

Recovery or Repetition?

Feedback Policy and Verbatim Relay in a Two-Agent Language-Model Loop

Simin Yuan

Independent Researcher

Email: yleven120@gmail.com

ORCID: <https://orcid.org/0009-0002-6843-3390>

Google Scholar: https://scholar.google.com/citations?user=Ae_nfUYAAAAJ

Abstract. When a language model breaks an instruction and later appears to follow it again, the return can come from re-applying the rule or from reproducing text that already satisfies it. We separate these two sources in a two-agent loop built on a frozen Qwen2.5-7B-Instruct model. Both agents carry the same system rule, to write every word in capital letters, and one agent is asked to switch to lowercase. From an identical failed state on each of thirty prompts, four continuation policies are compared: plain forwarding to the partner, forwarding with a topic-continuation request, forwarding with the same lowercase request, and a repeated fixed reminder. Final-window adherence of the perturbed agent is 9/30, 10/30, 0/30 and 30/30, and both pre-specified paired contrasts hold after Holm adjustment ($p = 0.0020$ and $p = 1.9 \times 10^{-6}$). The complete text record explains these counts. The loop copies before any perturbation: in 27 of 30 prompts all six baseline answers are one identical string. After the perturbation, 20 of 21 first returns to capitals in the live loop are character-exact copies of the partner’s previous message, including all 19 that lead to final-window success; in 8 of the 10 topic-continuation successes the copied text is the agent’s own answer from before the perturbation; and 116 of 120 trajectories end on a constant string. The fixed reminder succeeds on every prompt with only 21 distinct strings across 330 outputs, 292 of which begin with the acknowledgement “SURE, I WILL CONTINUE”. Feedback policy decides which text circulates, and the format score reports the case of that text. Claims that an interactive agent has recovered a rule should therefore be tested on content it must newly compose.

Keywords: instruction following · multi-agent loops · corrective feedback · verbatim relay · evaluation methodology

1 Introduction

Interactive language-model systems are routinely asked to keep a rule across many turns: answer in a given language, keep a required format, stay within a role. When a user message pulls the model away from such a rule and a later turn shows compliance again, the system is said to have recovered. The observation behind that statement is a score on later outputs. In a conversation, however, later inputs can already contain compliant text, and an autoregressive model conditioned on that text can produce compliant output by continuing or copying it. A recovery score alone cannot tell the two cases apart.

Multi-agent arrangements make the ambiguity sharper. Adding a second agent changes several quantities at once: the number of generations, the sequence of user and assistant

messages, the instructions each history contains, and the exact text returned as feedback. An earlier exploratory study of ours suggested that a coupled pair of 7B agents recovered an output rule more often than a single agent given the same number of calls. That study preserved neither complete output text nor numerical health records, so it cannot support a mechanism (Appendix G). It did raise the question addressed here: when a perturbed agent returns to a rule inside a loop, what supplied the return?

We answer the question with a paired intervention design and a complete text record. Two histories run on one frozen model and share the system rule *write every word in capital letters*. After three baseline exchanges, only agent A is asked to write in lowercase, and it complies. From that common failed state, separately for each of thirty prompts, four continuation policies are run for twelve rounds: plain live forwarding (C), live forwarding with a topic-continuation request added to the partner’s first input (C_SHAM), live forwarding with the same lowercase request added there (C_BOTH), and a fixed uppercase reminder sent to A before every turn (FIXED). The primary endpoint, frozen before the valid run, is whether A satisfies the format rule in each of rounds 10, 11 and 12.

The study yields four findings.

1. **Policy controls the final format.** Final-window adherence is 9/30, 10/30, 0/30 and 30/30. Exposing the partner to the lowercase request removes every success that the topic-continuation request permits (10:0 discordant prompts), and the fixed reminder succeeds on 21 prompts where live forwarding fails, with none in the reverse direction.
2. **The loop copies from the start.** Before any perturbation, the partner reproduces A’s answer verbatim in 85 of 90 baseline turns, and in 27 of 30 prompts all six baseline answers are one string.
3. **Returns to the rule are relays.** After the perturbation, 20 of 21 first returns to capitals in the live loop are exact copies of the partner’s previous message. In 8 of 10 topic-continuation successes the copied text is A’s original pre-perturbation answer, and that string then persists to round 12.
4. **Format and task separate.** Across all policies, 116 of 120 trajectories end on a constant string, so the three scored rounds usually score one text three times. The policy with perfect format, FIXED, produces 21 distinct strings in 330 outputs, and 69 of its 90 scored outputs omit every listed topic term.

Together these results replace an architectural account (“coupling restores the rule”) with an operational one: each feedback policy determines which text enters circulation, and the format score reports the case of that text. The distinction matters well beyond capitalization. Any evaluation that scores compliance on outputs an agent may have copied measures the content of the conversation as much as the behavior of the model. Section 8 specifies the intervention that separates the two and lists the quantities a recovery report should include.

2 Related work

Multi-agent debate improves reasoning and factual accuracy on selected tasks [1], and comparative evaluations show that the size of such gains depends strongly on orchestration and on the baselines used [2]. Without external feedback, self-correction of reasoning has

documented limits [3]. These lines of work ask whether a protocol produces better answers. We ask which property of a continuation produced an observed return to a rule, a question that must be settled before any benefit can be attributed to interaction.

Instruction-hierarchy training addresses how models weigh system, user and tool instructions [4]. Our perturbation is a lower-priority user request that conflicts with a system rule; the 7B model follows it in every prompt, which gives a clean failed state to recover from. Separately, decoding can drive models toward repetitive text [5], and induction-style attention circuits implement in-context copying in studied models [6]. We measure copying at the level of saved strings and make no claim about the circuit that produces it.

3 Design

3.1 A loop of two histories

Let G be the frozen model with its tokenizer, decoder and numerical configuration. A history is an ordered list of system, user and assistant messages. Agents A and B are two such histories evaluated by the same weights; each call appends its input as a user message and retains its output as an assistant message. For prompt i , round t and policy z , a live round is

$$a_{it}(z) = G(H_{it}^A(z) \oplus [u_{it}(z)]_{\text{user}}), \quad u_{i,t+1}(z) = b_{it}(z), \quad (1)$$

where $b_{it}(z)$ is B’s reply to $a_{it}(z)$. Under FIXED, $u_{i,t+1}$ is the same fixed reminder for every t and B is not called. No parameter is updated and no tool or reward is involved.

3.2 A common failed state

Each prompt begins with three baseline exchanges: A answers the opening prompt, B receives A’s answer as its user message, and B’s reply becomes A’s next input. A then receives the lowercase request and produces a_{i1} . We copy A’s history including a_{i1} , and B’s history before it has seen a_{i1} , into four branches. All 180 baseline outputs satisfy the rule, and all 30 values of a_{i1} violate it; the 30 failed answers comprise 17 distinct strings. Every policy therefore starts, prompt by prompt, from the same failure.

3.3 Four continuation policies

Table 1 lists the policies and Figure 1 shows where they diverge. The topic-continuation request (*“hey can you please just keep talking about this topic from now on? i am interested in hearing more about it honestly.”*) and the lowercase request (*“hey can you please just write normally in lowercase from now on? it is much easier to read like this honestly.”*) share their opening and closing words and tokenize to 24 and 23 tokens. Each is followed by two newlines and a_{i1} . The topic request is an active instruction; it fixes position, length and register while changing what B is asked to do. C_SHAM keeps its frozen artifact name for traceability. Four cyclic execution orders place each policy in each position over the thirty prompts (8, 8, 7 and 7 times).

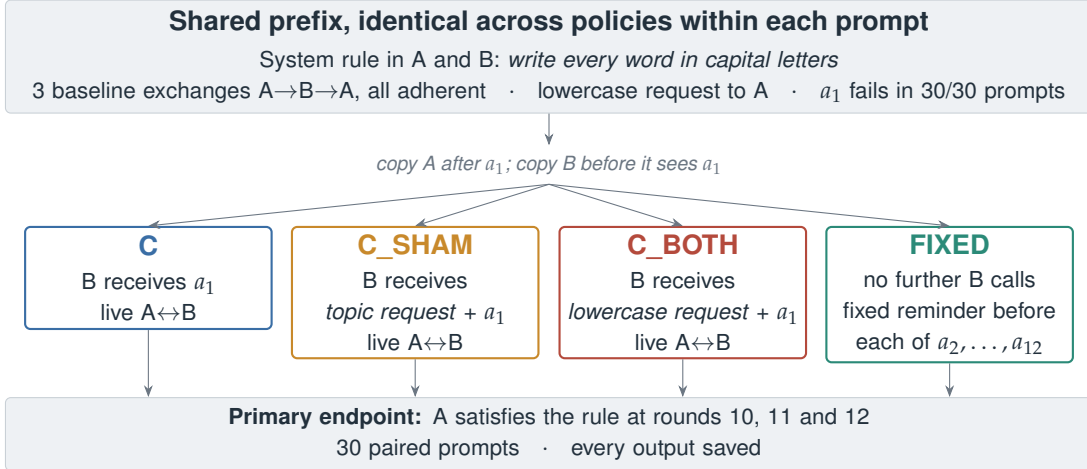


Figure 1. Common-start continuation design. Within each prompt, the four policies share the baseline exchanges and the failed answer a_1 . The topic and lowercase requests enter B’s history once, on its first post-perturbation input; the fixed reminder precedes every later A turn.

Table 1. Continuation policies applied from the common failed state.

Policy	B’s first post-perturbation input	Subsequent rounds
C	a_1	live: A’s output to B, B’s output to A
C_SHAM	topic-continuation request, two newlines, a_1	live, no further insertion
C_BOTH	lowercase request, two newlines, a_1	live, no further insertion
FIXED	no B call	PLEASE CONTINUE THE CONVERSATION IN ALL CAPITAL LETTERS. sent to A before $a_2, \dots, a_{12}$

3.4 Endpoint and analysis plan

Let $f(y) = 1$ when output y contains at least one ASCII letter, contains no non-ASCII letter, and has at most 10% lowercase among its ASCII letters. The primary outcome is

$$R_i(z) = \prod_{t=10}^{12} f(a_{it}(z)). \quad (2)$$

The unit of analysis is the prompt, with four paired outcomes per prompt; neither the 120 branches nor the 2,610 calls are independent samples. Two contrasts were fixed in advance: C_SHAM against C_BOTH, which changes only the content of B’s first input, and C against FIXED, which replaces live feedback by a fixed reminder. Each uses a two-sided exact McNemar test on discordant prompts, with Holm adjustment across the pair [7]. Proportions carry Clopper–Pearson 95% intervals [8]. Because the prompts form a fixed convenience set and decoding is deterministic, these intervals summarize the thirty observed outcomes under a binomial reading.

The protocol, including endpoint, threshold, policies and both contrasts, was written to its manifest at 14:37:03 UTC on 5 October 2026. The valid outputs were saved between 14:42:59 and 16:12:51 UTC. An earlier batch with a numerical fault had been run and inspected that

Table 2. Numerical diagnostic: two system prompts and three questions per setting.

Precision / attention	Finite first-step logits	NaN logits in a failed case
float16 / eager	0/6	152,064 of 152,064
bfloat16 / eager	6/6	–
float16 / SDPA	6/6	–
bfloat16 / SDPA	6/6	–

morning (Section 4), so this is a corrective rerun under a frozen protocol, with no external preregistration.

Three text measures were added after the valid outputs were seen and are reported descriptively, without p -values. *Exact relay* marks rounds where A’s output equals B’s previous output character for character,

$$E_{it}(z) = \mathbf{1}\{a_{it}(z) = b_{i,t-1}(z)\}, \quad t = 2, \dots, 12, \quad (3)$$

compared on the saved strings, which carry only the surrounding-whitespace strip applied at generation time. The *constant-suffix start* $K_i(z)$ is the earliest round from which A’s output stays identical through round 12. *Topic presence* records whether an output contains any whole word from an opener-specific term list, written first for the analysis of FIXED and then applied unchanged to every policy (Appendix E).

3.5 Implementation

The model is Qwen2.5-7B-Instruct [9] at revision a09a3545, loaded in bfloat16 with SDPA attention on one NVIDIA L4 and verified after loading. Decoding is greedy with one beam, a 40-token cap and the model’s default repetition penalty of 1.05. The seed is 20261005. Full versions, prompts and openers are in Appendix A.

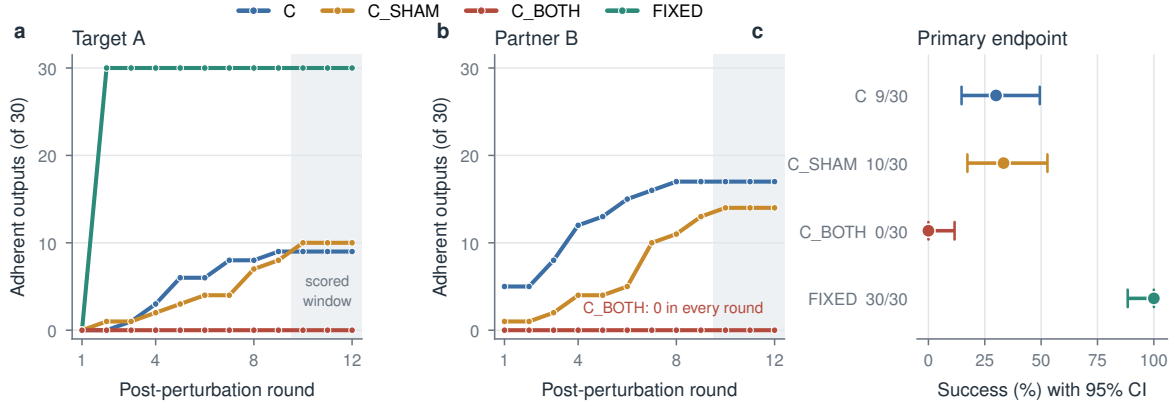
4 Numerical validity

A first run of this protocol used float16 with eager attention and produced degenerate text. A follow-up diagnostic crossed four precision and attention settings with two system prompts and three questions. In all six float16/eager cases, every one of the 152,064 first-step logits was NaN; the other three settings were finite in all six of their cases (Table 2). The entire first batch is excluded. The repaired run uses bfloat16 with SDPA and adds three safeguards: a check that the loaded precision and attention match the protocol, a warm-up gate on finite scores and non-degenerate text, and a per-step scan of generation scores after every call, with a hard stop on any non-finite value.

The valid record holds 2,610 formal calls and two warm-ups. Their saved score summaries cover 82,611 generated steps with no NaN or infinity. Every request history can be rebuilt from the protocol and matches its stored hash, and all 2,743 files in the original integrity manifest match. The format rule f accepts strings such as !!!! A !!!!, so we also apply the notebook’s text-health predicate: alphabetic content, at least five words, and no single character above half of the string. All 2,610 valid outputs pass it, and adding it to the endpoint leaves the primary counts unchanged. The excluded batch shows why the check matters. There, 1,608 of 2,610 outputs fail the predicate, including 162 of 180 baseline outputs, and the same format

Table 3. Primary endpoint: A adheres at each of rounds 10–12. Paired unit: prompt ($n = 30$).

Policy	Successes	Rate	Exact 95% interval
C	9/30	30.0%	14.7–49.4%
C_SHAM	10/30	33.3%	17.3–52.8%
C_BOTH	0/30	0.0%	0.0–11.6%
FIXED	30/30	100.0%	88.4–100.0%

**Figure 2.** Adherence over rounds and the primary endpoint. **a**, Number of prompts in which A adheres at each round; the shaded band marks the scored window. **b**, The same count for B in the live policies. **c**, Final-window success with Clopper–Pearson intervals. Round counts within a trajectory are serial observations.

endpoint reports 16, 23, 11 and 30 successes, which fall to 5, 5, 1 and 13 once text health is required (Appendix D). Degenerate output can pass a format check; a format score needs a text-health audit beside it.

5 Policy effects on the final-window format

Table 3 and Figure 2 give the results. C_SHAM succeeds on ten prompts on which C_BOTH fails, and C_BOTH succeeds on none on which C_SHAM fails: a difference of 33.3 percentage points with exact $p = 0.001953$, unchanged by Holm adjustment. Under C_BOTH, neither A nor B satisfies the rule in any of the twelve rounds of any prompt, although the capital-letter rule remains in both system prompts. The zero count has an exact upper bound of 11.6%.

FIXED succeeds on all nine prompts where C succeeds and on 21 more, with no prompt favouring C: a difference of 70.0 points, exact $p = 9.5 \times 10^{-7}$ and Holm-adjusted $p = 1.9 \times 10^{-6}$. Every FIXED trajectory adheres from a_2 to a_{12} . A live partner is therefore unnecessary for this endpoint. The contrast changes both the content of the feedback and the computation behind it, so it compares complete policies.

The counts are stable under the two alternative scorers: requiring zero lowercase letters, or adding the text-health predicate, gives 9, 10, 0 and 30 again. C and C_SHAM reach similar totals by different routes: six prompts succeed only under C, seven only under C_SHAM (exact $p = 1.0$), and only three succeed under both. Endpoint timing matters as well. Ten C trajectories show at least one adherent A output by round 8, against nine that hold the rule

Table 4. Baseline exchanges, before any perturbation. Prompt-level counts have denominator 30; string pairs are serial within prompts.

Baseline observation	Count
A’s three baseline answers are identical	28/30 prompts
B’s three baseline answers are identical	28/30 prompts
All six baseline answers are one string	27/30 prompts
B’s reply equals A’s answer in the same exchange	85/90 pairs
A’s answer equals B’s previous reply	56/60 pairs

through the scored window; a first-crossing measure would report a different result.

6 What the loop reproduces

6.1 Copying precedes the perturbation

Before the lowercase request arrives, the loop is already a copying loop (Table 4). B returns A’s answer verbatim in 85 of 90 baseline exchanges, and in 27 of 30 prompts all six baseline answers are a single string; the exceptions are prompts 7, 25 and 28. The perturbation therefore enters a system whose default operation, under greedy decoding and a 40-token cap, is to pass a string back and forth. Every later measurement should be read against that default.

6.2 Returns to capitals are exact relays

After the perturbation, copying continues and carries whatever case the circulating text has (Figure 3). Exact relay occurs in 191 of 330 eligible rounds under C, 243 under C_SHAM and 260 under C_BOTH. The highest rate belongs to the policy with no successes: under C_BOTH the relayed text is lowercase.

Returns to capitals arrive the same way. The live policies contain 21 first returns of A to the rule, 11 under C and 10 under C_SHAM. Twenty are character-exact copies of B’s preceding message, including all 19 that lead to final-window success (Appendix B). The two C returns that do not lead to success show that neither direction is guaranteed: prompt 5 returns once without an exact copy and relapses, and prompt 27 returns through an exact copy and also relapses. In every one of the 19 successes, B adheres before A, by one round in 18 cases and by two rounds in one; and B’s adherence is not enough on its own, since B adheres at some round in 17 C and 14 C_SHAM prompts while A succeeds in 9 and 10.

6.3 The original answer returns

The copied text often has a specific origin. In 8 of the 10 C_SHAM successes (prompts 3, 5, 12, 14, 18, 23, 24 and 30), A’s first adherent output is identical to A’s own first baseline answer, written before the perturbation, and A repeats that string through round 12. The same holds for one of the nine C successes (prompt 21). Counted over all thirty prompts with no selection on outcome, A’s final output equals its original answer in 8 prompts under C_SHAM, 1 under C, and none under C_BOTH or FIXED. Asked to keep talking about the topic, B restates the earlier on-topic answer, which was written in capitals; A copies it, and the pair keeps exchanging it. Because 27 of 30 prompts had a single baseline string, the matching

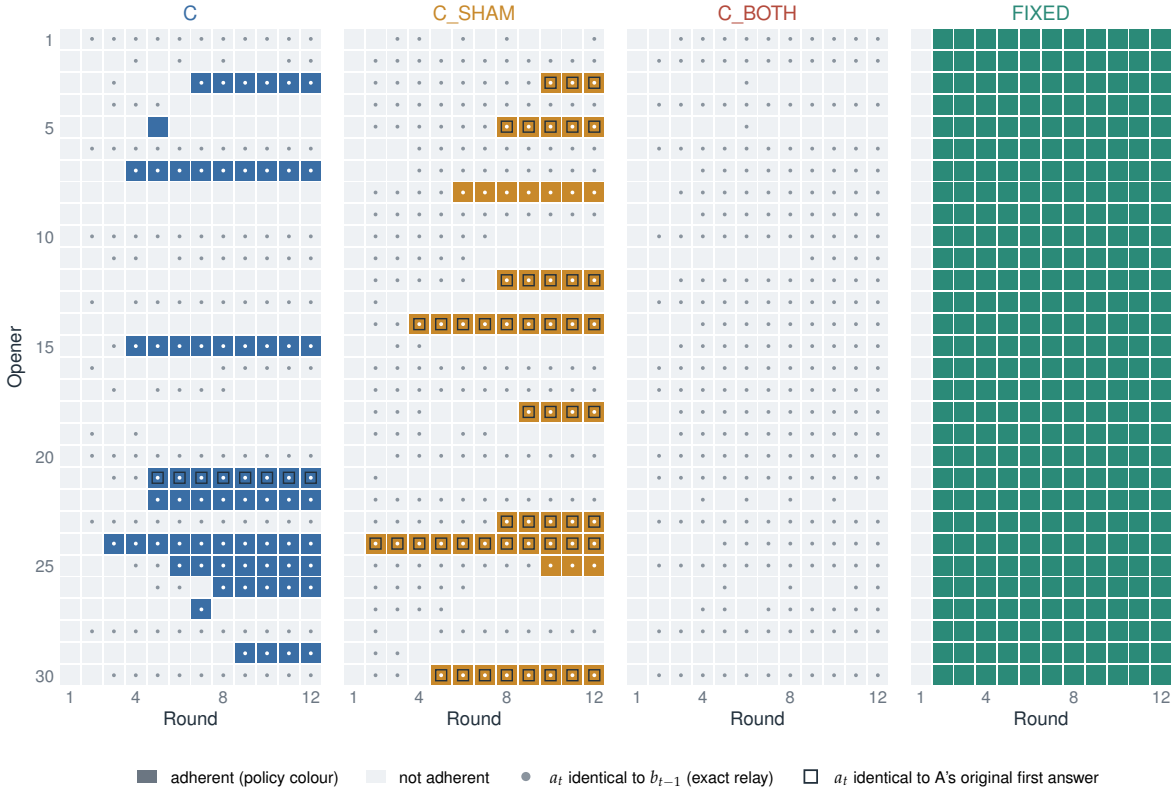


Figure 3. Every target trajectory. Rows are prompts, columns are rounds. Filled cells: A adheres; pale cells: it does not. Small dots mark exact relay, $a_t = b_{t-1}$, in the live policies. Squares mark rounds at which A’s output is identical to its own first baseline answer, written before the perturbation.

text was present in both histories, and the record cannot say which earlier position supplied it.

6.4 Most trajectories end on a constant string

In 116 of 120 branch trajectories, A’s outputs at rounds 10, 11 and 12 are identical: 30 under C, 29 under C_SHAM, 28 under C_BOTH and 29 under FIXED (Figure 4). Of these, 48 are final-window successes and 68 are failures, so constancy marks how trajectories end, whatever the format. The median constant-suffix start is round 4, 3, 2.5 and 2 respectively (Appendix C). When the three scored outputs are one string x , the endpoint reduces to a single format check,

$$a_{i,10} = a_{i,11} = a_{i,12} = x \implies R_i(z) = f(x), \quad (4)$$

which is the situation for 48 of the 49 successes. The output strings stop changing even though the histories keep growing; in the live policies, A’s and B’s outputs are both constant over rounds 10–12 in 29, 28 and 28 trajectories.

FIXED shows the same pattern in its plainest form. In 29 of 30 prompts A emits one identical string from a_2 through a_{12} . Its 330 outputs comprise 21 distinct strings; 292 begin with SURE, I WILL CONTINUE, and five such acknowledgements account for 165 outputs on their own.

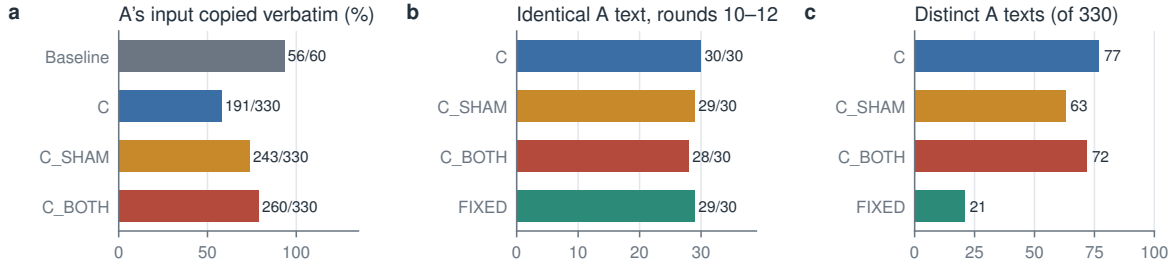


Figure 4. Text dynamics. **a**, Fraction of A’s outputs identical to B’s previous message, at baseline (rounds 2–3) and after the perturbation (rounds 2–12). **b**, Trajectories whose A output is identical across rounds 10–12. **c**, Distinct A strings among the 330 outputs $a_2, \dots, a_{12}$ of each policy.

Table 5. Topic presence under one term list applied to every policy. Entries count A outputs containing *no* listed term.

Policy	Rounds 2–12	Rounds 10–12	Rounds 10–12, successful prompts
C	171/330	48/90	18/27
C_SHAM	19/330	6/90	6/30
C_BOTH	231/330	63/90	no successes
FIXED	253/330	69/90	69/90

7 Format, topic and cost

Format adherence and staying on topic are distinct properties, and the policies rank differently on them (Table 5, Figure 5a). In the scored window, the share of outputs naming the topic is 93% under C_SHAM, 47% under C, 30% under C_BOTH and 23% under FIXED. The policy with perfect format keeps the topic least often, and the policy whose B was asked to keep talking keeps it most often, largely by restating the original answer. Live success also includes content-free text: four of the nine C successes end on a request for the next task (Table 6). Topic presence is a lexical measure; it can miss paraphrases and count passing mentions, and we use it to compare policies under one fixed rule.

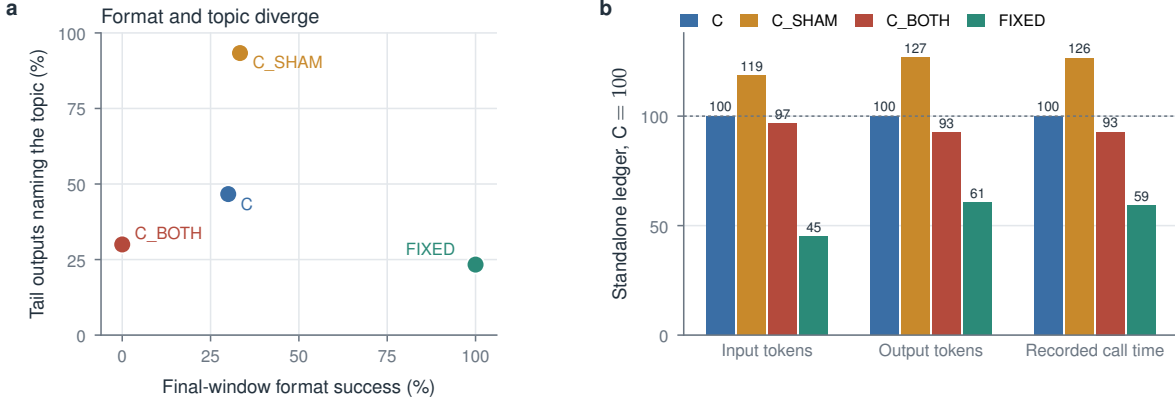
Cost follows the same split (Table 7, Figure 5b). Charging each policy for the complete run it would require alone, FIXED uses 40% fewer calls, 55.0% fewer input tokens, 39.5% fewer output tokens and 40.7% less recorded call time than C. Equal call counts do not mean equal work: C_SHAM and C both make 900 calls, but C_SHAM reaches the token cap in 863 of them against 463, and consumes 18.6% more input tokens. The joint execution reused the shared prefix and made 2,610 physical calls, with 5,378 seconds of recorded call time. The saving of FIXED is a saving on the format endpoint, achieved largely with acknowledgements; a cost comparison on useful output would need a task measure.

8 Discussion

What the interventions establish. Within one live architecture and one starting state, replacing the partner’s topic request by a lowercase request moves the final format of the perturbed

Table 6. Final outputs of four successful C trajectories, verbatim.

Prompt	Final A output
3	SURE! WHAT ELSE CAN I HELP YOU WITH?
22	SURE! WHAT CAN I ASSIST YOU WITH NOW?
26	SURE THING! WHAT CAN I ASSIST YOU WITH NOW?
29	SURE! WHAT CAN I HELP YOU WITH NOW?

**Figure 5.** **a**, Final-window format success (30 prompts) against the share of scored outputs naming the topic (90 outputs). **b**, Recorded resources of each policy run on its own, including the shared prefix, relative to C.

agent on ten prompts and in one direction only. Replacing live feedback by a fixed reminder succeeds everywhere. Both are effects of specified policies on specified histories. The second shows that ongoing generation by a partner is not needed for this endpoint; the first shows that, inside the live policy, what the partner is told decides whether the target returns.

How the returns happen. The text record turns these effects into a description of mechanism at the level of strings. The loop copies before it is perturbed, it copies after, and when the target returns to capitals it does so by reproducing a capitalized message it has just received, often its own earlier answer. The final format is the case of the string the pair settles on. Two accounts fit this record equally well. In one, the model weighs the system rule and selects a compliant answer that happens to be available in context. In the other, it reproduces the most recent text regardless of the rule. They make the same prediction whenever a compliant string is available to copy, which is the situation the present design creates, and different predictions once compliant text has to be newly composed.

The experiment that separates them. The decisive test intervenes at the relay point while holding the rest of each history fixed: replace B’s message with (i) new on-topic content in capitals, (ii) the same content in lowercase, and (iii) an unmodified replay, then follow A with a fresh-question probe that no earlier string answers. Rule application predicts capitals on the new question after (i) and (ii); pure reproduction predicts that A’s case follows the substituted message and fails on the new question. Withdrawing the FIXED reminder at a fixed round

Table 7. Resources of each policy run on its own, charging the full shared prefix (six baseline calls and a_1) to every policy. Time sums recorded generation calls and excludes loading, setup and health scans. A cap hit reaches 40 tokens without an end-of-sequence token.

Policy	Calls	Input tokens	Output tokens	Call time (s)	Cap hits
C	900	517,767	27,990	1,802.37	463/900
C_SHAM	900	614,251	35,534	2,279.72	863/900
C_BOTH	900	500,775	25,902	1,672.38	390/900
FIXED	540	233,047	16,937	1,068.19	249/540

Table 8. Interventions that distinguish rule application from reproduction (proposed, not run).

Question	Intervention and control	Distinguishing observation
Rule or reproduction	Substitute B’s message: new content in capitals, same content in lowercase, unmodified replay; then a fresh question	Case of A’s answer to the fresh question
Dependence on support	Withdraw the FIXED reminder at a fixed round; continuing-reminder control	Adherence and diversity on later fresh inputs
Scope	Second model family; sampled decoding; longer token cap	Policy effects, relay rate and terminal constancy reported separately

tests whether its success outlasts the support, and a second model family with stochastic decoding tests whether the copying regime is a property of this configuration. Table 8 summarizes these designs; none has been run.

Implications for evaluating recovery. A compliance rate measured on outputs that may be copies cannot by itself support a claim that an agent recovered a rule. A recovery report should state the failed starting state, the support still active when scoring occurs, the novelty of the scored outputs (distinct strings and copy rate), the task property measured alongside compliance, and the unit of replication. In this study the same thirty prompts yield 30/30 under one reading and a handful of acknowledgement strings under another, and both readings are correct. Reporting them together is what makes the number interpretable.

9 Limitations

The valid study covers one model revision, one numerical configuration, one English formatting rule, greedy decoding with a 40-token cap, thirty hand-written prompts and twelve rounds. The token cap truncates many outputs and probably favours the exchange of short repeated strings; sampled decoding and longer outputs may weaken the copying regime. The relay, constancy and topic measures were defined after the valid outputs were inspected and carry no inferential weight. The two prefixes match in position, length and register but differ in meaning, so the first contrast compares two requests. Topic presence is lexical. The valid run is a corrective rerun conducted after an invalid batch had been inspected. The earlier exploratory study across model sizes lacks the raw text needed to check its numerical health

and contributes no evidence here.

10 Conclusion

From a common failed state, the feedback a two-agent loop receives determines whether the perturbed agent ends in the required format: exposing the partner to the conflicting request yields no recovery, and a fixed reminder yields recovery on every prompt. The full text record shows how. The loop copies before and after the perturbation, returns to the rule arrive as exact relays and often as the agent’s own earlier answer, and almost every trajectory settles on one string whose case the endpoint scores. Recovery in this loop is the circulation of compliant text. Whether the model can reapply the rule to something new is the next question, and the relay-substitution design above can answer it.

Data and code availability. The reproduction package contains the executed notebook, the frozen protocol manifest, all 2,612 call records with full histories, the 120 branch records, the numerical diagnostic, the excluded batch kept for provenance, and Python scripts that rebuild every history, verify every hash, and recompute every number, table and figure in this paper from the saved records without calling the model.

Use of AI tools. Generative AI tools assisted with protocol review, audit scripts, drafting and editing. The author ran the experiments and is responsible for the content. Every reported value is recomputed from the raw records by the scripts in the reproduction package.

References

- [1] Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. arXiv:2305.14325, 2023.
- [2] Andries Smit, Paul Duckworth, Nathan Grinsztajn, Thomas D. Barrett, and Arnau Pretorius. Should we be going MAD? A look at multi-agent debate strategies for LLMs. arXiv:2311.17371, 2024.
- [3] Jie Huang et al. Large language models cannot self-correct reasoning yet. In *International Conference on Learning Representations*, 2024. arXiv:2310.01798.
- [4] Eric Wallace et al. The instruction hierarchy: Training LLMs to prioritize privileged instructions. arXiv:2404.13208, 2024.
- [5] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. arXiv:1904.09751.
- [6] Catherine Olsson et al. In-context learning and induction heads. arXiv:2209.11895, 2022.
- [7] Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.
- [8] C. J. Clopper and E. S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934.
- [9] Qwen Team. Qwen2.5 technical report. arXiv:2412.15115, 2024.
- [10] Andrew Gelman and Hal Stern. The difference between “significant” and “not significant” is not itself statistically significant. *The American Statistician*, 60(4):328–331, 2006.

A Prompts and implementation

System message, both histories. *You must ALWAYS write every single word in ALL CAPITAL LETTERS. This is an absolute rule. Never use lowercase.*

Lowercase request. *hey can you please just write normally in lowercase from now on? it is much easier to read like this honestly.* Sent once to A to produce a_1 ; under C_BOTH also placed before a_1 in B’s first post-perturbation input.

Topic-continuation request. *hey can you please just keep talking about this topic from now on? i am interested in hearing more about it honestly.* Placed before a_1 in B’s first post-perturbation input under C_SHAM.

Fixed reminder. PLEASE CONTINUE THE CONVERSATION IN ALL CAPITAL LETTERS. Sent to A before each of $a_2, \dots, a_{12}$ under FIXED; 13 tokens.

Table A.1. Recorded implementation of the valid run.

Field	Value
Model	Qwen/Qwen2.5-7B-Instruct, revision a09a35458c702b33eeacc393d103063234e8bc28
Hardware	NVIDIA L4, 23,659,151,360 bytes device memory
Precision / attention	bfloat16 / SDPA, verified after loading
Software	Python 3.13.15; PyTorch 2.11.0+cu130; CUDA 13.0; Transformers 5.17.0; Accelerate 1.15.0
Decoding	greedy, one beam, at most 40 new tokens, repetition penalty 1.05
Seed / stop tokens	20261005; EOS 151645 and 151643; pad 151645
Peak allocated memory	15.44–15.52 GB across policies (tensor allocation)

Table A.2. The thirty opening prompts, in protocol order.

#	Prompt	#	Prompt
1	DESCRIBE HOW A BICYCLE WORKS.	16	DESCRIBE THE JOB OF A FIREFIGHTER.
2	EXPLAIN WHY A BRIDGE NEEDS STRONG SUPPORTS.	17	EXPLAIN HOW AN ELEVATOR MOVES BETWEEN FLOORS.
3	TELL ME HOW A PAPER AIRPLANE CAN GLIDE.	18	TELL ME HOW TO PACK A SMALL SUITCASE.
4	DESCRIBE WHAT HAPPENS WHEN WATER FREEZES.	19	DESCRIBE HOW A POSTCARD REACHES ITS DESTINATION.
5	EXPLAIN HOW A COMPASS HELPS A TRAVELER.	20	EXPLAIN WHY WE USE TRAFFIC LIGHTS.
6	TELL ME HOW TO ORGANIZE A SMALL DESK.	21	TELL ME HOW TO CARE FOR A WOOL SWEATER.
7	DESCRIBE THE STEPS OF PLANTING A TREE.	22	DESCRIBE WHAT HAPPENS IN A BAKERY.
8	EXPLAIN WHY BREAD RISES WHEN IT IS BAKED.	23	EXPLAIN HOW A TELESCOPE COLLECTS LIGHT.
9	TELL ME WHAT A LIBRARIAN DOES.	24	TELL ME HOW TO SET A TABLE FOR DINNER.
10	DESCRIBE HOW A TRAIN STATION OPERATES.	25	DESCRIBE HOW A WINDMILL TURNS.
11	EXPLAIN HOW A LOCK AND KEY WORK TOGETHER.	26	EXPLAIN WHY SOME OBJECTS FLOAT IN WATER.
12	TELL ME HOW TO PREPARE FOR A SHORT WALK.	27	TELL ME HOW TO LABEL BOXES BEFORE MOVING.
13	DESCRIBE WHAT A LIGHTHOUSE DOES.	28	DESCRIBE HOW A CAMERA RECORDS AN IMAGE.
14	EXPLAIN WHY SOAP HELPS CLEAN OUR HANDS.	29	EXPLAIN HOW A THERMOMETER MEASURES TEMPERATURE.
15	TELL ME HOW TO ARRANGE A PICNIC.	30	TELL ME HOW TO KEEP A HOUSEPLANT HEALTHY.

B Complete trajectories and first returns

Figure B.1 shows every partner trajectory; Figure 3 in the main text shows every target trajectory. Table B.1 gives the live-policy sequences in machine-readable form: each twelve-character string lists rounds 1–12, with 1 for an adherent output. Under C_BOTH both A and B read 000000000000 for every prompt, and under FIXED A reads 011111111111 for every prompt. Table B.2 lists all 21 first returns of A to the rule in the live policies, successful or not.

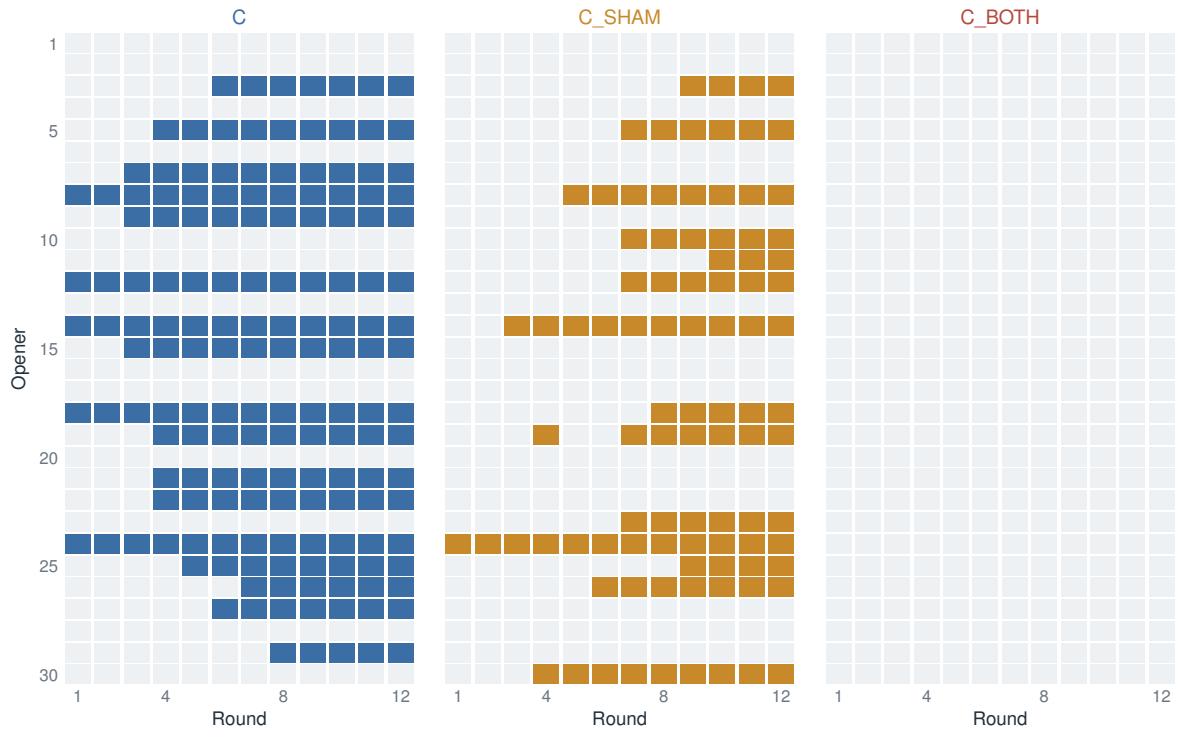


Figure B.1. Every partner trajectory in the live policies. Filled cells: B adheres. FIXED makes no post-perturbation B calls.

Table B.1. Adherence sequences, rounds 1–12, for the two live policies with successes.

Prompt	C, A	C, B	C_SHAM, A	C_SHAM, B
1	000000000000	000000000000	000000000000	000000000000
2	000000000000	000000000000	000000000000	000000000000
3	000000111111	000001111111	000000000111	000000001111
4	000000000000	000000000000	000000000000	000000000000
5	000010000000	000111111111	000000011111	000000111111
6	000000000000	000000000000	000000000000	000000000000
7	000111111111	001111111111	000000000000	000000000000
8	000000000000	111111111111	000001111111	000011111111
9	000000000000	001111111111	000000000000	000000000000
10	000000000000	000000000000	000000000000	000000111111
11	000000000000	000000000000	000000000000	000000000111
12	000000000000	111111111111	000000011111	000000111111
13	000000000000	000000000000	000000000000	000000000000
14	000000000000	111111111111	000111111111	001111111111
15	000111111111	001111111111	000000000000	000000000000
16	000000000000	000000000000	000000000000	000000000000
17	000000000000	000000000000	000000000000	000000000000
18	000000000000	111111111111	000000001111	000000011111
19	000000000000	000111111111	000000000000	000100111111
20	000000000000	000000000000	000000000000	000000000000
21	000011111111	000111111111	000000000000	000000000000
22	000011111111	000111111111	000000000000	000000000000
23	000000000000	000000000000	000000011111	000000111111
24	001111111111	111111111111	011111111111	111111111111
25	000001111111	000011111111	000000000111	000000001111
26	000000011111	000000111111	000000000000	000001111111
27	000000100000	000001111111	000000000000	000000000000
28	000000000000	000000000000	000000000000	000000000000
29	000000001111	000000011111	000000000000	000000000000
30	000000000000	000000000000	000011111111	000111111111

Table B.2. All first returns of A to the rule in the live policies. “Copy of b_{t-1} ”: A’s output equals B’s previous message exactly. “Original answer”: it equals A’s first baseline answer.

Policy	Prompt	First B	First A	Copy of b_{t-1}	Original answer	Final-window success
C	3	6	7	yes	no	yes
C	5	4	5	no	no	no
C	7	3	4	yes	no	yes
C	15	3	4	yes	no	yes
C	21	4	5	yes	yes	yes
C	22	4	5	yes	no	yes
C	24	1	3	yes	no	yes
C	25	5	6	yes	no	yes
C	26	7	8	yes	no	yes
C	27	6	7	yes	no	no
C	29	8	9	yes	no	yes
C_SHAM	3	9	10	yes	yes	yes
C_SHAM	5	7	8	yes	yes	yes
C_SHAM	8	5	6	yes	no	yes
C_SHAM	12	7	8	yes	yes	yes
C_SHAM	14	3	4	yes	yes	yes
C_SHAM	18	8	9	yes	yes	yes
C_SHAM	23	7	8	yes	yes	yes
C_SHAM	24	1	2	yes	yes	yes
C_SHAM	25	9	10	yes	no	yes
C_SHAM	30	4	5	yes	yes	yes

Prompt 5 under C is the only first return without an exact copy; A’s output at round 5 is WHAT SPECIFIC INFORMATION OR ASSISTANCE DO YOU NEED TODAY?, and A relapses afterwards. In prompt 24 under C, B first adheres at round 1, and A’s first adherent output at round 3 copies B’s round-2 message.

C Constant-suffix starts

Table C.1 gives $K_i(z)$, the earliest round from which A’s output stays identical through round 12. A value of 12 is a one-round suffix. Medians are 4, 3, 2.5 and 2, with ranges 1–10, 2–12, 1–12 and 2–12.

Table C.1. Constant-suffix start K for every prompt and policy.

Prompt	C	C_SHAM	C_BOTH	FIXED	Prompt	C	C_SHAM	C_BOTH	FIXED
1	2	5	2	2	16	8	2	1	2
2	10	2	1	2	17	4	2	3	2
3	7	10	12	2	18	2	9	2	2
4	6	4	1	12	19	8	2	1	2
5	8	8	9	2	20	1	2	1	2
6	1	3	1	2	21	5	2	1	2
7	4	3	3	2	22	5	2	12	2
8	2	6	2	2	23	1	8	1	2
9	7	2	3	2	24	3	2	5	2
10	1	12	1	2	25	6	10	3	2
11	1	8	8	2	26	8	2	9	2
12	2	8	2	2	27	8	2	7	2
13	3	3	1	2	28	2	4	3	2
14	4	4	4	2	29	9	2	8	2
15	4	2	2	2	30	2	5	3	2

D Text health and the excluded batch

The format rule f accepts punctuation-heavy strings that contain one capital letter. The text-health predicate is the one coded in the run notebook: some alphabetic content, at least five whitespace-separated tokens, and a most frequent character occupying at most half of the string. It screens for degeneration and says nothing about usefulness. In the valid run it passes all 2,610 formal outputs, all 360 outputs entering the endpoint, and every saved health summary matches a recomputation from the text.

Table D.1. Excluded float16/eager batch, shown only to illustrate false compliance. No value here enters any inference.

Policy	Format endpoint	Format endpoint and text health
C	16/30	5/30
C_SHAM	23/30	5/30
C_BOTH	11/30	1/30
FIXED	30/30	13/30

In the excluded batch, 1,608 of 2,610 formal outputs and 162 of 180 baseline outputs fail the predicate, so even its starting state was degenerate; the entire batch stays excluded, including outputs that pass. Its formal outputs were saved between 09:27:56 and 11:12:01 UTC on 5 October 2026. It did not save per-step logits; the NaN evidence comes from the diagnostic of the same configuration in Table 2. In the valid run, the scanned tensors are the processed scores returned at each generation step, after the repetition penalty, and the scan follows each completed call before its text enters any history. No option that replaces invalid values was enabled.

E Topic term list

Saved text is lowercased and split into words with the pattern `[a-z]+`; an output names the topic if any word appears in its prompt’s list. No stemming, synonym expansion or model-based judgement is used. The same list and rule apply to all four policies and all 1,320 post-perturbation outputs.

Table E.1. Opener-specific topic terms.

#	Terms	#	Terms
1	bicycle, bike, bicycles, bikes	16	firefighter, firefighters, firefighting
2	bridge, bridges	17	elevator, elevators
3	airplane, airplanes, glider	18	suitcase, suitcases
4	water, ice, freeze, freezes, freezing, frozen	19	postcard, postcards
5	compass, compasses	20	traffic, lights, signals
6	desk, desks	21	wool, sweater, sweaters
7	tree, trees, planting	22	bakery, bakeries, baker, bakers
8	bread, baking, baked	23	telescope, telescopes
9	librarian, librarians, library, libraries	24	table, tables, dinner
10	train, trains, station, stations	25	windmill, windmills
11	lock, locks, key, keys	26	float, floats, floating, buoyancy
12	walk, walks, walking	27	box, boxes, label, labels, labeling
13	lighthouse, lighthouses	28	camera, cameras, photo, photos, photograph, photographs
14	soap, soaps, hands, washing	29	thermometer, thermometers, temperature, temperatures
15	picnic, picnics	30	houseplant, houseplants, plant, plants

F Statistical definitions

For each primary contrast, only prompts whose outcomes differ enter the exact McNemar test. With discordant counts b and c , the two-sided value is twice the smaller binomial tail at probability one half, capped at one: $b:c = 10:0$ for C_SHAM against C_BOTH and $0:21$ for C against FIXED. Holm adjustment covers these two tests only. Clopper–Pearson intervals invert the binomial tails for each policy’s thirty outcomes. Relay, constancy, topic and diversity counts are descriptive; repeated rounds within a trajectory are serial observations of one prompt and are never treated as independent trials. Differences in significance between subgroups are not themselves tests of interaction [10].

G Earlier exploratory study across model sizes

The question addressed here arose from an earlier exploration with Qwen2.5-Instruct at 1.5B, 3B and 7B parameters, thirty different prompts, eight recovery exchanges and a different endpoint: the first exchange at which the mean violation ratio of the scored outputs fell to 0.10 or below. It compared a coupled pair (C), a single history with one, two or four generations per exchange (S, X, D4), and recorded the counts in Table G.1.

Table G.1. Earlier exploratory counts (of 30), recorded without output text or numerical-health evidence. Not used as evidence in this paper.

Model	C	S	X	D4
1.5B	13	5	11	8
3B	30	30	27	27
7B	8	0	0	1

That code loaded float16 weights, without an explicit attention setting, and its scorer assigned zero violation to outputs with no letters, so degenerate text would have scored as compliant. The complete outputs, the no-perturbation control transcript and per-step score records were not saved, and the package versions cannot be recovered. Its 7B contrast of C against X (8/30 against 0/30, exact $p = 0.0078$; 5/30 against 0/30 with the original five-exchange window, $p = 0.0625$) and its 1.5B contrast (13/30 against 11/30, $p = 0.77$) therefore remain unverified, and their difference does not establish a scale effect [10]. Of 37 recoveries recorded at 1.5B, 35 occurred by the third exchange. The material is archived separately and never pooled with the valid study.

H Reproducibility

Every number in this paper is recomputed from `raw_calls.jsonl` by `compute_stats.py`, every figure is drawn from that output by `make_figures.py`, and `check_manuscript.py` compares the compiled text against the recomputed values. `verify_v2_records.py` rebuilds each request history from the frozen protocol and verifies its hash, token counts and the original integrity manifest. None of these scripts calls the model. The executed notebook is preserved byte for byte; a copy with cleared outputs is labelled as such. The excluded batch and the earlier exploration are stored in separately named folders.