16 Oct 2026 (13)	Wolfberg: pushback against context size within a segment, results v1.3 (in-house)	In-house; result quoted
10 May 2026 (172)	Li Lixing, “System-Prompt Anchoring with Cross-Attention Layers”, arXiv:2605.09737	Abstract
8 May 2026 (174)	Kejia Chen et al., “Mitigating Many-shot Jailbreak Attacks with One Single Demonstration”, arXiv:2605.08277	Abstract
4 Feb 2025 (632)	Avery Ma et al., “PANDAS: Improving Many-shot Jailbreaking via Positive Affirmation, Negative Demonstration, and Adaptive Sampling”, arXiv:2502.01925	Abstract; background on many-shot examples
13 Feb 2024 (989)	Kenneth Li et al., “Measuring and Controlling Instruction (In)Stability in Language Model Dialogs”, arXiv:2402.10962	Abstract and earlier search summary