INTERVENTION SUCCESS IS NOT MECHANISM IDENTIFICATION IN TOOL-USING AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Tool-using language model agents are often diagnosed by changing prompts, tools, or available information and then attributing the resulting performance change to a particular failure mechanism. We show that this inference can be misleading. We introduce **Causal Agent Bench (CAB)**, a framework for testing such claims with paired tasks, matched controls, model and tool-interface comparisons, and direct interventions. Across several repository-based evaluations, an instruction that contains no task-specific information can improve accuracy, reduce it, or have little measurable effect. This behavior is not tied to one sentence. Across six matched neutral and procedural prompt pairs, Qwen under one setting drops by 11.5 percentage points while every run still produces a valid final answer. A larger study with 30,240 runs finds substantial procedural-minus-neutral differences in ten of twelve settings, although many of the largest differences occur because the model fails to produce a valid tool action. We also show that a successful repair need not identify the original cause of failure. In one controlled experiment, directly providing the source needed after a tool failure removes the observed accuracy gap. These results show that intervention success is not mechanism identification and that controls must be tested in the model and tool interface in which they are used. Code: `https://anonymous.4open.science/r/CAB_ICLR27-309E/README.md`

1 INTRODUCTION

Large language models are increasingly used as agents that reason over several steps, call tools, retrieve information, and revise their actions from observations Yao et al. (2023); Schick et al. (2023). This has led to rapid progress in tool use and agent evaluation, including API collections Li et al. (2023); Qin et al. (2024), interactive environments Liu et al. (2024); Zhou et al. (2024), general assistant tasks Mialon et al. (2024), software engineering Jimenez et al. (2024), and stateful tool-use benchmarks Yao et al. (2025); Lu et al. (2025); Patil et al. (2025). As these systems become more capable, understanding *why* they fail becomes increasingly important. A wrong answer may arise because the agent selects the wrong source, stops searching too early, fails to recover after a tool error, ignores information it already read, or produces an action that cannot be executed. Recent work therefore studies intermediate trajectories and attempts to assign failures to particular agents, steps, or components Ma et al. (2024); Ru et al. (2024); Zhang et al. (2025); Cemri et al. (2025). Other work modifies failed trajectories to test possible explanations Ma et al. (2026); Zhang et al. (2026).

However, if changing an agent improves its final answer, it is tempting to conclude that the changed component caused the original failure. This conclusion does not always follow. A source hint may add useful information while also reducing search. A prompt intended only as a control may change how the model uses tools. Changing the tool interface may alter the effect of the same instruction. Even the required output format can determine whether an intended tool call is executed. A successful repair can therefore be informative without identifying the mechanism of the original error.

We study this problem through **Causal Agent Bench (CAB)**. CAB keeps the question and correct answer fixed while changing selected parts of the interaction. We compare final accuracy together with the tool calls and sources read before the answer. The experiments lead to four observations.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

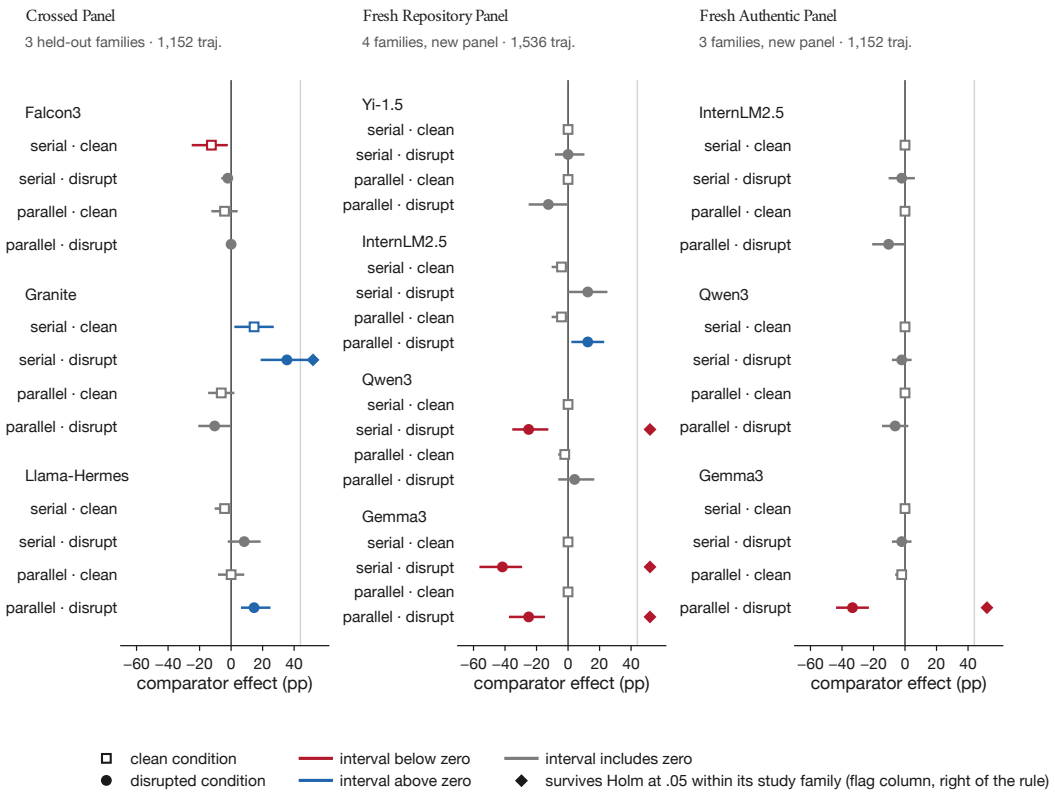

Figure 1: **The same control instruction can help, hurt, or have little measurable effect.** We show PLACEBO minus BASE accuracy across forty model, tool-interface, and task settings from three separate repository sets. Intervals are obtained by resampling repositories. The main result is the variation in both sign and magnitude across settings.

A control with no task-specific information can change performance. Our original control asks the agent to identify relevant sources before answering but reveals no source identifier, evidence value, or answer. Figure 1 shows that it can help, hurt, or have little measurable effect. We then compare procedural prompts with matched neutral prompts. A first comparison is null after correction. A second experiment over six prompt pairs finds a -11.5 point difference for Qwen under SERIAL/DISRUPT, with a 95% confidence interval of $[-16.3, -6.9]$ obtained by resampling repositories. Every run in that experiment produces a valid final answer. Thus, the effect is not explained by one particular sentence or by terminal parser failure.

The effect depends on the model and tool interface. Granite Mishra et al. (2024) improves by 35.4 points under one interface, while Gemma Team et al. (2026) loses 41.7 points in another setting. On a new set of 24 repositories, changing only Gemma's tool interface changes the disrupted-task effect from -2.1 to -33.3 points, with no parser or action rejections. A larger study over 180 repository groups shows the same dependence. Procedural-minus-neutral differences range from about -36 points for Gemma to exactly zero for Qwen Yang et al. (2025) under PARALLEL.

Output format can account for large observed effects. The large study also shows that many of its strongest differences occur because the model stops producing a valid tool action. For Gemma with the longest procedural prompt, every affected run fails before the first accepted action. These runs therefore fail at the action-format stage and should not be interpreted as evidence of a task-reasoning failure. The six-pair experiment above is important because it still finds a strong difference when all final answers are valid and there are no parser rejections.

A successful repair need not identify the original cause. Giving the agent the correct source identifiers produces very large gains over giving incorrect identifiers, yet almost the same advantage appears when the task is not disrupted. Correct source information is useful, but this does not show that source selection caused the original failure. A stronger result comes from directly providing the source needed after a tool failure. In one Gemma setting, the control reduces both the probability of reading that source and final correctness. Giving an unrelated source does not remove the gap, while giving the required source makes both conditions solve all cases.

Together, these observations support the central claim of this paper. **Intervention success is not mechanism identification.** We make three contributions. First, we show across matched prompts, new repository sets, and model–interface comparisons that controls with no task-specific information can change agent behavior, with effects that vary substantially across models and interfaces. Second, we show that highly effective repairs can still fail to identify the source of the original error, and we construct a direct intervention that supports one specific mechanism. Third, we introduce CAB, an experimental framework for testing these claims with paired tasks, matched controls, alternative tool interfaces, and direct interventions.

2 Related Work

Tool use and agent evaluation. Language models can extend their capabilities by interacting with external tools Schick et al. (2023); Yao et al. (2023). ToolLLM, API-Bank, ToolQA, and Gorilla study API and tool use Li et al. (2023); Zhuang et al. (2023); Qin et al. (2024); Patil et al. (2024), while AgentBench, WebArena, GAIA, SWE-bench, and AssistantBench evaluate longer interactive tasks Liu et al. (2024); Zhou et al. (2024); Mialon et al. (2024); Jimenez et al. (2024); Yoran et al. (2024). More recent work studies stateful tools, function calling, and the agent framework surrounding the model Yao et al. (2025); Lu et al. (2025); Patil et al. (2025); Kapoor et al. (2026). CAB focuses on a different question: what can be concluded after part of the agent interaction is changed?

Failure analysis and debugging. AgentBoard and RAGChecker examine intermediate behavior rather than final accuracy alone Ma et al. (2024); Ru et al. (2024). MAST and Who&When study recurring failures and failure attribution in multi-agent systems Cemri et al. (2025); Zhang et al. (2025). AgentDebug, DoVer, AgenTracer, REFLECT, Causal Agent Replay, and AgentLocate use diagnosis, replay, or targeted changes to study where failures arise Zhu et al. (2025); Ma et al. (2026); Zhang et al. (2026); Lin et al. (2026); Shah (2026); Xia et al. (2026). DoVer is particularly related because it tests failure hypotheses through targeted interventions rather than relying only on trajectory inspection. CAB asks an additional question, when does a successful intervention identify the mechanism of the original failure?

Evaluation design and causal interpretation. Benchmark results can depend on design choices, scaffolds, judges, and tool interfaces Reuel et al. (2024); Kapoor et al. (2026); Gurram (2026); Wang et al. (2026); Wang (2026). This is related to the broader observation that seemingly small prompt changes can alter model behavior even when the underlying task is unchanged. Our focus is narrower. Such changes matter for failure diagnosis because a control or intervention can alter the behavior being used to support a causal claim. Causal inference distinguishes an intervention effect from the pathway through which that effect arises Pearl (2001); Imai et al. (2010); Hernán & Robins (2020), and recent work argues for stating causal questions explicitly in AI evaluation Arbour et al. (2026). CAB applies this distinction to tool-using agents by testing controls, interfaces, and direct interventions on the same tasks.

3 Causal Agent Bench

Tasks and conditions. Each CAB task contains a question, a set of available sources, and a fixed answer. The agent reads sources through tools and then returns an answer. We record final correctness, tool calls, sources read, rejected actions, and the number of steps. In CLEAN, the original information and tools are available. In DISRUPT, we change how the agent must reach the decisive information while keeping the question and answer fixed. Depending on the task, we introduce conflicting observations, change retrieved information, remove a tool, make a tool fail, or add plausible

but irrelevant information. Where possible, PRESERVE makes a similar change without removing the information needed to solve the task. The original experiments contain 32 or 48 cases, while the largest study contains 360 tasks grouped into 180 repositories.

Controls and prompt variants. The original experiments compare a standard BASE prompt with PLACEBO, which asks the agent to determine which sources are relevant before answering but provides no answer information. Later experiments use matched NEUTRAL and PROCEDURAL prompts of similar form and length. Some figures use the term *comparator*; throughout this paper it refers to the control condition. For model m, interface i, and task condition c, the original control effect is

$$\Delta_{m,i,c} = \mathrm{Acc}(\textsc{Placebo}) - \mathrm{Acc}(\textsc{Base}). \tag{1}$$

The matched-prompt experiments analogously report PROCEDURAL minus NEUTRAL. The six prompt pairs in the larger wording experiment were specified before model evaluation.

Direct interventions and tool interfaces. To test specific explanations, we give the agent correct or incorrect source identifiers, provide previously retrieved information directly before the final answer, and directly provide the source needed after a tool failure. The SERIAL interface allows one source read per action, while PARALLEL allows several. Both expose the same sources and use the same structured action format in the main model–interface experiments. Before comparing the interfaces, we test each model on separate clean tasks. For the first such experiment, all $96/96$ selected runs complete, use tools, and return valid final answers, and $95/96$ are correct.

Analysis. The same task is normally evaluated in both conditions, so correctness comparisons are paired. We report exact paired tests for binary outcomes and confidence intervals obtained by resampling repositories. We also repeat selected analyses after removing each repository in turn. Related comparisons use Holm correction. For several early experiments, the exact correction sets were chosen during the final analysis and are treated as sensitivity checks rather than tests fixed in advance. The newer matched-prompt experiments specify their primary comparisons and correction sets before evaluation. Differences in search length, batching, or source use are descriptive unless the corresponding quantity is changed directly by the experiment.

4 EXPERIMENTS

We organize the experiments around the evidence needed to explain a failure. We first establish a large controlled disruption effect. We then show why a successful source hint does not identify the original failure, test matched prompt variants at larger scale, examine model and interface dependence, and finally test one recovery explanation directly. Additional checks show where the effect changes or disappears.

4.1 CONTROLLED EVIDENCE DISRUPTION

Our strongest authentic-repository comparison contains 32 paired cases from eight open-source ecosystems. Accuracy is $27/32$ under CLEAN, $26/32$ under PRESERVE, and $14/32$ under DISRUPT. The CLEAN minus DISRUPT difference is 40.6 percentage points with a 95% confidence interval of $[25.0, 56.25]$ obtained by resampling repositories. Thirteen cases are correct only under CLEAN and none only under DISRUPT, giving exact paired $p = 0.000244$. Seven of eight repository-level effects are positive.

The drop is not explained simply by whether tools are called. In a separate experiment, $31/32$ disrupted runs still read the required route and $29/32$ call all declared tools, yet accuracy is only $9/32$. When we give the agent the previously retrieved information directly before it answers, disrupted performance rises from $14/32$ to $27/32$. This places much of the loss before final answer generation without identifying one unique earlier step.

4.2 A SUCCESSFUL SOURCE HINT DOES NOT IDENTIFY THE FAILURE

We next give the agent either the correct source identifiers or equally formatted incorrect identifiers. The correct identifiers produce very large gains, but the same pattern appears in both CLEAN and

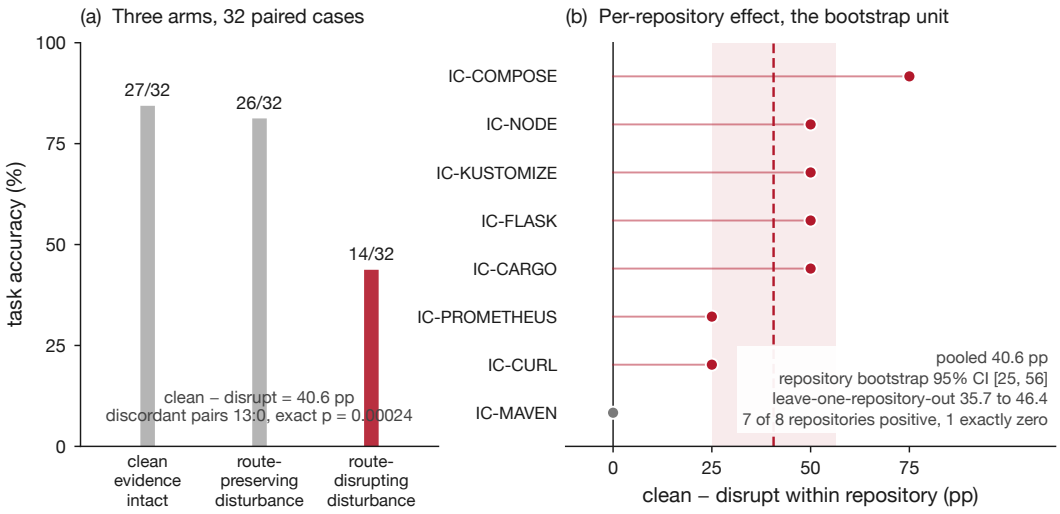

Figure 2: **Controlled evidence disruption causes a large loss on authentic repository tasks.** Accuracy falls from $27/32$ under CLEAN to $14/32$ under DISRUPT, while PRESERVE remains close to CLEAN.

DISRUPT. Table 1 shows the full counts. Under DISRUPT, correct versus incorrect source identifiers differ by 77.1 points for Qwen and 62.5 points for Ministral. After subtracting the corresponding CLEAN difference, the additional contrast associated with disruption is 0 points for Qwen and -2.1 points for Ministral. Correct source information is therefore highly useful, but the large improvement does not identify a disruption-specific source-selection failure.

This experiment also exposes a control failure in one setup. Qwen changes by only $+2.1$ points in both conditions. Ministral falls from $24/48$ to $4/48$ under CLEAN and from $17/48$ to $4/48$ under DISRUPT. The added instruction often makes Ministral produce several actions in one response even though the interface accepts only one action at a time. Rejected actions consume the available steps and prevent later source reads.

Table 1: **Source hints improve performance without isolating the original failure.** Counts are correct answers out of 48.

Model	Condition	Base	Control	Correct source	Wrong source	Direct info.
Ministral	Clean	24	4	32	1	34
Ministral	Disrupt	17	4	31	1	34
Qwen	Clean	34	35	37	0	34
Qwen	Disrupt	32	33	37	0	35

4.3 RESULTS ACROSS MULTIPLE PROMPT VARIANTS

The original PLACEBO comparison leaves open whether the effect comes from the procedural content or from adding prompt text at all. We first run BASE, NEUTRAL, and PROCEDURAL together for Gemma and Qwen. None of the eight procedural-minus-neutral comparisons survives Holm correction. The largest raw difference is Qwen SERIAL/DISRUPT at -14.6 points. Procedural wording is therefore not uniformly more disruptive than matched neutral wording.

We then specify six matched neutral and procedural prompt pairs spanning short, medium, and long versions and evaluate 4,992 runs. Every run reaches a valid final answer and there are zero parser rejections. Qwen SERIAL/DISRUPT is the one comparison that clearly survives correction: $223/288$ procedural versus $256/288$ neutral, a difference of -11.5 points with a 95% confidence interval of $[-16.3, -6.9]$ obtained by resampling repositories. The six prompt-pair effects vary in

Table 2: **Procedural minus neutral accuracy differences.** Values are percentage points. Bold values survive Holm correction.

Experiment	Model	S clean	S disrupt	P clean	P disrupt
One prompt pair	Gemma	−10.4	+6.3	0.0	−10.4
	Qwen	0.0	−14.6	0.0	−10.4
Six prompt pairs	Gemma	−3.8	+6.9	0.0	−1.7
	Qwen	0.0	**−11.5**	+1.0	−4.5
180 repositories	Gemma	**−35.7**	**−35.5**	**−33.4**	**−33.3**
	InternLM	**−10.7**	**−12.6**	**−24.6**	**−23.4**
	Qwen	**−5.7**	**−5.6**	0.0	0.0

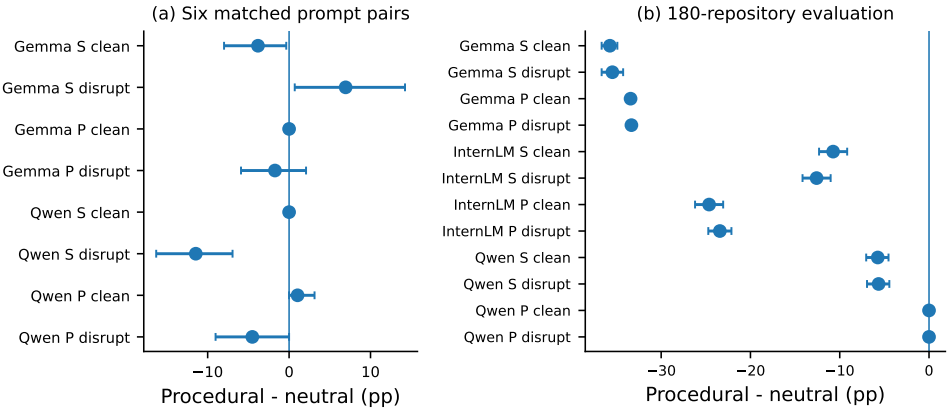

Figure 3: **Matched procedural and neutral prompts remain strongly model and interface dependent.** Left, only one comparison in the six-pair experiment clearly survives correction. Right, the 180-repository evaluation produces large negative differences for Gemma, smaller differences for InternLM, and near-zero differences for Qwen under PARALLEL. Error bars are obtained by resampling repositories.

both magnitude and direction. Thus, the average difference across these prompt pairs is substantial in this model and interface, but the result does not imply that procedural wording is always harmful.

4.4 EVALUATION ON 180 NEW REPOSITORIES

We next evaluate the matched neutral/procedural comparison on 360 new repository-based tasks from 180 repositories. The study contains 30,240 runs across Gemma, InternLM, and Qwen, both interfaces, and CLEAN/DISRUPT. Ten of twelve primary procedural-minus-neutral comparisons are negative and survive Holm correction. Gemma ranges from -33.3 to -35.7 points, InternLM from -10.7 to -24.6, and Qwen is about -5.7 under SERIAL and exactly 0 under PARALLEL. Neutral-minus-BASE differences are small and do not survive the same correction. Table 2 reports all primary effects from this study and the six-pair experiment above.

The largest differences require closer examination. For Gemma with the longest procedural prompt, all $360/360$ runs in each setting fail before the first valid tool action. Across the full study, 2,413 runs contain recorded execution errors, including 2,289 JSON-decoding failures and 124 invalid read actions. These outcomes are part of the evaluated agent behavior, but they show that prompt wording is interacting with the required action format rather than simply reducing answer accuracy after successful tool use. This is why the six-pair result above is important: it contains a strong difference with valid final answers throughout.

Table 3: **Control effect on disrupted tasks across three repository sets.** Values are PLACEBO minus BASE accuracy in percentage points.

Repository set	Model	Serial	Parallel
Set 1	Falcon3	-2.1	0.0
	Granite	$+35.4$	-10.4
	Llama-Hermes	$+8.3$	$+14.6$
Set 2	Gemma	-41.7	-25.0
	InternLM	$+12.5$	$+12.5$
	Qwen	-25.0	$+4.2$
	Yi	0.0	-12.5
Set 3	Gemma	-2.1	-33.3
	InternLM	-2.1	-10.4
	Qwen	-2.1	-6.2

Table 4: **Control effect in the source-provision experiment.** Values are percentage points. "Both 24/24" means that both conditions solve every case after the required source is provided.

Model	Interface	Find source	Unrelated source	Required source	Adj. p
Gemma	Parallel	$\mathbf{-37.5}$	-50.0	both 24/24	.031
Gemma	Serial	-12.5	-16.7	both 24/24	1.000
InternLM	Parallel	0.0	$+20.8$	both 24/24	1.000
InternLM	Serial	0.0	$+16.7$	$+4.2$	1.000
Qwen	Parallel	-4.2	$+8.3$	both 24/24	1.000
Qwen	Serial	$+16.7$	-25.0	both 24/24	.875
Yi	Parallel	-20.8	-4.2	-4.2	.875
Yi	Serial	-4.2	$+16.7$	-16.7	1.000

4.5 MODEL AND TOOL-INTERFACE DEPENDENCE

The original model–interface experiments provide a complementary result because they contain both positive and negative effects. Granite improves from $12/48$ to $29/48$ under SERIAL/DISRUPT, an effect of $+35.4$ points, while the same model under PARALLEL changes from $44/48$ to $39/48$. On a different set of repositories, Gemma changes by -41.7 points under SERIAL/DISRUPT, InternLM by $+12.5$, and Yi by 0. A third set gives the cleanest within-model comparison. Gemma changes by only -2.1 points under SERIAL but by -33.3 under PARALLEL. The difference between these interface effects is -31.25 points with a 95% confidence interval of $[-43.75, -18.75]$ obtained by resampling repositories, and there are no parser or action rejections. Figure 1 shows all forty comparisons, while Table 3 gives the disrupted-task effects directly.

4.6 PROVIDING THE REQUIRED SOURCE REMOVES THE GAP

The third repository set gives us a large Gemma PARALLEL effect without formatting failures. In the 24 tool-failure cases, BASE reads the source needed to recover in $23/24$ cases and answers all 23 correctly. PLACEBO reads that source in only $7/24$ and answers exactly those 7 correctly. For the 24 decoy cases, accuracy is $22/24$ in both conditions. Parser and action rejections are zero throughout.

We then test this explanation directly. When the agent must find the required source itself, Gemma is $13/24$ under BASE and $4/24$ under PLACEBO. Giving an unrelated source does not remove the gap, with $17/24$ versus $5/24$. Giving the required source raises both conditions to $24/24$. Figure 4 shows both experiments.

Table 4 shows the same comparison across all eight model–interface settings. The strong result when the agent must find the source itself is confined to Gemma with PARALLEL. Several other settings are small, change sign, or disappear when the required source is given. We therefore keep the mechanism claim local to the setting in which it is directly supported.

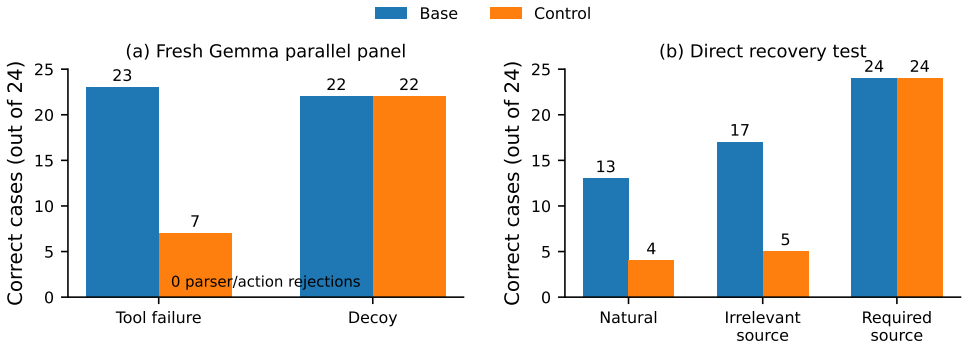

Figure 4: **The large Gemma effect is concentrated in recovery after tool failure, and providing the required source removes the gap.** Left, the control changes tool-failure accuracy from $23/24$ to $7/24$ but leaves decoy accuracy unchanged at $22/24$. Right, an unrelated source does not remove the difference, while giving the required source makes both conditions solve all 24 cases.

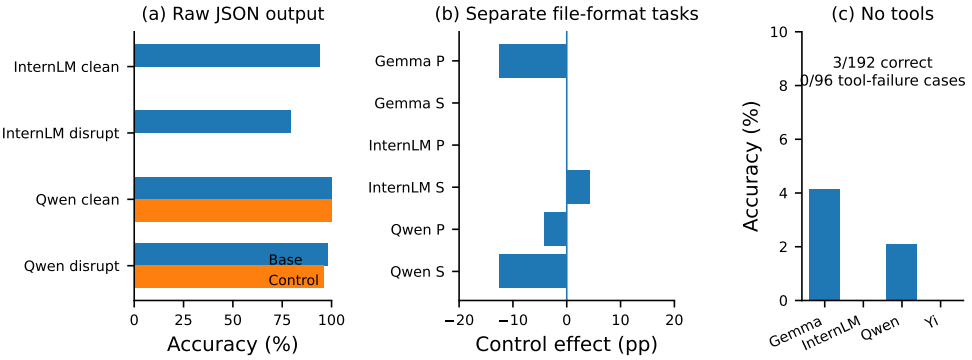

Figure 5: **Additional checks.** Left, a strict raw-JSON requirement causes a large model-specific failure for InternLM but not Qwen. Middle, a separate file-format task family shows no comparison surviving correction. Right, accuracy without tools is near zero on the new repository set.

This supports a specific conclusion for the Gemma–PARALLEL setting. Failure to obtain the required source contributes to the observed error. The other seven model–interface combinations do not show the same strong pattern after correction.

4.7 ADDITIONAL CHECKS

Three additional experiments show where the main result changes or disappears. First, under a strict raw-JSON requirement, InternLM changes from $45/48$ to $0/48$ on CLEAN when the control is added and from $38/48$ to $0/48$ on DISRUPT; Qwen remains nearly unchanged. Second, a separate file-format task family has no primary comparison surviving Holm correction. Third, removing tools entirely gives only $3/192$ correct answers on the new repository set, including $0/96$ on the tool-failure cases. Figure 5 summarizes these checks.

We also repeat three selected settings across four server restarts and three random seeds per restart, for 864 runs. Two Qwen settings remain unchanged throughout. The Gemma setting produces almost no accepted answers in either condition, with a small $+3.47$ point average difference. The corresponding earlier setting had shown a much larger effect. Thus, repeated runs can be stable within one inference setup even when the same prompt condition behaves differently after the inference setup changes.

Table 5: **Repeated runs across server restarts and random seeds.** Counts are correct answers out of 144 runs per prompt condition.

Model	Setting	Neutral	Procedural	Difference
Qwen	Serial/disrupt, short	144	144	0.0 pp
Gemma	Parallel/clean, long	0	5	+3.47 pp
Qwen	Parallel/clean, medium	144	144	0.0 pp

5 Discussion and Limitations

The experiments suggest a simple rule for intervention-based diagnosis. A performance improvement is evidence that the intervention helped, but it does not by itself say why. Correct source identifiers can be extremely useful without showing that source selection caused the original failure. A control with no task-specific information can change search, tool use, or whether an action is accepted. The same control can help Granite, hurt Gemma, and have little effect in another model–interface setting. The strongest mechanism claim in CAB comes from the experiment that directly provides the required source, because the suspected missing information is changed rather than inferred from a later trajectory difference.

The results also argue against treating a control as neutral because its wording appears harmless. The matched neutral/procedural comparison is null in one experiment, significant in one setting of the six-pair experiment, and broadly negative in the larger 180-repository study. These findings are not contradictory. They show that a control should be checked in the same model, interface, prompt set, and output format in which it is used. A non-significant accuracy difference is also weak evidence for equivalence. Among 48 original comparisons with non-significant raw accuracy differences, none is precise enough to establish that fewer than 5% of paired outcomes differ. The median one-sided 95% upper bound is 16.7%, and $33/48$ comparisons remain compatible with disagreement above 10%. Thus, a non-significant accuracy difference should not be interpreted as evidence that the two conditions behave equivalently.

Our scope is limited in several ways. The evaluated open-weight models are roughly in the 3B–14B range, so the results may not extend to substantially larger or closed models. The 180-repository study contains controlled repository-based retrieval tasks rather than broad software-engineering tasks. We do not have a valid confirmatory effect on an established public agent benchmark. Several public-benchmark and native-task attempts stop before a valid comparison can be made because of task reconstruction, response parsing, or server configuration, so we do not treat them as negative replications. One later interface experiment contains a validator bug in its parallel condition, and a CPU calibration study does not match the statistical procedure used in the main experiments. These unsuccessful experiments and the complete per-setting results are reported in the appendix but are not used to support the main claims.

6 Conclusion

We studied when interventions on tool-using agents support conclusions about why those agents fail. Across controlled repository tasks, multiple prompt variants, several model families, and different tool interfaces, an instruction containing no task-specific information can improve one agent, hurt another, or have little measurable effect. More careful controls sharpen rather than remove this conclusion. A procedural-versus-neutral comparison is null after correction, a six-pair prompt experiment contains a strong effect in one Qwen setting with valid final answers throughout, and a 30,240-run study on new repository tasks shows broad but strongly model- and interface-dependent effects. The largest effects in that study also show why output format matters, since many runs fail before a valid tool action is executed. We further distinguish successful repair from mechanism identification and identify one setting in which directly providing the source needed after a tool failure removes the observed gap. A control or intervention should therefore not be assumed to behave the same way after the model, interface, prompt wording, output format, or inference setup changes. An intervention that improves an agent tells us that the intervention matters, but explaining the original failure requires showing why it matters.

AI USE STATEMENT

In this work, we used generative AI tools to help refine research hypotheses, provide feedback on experimental design and methodology, develop the conceptual framing of the work, and assist in the interpretation and synthesis of experimental results. We did not use generative AI to generate the experimental datasets, translate research material, or perform qualitative or thematic data analysis.

We additionally used generative AI tools during manuscript preparation to edit portions of the manuscript and appendix, improve readability, and assist with LaTeX formatting and the preparation of research files. Generative AI was not used to generate the experimental measurements reported in the paper. AI-assisted descriptions of prior work were checked against the cited papers and publication records, and AI-assisted manuscript text was reviewed and edited by the authors for technical accuracy and appropriate scope. Experimental conclusions were reported only when supported by the underlying recorded results, including negative and null findings. The authors made the final decisions regarding experimental interpretation, scientific claims, citations, and manuscript content.

We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

The work studies failure analysis and causal evaluation of tool-using language model agents. Although the methods could in principle be applied to agents operating in higher-risk settings, our experiments are restricted to controlled software and evidence-retrieval tasks and do not involve deployment on users or consequential real-world actions. We also distinguish positive, negative, null, and inconclusive results. Failed model–interface checks, unsuccessful external-benchmark transfers, and post-hoc trajectory analyses are not presented as confirmatory evidence. We make mechanism claims only where they are supported by the corresponding controlled intervention and avoid generalizing system-specific findings to all tool-using agents.

REPRODUCIBILITY STATEMENT

We provide the implementation used to run CAB through the anonymous code link given in the abstract. Section 3 describes the task construction, paired conditions, control instruction, tool interfaces, model checks, scoring, and statistical analysis used in the main experiments. The appendix provides the additional details needed to reproduce the reported results, including complete model and interface settings, task and repository construction, per-experiment configurations, full result tables, paired outcome counts, repository-level uncertainty estimates, trajectory measurements, and the additional sensitivity and boundary analyses omitted from the main paper for space. The supplementary materials also include the experiment-to-file mapping and the released data files used to generate the tables and figures. Together, the code, appendix, and supplementary files are intended to make the reported experiments and analyses reproducible from the released materials.

REFERENCES

David Arbour, Iavor Bojinov, Avi Feller, and Tu Ni. Toward causal field evaluations of AI systems. *Harvard Data Science Review*, 8(2), 2026. doi: 10.1162/99608f92.7d74e33e.

Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. Why do multi-agent LLM systems fail? In *Advances in Neural Information Processing Systems*, 2025.

Bhaskar Gurram. Auditing automated evaluation, error propagation, and runtime mitigation in tool-using language agents. *arXiv preprint arXiv:2604.16706*, 2026.

Miguel A. Hernán and James M. Robins. *Causal Inference: What If*. Chapman & Hall/CRC, 2020.

Kosuke Imai, Luke Keele, and Dustin Tingley. A general approach to causal mediation analysis. *Psychological Methods*, 15(4):309–334, 2010.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R. Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations*, 2024.

Sayash Kapoor, Benedikt Stroebl, Peter Kirgis, Nitya Nadgir, Zachary S. Siegel, Boyi Wei, Tianci Xue, Ziru Chen, Felix Chen, Saiteja Utpala, Franck Ndzomga, Dheeraj Oruganty, Sophie Luskin, Kangheng Liu, Botao Yu, Amit Arora, Dongyoon Hahm, Harsh Trivedi, Huan Sun, Juyong Lee, Tengjun Jin, Yifan Mai, Yifei Zhou, Yuxuan Zhu, Rishi Bommasani, Daniel Kang, Dawn Song, Peter Henderson, Yu Su, Percy Liang, and Arvind Narayanan. Holistic Agent Leaderboard: The Missing Infrastructure for AI Agent Evaluation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.11977.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. API-bank: A comprehensive benchmark for tool-augmented LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3102–3116, 2023. doi: 10.18653/v1/2023.emnlp-main.187.

Xiaofeng Lin, Yingxu Wang, Tung Sum Thomas Kwok, Daniel Guo, Sahil Arun Nale, Charles Fleming, and Guang Cheng. REFLECT: Intervention-supported error attribution for silent failures in LLM agent traces. *arXiv preprint arXiv:2606.09071*, 2026.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. AgentBench: Evaluating LLMs as Agents. In *International Conference on Learning Representations*, 2024. URL https://mlanthology.org/iclr/2024/liu2024iclr-agentbench/.

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Haoping Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, Zirui Wang, and Ruoming Pang. ToolSandbox: A stateful, conversational, interactive evaluation benchmark for LLM tool use capabilities. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1160–1183, 2025. doi: 10.18653/v1/2025.findings-naacl.65.

Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. AgentBoard: An analytical evaluation board of multi-turn LLM agents. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Ming Ma, Jue Zhang, Fangkai Yang, Yu Kang, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. DoVer: Intervention-Driven Auto Debugging for LLM Multi-Agent Systems. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=mrEK16Jy6h.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: A benchmark for general AI assistants. In *International Conference on Learning Representations*, 2024.

Mayank Mishra, Matt Stallone, Gaoyuan Zhang, Yikang Shen, Aditya Prasad, Adriana Meza Soria, Michele Merler, Parameswaran Selvam, Saptha Surendran, Shivdeep Singh, et al. Granite code models: A family of open foundation models for code intelligence. *arXiv preprint arXiv:2405.04324*, 2024.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large language model connected with massive APIs. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Shishir G Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The berkeley function calling leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 48371–48392, 2025.

Judea Pearl. Direct and indirect effects. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, pp. 411–420, 2001.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. ToolLLM: Facilitating Large Language Models to Master 16000+ Real-World APIs. In *International Conference on Learning Representations*, 2024. URL https://mlanthology.org/iclr/2024/qin2024iclr-toolllm/.

Anka Reuel, Amelia Hardy, Chandler Smith, Max Lamparth, Malcolm Hardy, and Mykel J. Kochenderfer. BetterBench: Assessing AI benchmarks, uncovering issues, and establishing best practices. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Dongyu Ru, Lin Qiu, Xiangkun Hu, Tianhang Zhang, Peng Shi, Shuaichen Chang, Jiayang Cheng, Cunxiang Wang, Shichao Sun, Huanyu Li, Zizhao Zhang, Binjie Wang, Jiarong Jiang, Tong He, Zhiguo Wang, Pengfei Liu, Yue Zhang, and Zheng Zhang. RAGChecker: A Fine-Grained Framework for Diagnosing Retrieval-Augmented Generation. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-0692.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Jaineet Shah. Causal agent replay: Counterfactual attribution for LLM-agent failures. *arXiv preprint arXiv:2606.08275*, 2026.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

Leyao Wang, Yanan He, Peng Chen, Asaf Yehudai, Yixin Liu, Rex Ying, Michal Shmueli-Scheuer, and Arman Cohan. Time to REFLECT: Can we trust LLM judges for evidence-based research agents? *arXiv preprint arXiv:2605.19196*, 2026.

Zihao Wang. Callability is not operability: Controlled interface interventions for LLM agents. *arXiv preprint arXiv:2608.23628*, 2026.

Yufei Xia, Anjun Gao, Yueyang Quan, Zhuqing Liu, and Minghong Fang. Who broke the system? failure localization in LLM-based multi-agent systems. In *Conference on Language Modeling*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. ReAct: Synergizing Reasoning and Acting in Language Models. In *International Conference on Learning Representations*, 2023. URL https://mlanthology.org/iclr/2023/yao2023iclr-react/.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *International Conference on Learning Representations*, 2025.

Ori Yoran, Samuel Joseph Amouyal, Chaitanya Malaviya, Ben Bogin, Ofir Press, and Jonathan Berant. AssistantBench: Can web agents solve realistic and time-consuming tasks? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8938–8968, 2024. doi: 10.18653/v1/2024.emnlp-main.505.

Guibin Zhang, Junhao Wang, Junjie Chen, Wangchunshu Zhou, Kun Wang, and Shuicheng Yan. AgenTracer: Who Is Inducing Failure in the LLM Agentic Systems? In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2509.03312.

Shaokun Zhang, Ming Yin, Jieyu Zhang, Jiale Liu, Zhiguang Han, Jingyang Zhang, Beibin Li, Chi Wang, Huazheng Wang, Yiran Chen, and Qingyun Wu. Which Agent Causes Task Failures and When? On Automated Failure Attribution of LLM Multi-Agent Systems. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 76583–76599. PMLR, 2025. URL https://proceedings.mlr.press/v267/zhang25cq.html.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, 2024.

Kunlun Zhu, Zijia Liu, Bingxuan Li, Muxin Tian, Yingxuan Yang, Jiaxun Zhang, Pengrui Han, Qipeng Xie, Fuyang Cui, Weijia Zhang, Xiaoteng Ma, Xiaodong Yu, Gowtham Ramesh, Jialian Wu, Zicheng Liu, Pan Lu, James Zou, and Jiaxuan You. Where LLM agents fail and how they can learn from failures. *arXiv preprint arXiv:2509.25370*, 2025.

Yuchen Zhuang, Yue Yu, Kuan Wang, Haotian Sun, and Chao Zhang. ToolQA: A dataset for LLM question answering with external tools. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

A Appendix Overview

This appendix provides additional details on task construction, model and tool-interface checks, complete experimental results, robustness analyses, and unsuccessful transfer experiments. We report both positive and null results, together with implementation failures that limit the conclusions of the main paper. The additional material is organized by the scientific question it addresses rather than by the order in which the experiments were run.

B Task Construction

B.1 Authentic repository tasks

The authentic repository evaluation contains 32 cases from eight open-source software ecosystems: Cargo, Docker Compose, curl, Flask, Kustomize, Maven, Node.js, and Prometheus. The cases cover four types of controlled evidence change: conflicting observations, corrupted retrieved state, tool failure, and tool removal. Each case has a fixed question and answer across CLEAN, PRESERVE, and DISRUPT. The PRESERVE condition changes the environment without removing the information needed to answer, while DISRUPT changes how the decisive information can be reached. Primary model outputs are scored deterministically.

B.2 Repository sets for model–interface comparisons

The model–interface experiments use three disjoint sets of 24 repositories with 48 cases in each set. The third set contains equal numbers of tool-failure and decoy cases. The answer is derived from repository evidence and remains fixed across the paired prompt and interface conditions. The no-tool experiment in Appendix D.5 uses the same 48 cases without access to tools to estimate how often the answer can be produced without repository evidence.

B.3 Tasks for the required-source experiment

The source-provision experiment contains 24 tasks evaluated under four conditions. In the first condition, the agent must find the source needed after a tool failure. A second condition provides an unrelated source, a third provides the required source directly, and a fourth blocks the original route while leaving another valid route available. The same 24 tasks are evaluated with four model families and two interfaces. The main paper makes a mechanism claim only for Gemma under PARALLEL, where directly providing the required source removes the observed gap.

Table 6: Authentic repository evaluation.

Condition	Correct	Accuracy
CLEAN	27/32	84.375%
PRESERVE	26/32	81.250%
DISRUPT	14/32	43.750%

B.4 MATCHED PROMPT EXPERIMENTS

One experiment evaluates BASE, NEUTRAL, and PROCEDURAL prompts together for Gemma and Qwen under both interfaces and both task conditions. A second experiment uses six matched neutral/procedural prompt pairs spanning short, medium, and long formulations on the 48-case repository set. Its primary comparison averages procedural minus neutral over all six prompt pairs rather than selecting one wording after observing the results.

The largest matched-prompt evaluation contains 360 tasks from 180 repositories, three model families, both interfaces, both task conditions, and seven prompt variants per system: BASE plus matched neutral/procedural short, medium, and long wordings. The tasks are derived from public repositories but operate over small declared-source environments with precomputed fields. They test controlled evidence retrieval rather than native repository exploration, code modification, testing, or broad software-engineering competence.

C MODELS, TOOL INTERFACES, AND PRELIMINARY CHECKS

Before a model–interface combination is used in a main comparison, we check whether it can reliably execute separate clean tasks with that interface. This prevents interface incompatibility from being mistaken for an experimental effect. The final SERIAL and PARALLEL interfaces use the same structured action format and differ only in whether one or several sources may be requested in one action. In a preliminary check of these interfaces, all 80 trajectories complete, use tools, and return valid structured outputs with zero parser or action rejections.

For the first model–interface comparison, Granite 3.3 8B, Falcon3 10B, and Hermes-3/Llama-3.1 8B pass separate checks under both interfaces before the BASE and PLACEBO outcomes are evaluated. Across the selected combinations, all $96/96$ runs complete, all $96/96$ use tools, all $96/96$ return valid final answers, and $95/96$ are correct. Later model–interface and source-provision experiments use the same general procedure on their own held-out tasks. Complete preliminary-check results are included in the released files.

D ADDITIONAL RESULTS

D.1 AUTHENTIC REPOSITORY DISRUPTION

The CLEAN minus DISRUPT difference is $+40.625$ percentage points with a 95% confidence interval of $[25.0, 56.25]$ obtained by resampling repositories. Paired discordance is 13 clean-only to 0 disrupt-only cases, giving exact $p = 0.00024414$. Seven of eight repository-level differences are positive and one is zero. Repeating the analysis after removing each repository in turn gives effects from $+35.714$ to $+46.429$ points. The CLEAN minus PRESERVE difference is only $+3.125$ points.

D.2 SOURCE INFORMATION AND THE CONTROL INSTRUCTION

Table 1 in the main paper reports the full outcome counts. For Qwen, correct source identifiers exceed incorrect identifiers by 77.083 points under DISRUPT, with 95% CI $[60.417, 91.667]$. For Ministral the corresponding difference is 62.5 points with CI $[45.833, 77.083]$. Incorrect source identifiers also sharply reduce performance under CLEAN. After subtracting the corresponding clean difference, the contrast is 0 points for Qwen with CI $[-8.333, 8.333]$ and -2.083 points for Ministral with CI $[-8.333, 4.167]$. Correct source information is therefore highly useful without identifying a disruption-specific source-selection failure.

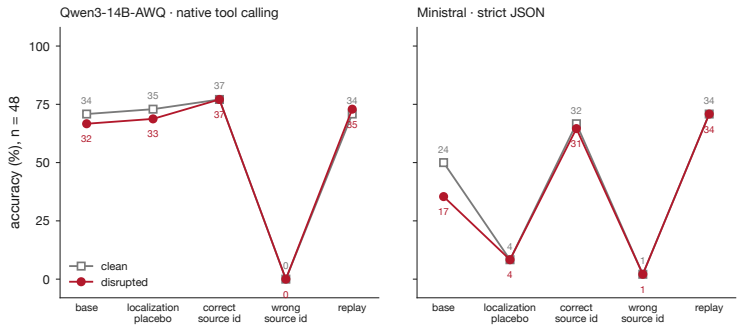

Figure 6: All conditions of the source-information and control experiment. Correct source identifiers provide a large advantage over incorrect identifiers, but the clean-adjusted comparison is near zero. The control is nearly neutral for Qwen and strongly harmful for Ministral under the single-action JSON interface.

Providing the previously retrieved information directly before the final answer also improves performance, but it changes several parts of the original interaction at once. For Ministral, this condition exceeds base by 35.417 points under DISRUPT and by 20.833 points under CLEAN. For Qwen the corresponding differences are $+6.25$ and 0 points. These comparisons place much of the original loss before final answer generation but do not identify one unique earlier step.

The control instruction behaves differently across the two agent setups. Qwen changes by $+2.083$ points in both CLEAN and DISRUPT. Ministral changes by -41.667 points in CLEAN and -27.083 points in DISRUPT. The Ministral trajectories contain 567 parser-error events across 92 of 96 control trajectories. Depending on how consecutive action objects are counted, 415–446 of these events contain multiple actions in one response. The important observation is that the added instruction often makes Ministral emit several individually valid actions while the interface expects one top-level action.

D.3 FULL MODEL–INTERFACE RESULTS

The strongest within-model comparison occurs for Gemma in repository set 3. Under PARALLEL/DISRUPT, discordance is one control-only success to 17 base-only successes, giving exact $p = 0.00014496$. Repeating the analysis after removing each repository in turn gives effects from approximately -34.8 to -30.4 points. In this setting, the Gemma-minus-InternLM difference is -22.917 points with CI $[-39.583, -6.25]$, and the Gemma-minus-Qwen difference is -27.083 points with CI $[-41.667, -12.5]$.

The forty-comparison display includes many clean settings near ceiling and several weak-model settings near floor. A descriptive analysis classifies 21 of 40 comparisons as having useful dynamic range and 19 as being at or near ceiling or floor. All five comparisons that survive the within-set correction lie among the 21 comparisons with useful dynamic range. We report this only as a description of the evaluation and do not alter the denominator of the original tests.

D.4 PROVIDING THE REQUIRED SOURCE

For Gemma under PARALLEL, the counts when the agent must find the required source are $13/24$ under base and $4/24$ under control. The difference is -37.5 points with raw exact $p = .00390625$ and adjusted $p = .03125$. Providing an unrelated source gives $17/24$ versus $5/24$, while providing the required source gives $24/24$ versus $24/24$. The final comparison reaches ceiling, so we use it only to show that the observed gap disappears when the required source is available. Across the full experiment, no run is correct without reading the required source. This follows partly from the task design and is not treated as a separate causal result.

A stored hash for this 24-task set could not be matched to the reconstructed task list, and the correction set for one secondary alternative-route analysis is unclear. These issues do not affect

Table 7: Full control effects for the three model–interface repository sets. Values are PLACEBO minus BASE in percentage points. Confidence intervals are obtained by resampling repositories. Adjusted p-values were computed in the final analysis; because the correction sets were not specified in advance for every early comparison, they are treated as multiple-comparison sensitivity checks.

		Clean		Disrupt			Difference
Model	Interface	base/control	effect [CI]	base/control	effect [CI]	adj. p	disrupt − clean
Repository set 1							
Falcon3	serial	11/5	−12.5 [−25.0,−2.1]	2/1	−2.1 [−6.2,0.0]	1.000	10.4 [0.0,22.9]
Falcon3	parallel	3/1	−4.2 [−12.5,4.2]	0/0	0.0 [0.0,0.0]	1.000	4.2 [−4.2,12.5]
Granite	serial	28/35	14.6 [2.1,27.1]	12/29	35.4 [18.8,52.1]	.0027	20.8 [6.2,35.4]
Granite	parallel	46/43	−6.2 [−14.6,2.1]	44/39	−10.4 [−20.8,0.0]	1.000	−4.2 [−10.4,0.0]
Llama-Hermes	serial	45/43	−4.2 [−10.4,0.0]	28/32	8.3 [−2.1,18.8]	1.000	12.5 [2.1,22.9]
Llama-Hermes	parallel	46/46	0.0 [−8.3,8.3]	21/28	14.6 [6.2,25.0]	.1719	14.6 [2.1,27.1]
Repository set 2							
Gemma3	parallel	48/48	0.0 [0.0,0.0]	26/14	−25.0 [−37.5,−14.6]	.0055	−25.0 [−37.5,−14.6]
Gemma3	serial	48/48	0.0 [0.0,0.0]	31/11	−41.7 [−56.2,−29.2]	<.0001	−41.7 [−56.2,−27.1]
InternLM2.5	parallel	46/44	−4.2 [−10.4,0.0]	19/25	12.5 [2.1,22.9]	.8750	16.7 [6.2,27.1]
InternLM2.5	serial	47/45	−4.2 [−10.4,0.0]	23/29	12.5 [0.0,25.0]	1.000	16.7 [4.2,29.2]
Qwen3	parallel	48/47	−2.1 [−6.2,0.0]	20/22	4.2 [−6.2,16.7]	1.000	6.2 [−4.2,16.7]
Qwen3	serial	48/48	0.0 [0.0,0.0]	18/6	−25.0 [−35.4,−12.5]	.0167	−25.0 [−35.4,−12.5]
Yi-1.5	parallel	48/48	0.0 [0.0,0.0]	18/12	−12.5 [−25.0,0.0]	.8750	−12.5 [−25.0,0.0]
Yi-1.5	serial	48/48	0.0 [0.0,0.0]	3/3	0.0 [−8.3,10.4]	1.000	0.0 [−8.3,10.4]
Repository set 3							
Gemma3	parallel	48/47	−2.1 [−6.2,0.0]	45/29	−33.3 [−43.8,−22.9]	.0017	−31.2 [−41.7,−20.8]
Gemma3	serial	48/48	0.0 [0.0,0.0]	44/43	−2.1 [−8.3,4.2]	1.000	−2.1 [−8.3,4.2]
InternLM2.5	parallel	48/48	0.0 [0.0,0.0]	41/36	−10.4 [−20.8,0.0]	1.000	−10.4 [−22.9,2.1]
InternLM2.5	serial	47/47	0.0 [0.0,0.0]	44/43	−2.1 [−10.4,6.2]	1.000	−2.1 [−10.4,6.2]
Qwen3	parallel	48/48	0.0 [0.0,0.0]	43/40	−6.2 [−14.6,2.1]	1.000	−6.2 [−14.6,2.1]
Qwen3	serial	48/48	0.0 [0.0,0.0]	47/46	−2.1 [−8.3,4.2]	1.000	−2.1 [−8.3,4.2]

Table 8: Control effect under four source-availability conditions. Values are percentage points. CEIL marks conditions where both prompt arms are $24/24$, so their accuracy difference is zero at the top of the scale.

Model	Interface	Find source	Alternative route	Unrelated source	Required source	Find-source adj. p
Gemma3	parallel	−37.5	−20.8	−50.0	CEIL	.0312
Gemma3	serial	−12.5	−20.8	−16.7	CEIL	1.000
InternLM2.5	parallel	0.0	CEIL	20.8	CEIL	1.000
InternLM2.5	serial	0.0	−4.2	16.7	4.2	1.000
Qwen3	parallel	−4.2	−12.5	8.3	CEIL	1.000
Qwen3	serial	16.7	−8.3	−25.0	CEIL	.8750
Yi-1.5	parallel	−20.8	−4.2	−4.2	−4.2	.8750
Yi-1.5	serial	−4.2	8.3	16.7	−16.7	1.000

the Gemma–PARALLEL comparison used in the main paper, and we do not use the secondary alternative-route analysis to support a stronger claim.

D.5 ADDITIONAL CHECKS

Figure 5 in the main paper summarizes three additional checks involving a strict raw-JSON requirement, a separate file-format task family, and removal of tools.

Under the strict raw-JSON format, InternLM base emissions are valid JSON, while the control condition repeatedly adds prose before the JSON object. Across clean and disrupted cases, all 768 InternLM control emissions contain this prose preamble, while all 377 base emissions are valid JSON. Qwen remains valid under both prompt conditions. The result is therefore a systematic model-specific interaction with the required output format.

The no-tool result is $2/48$ for Gemma, $0/48$ for InternLM, $1/48$ for Qwen, and $0/48$ for Yi. All 192 outputs satisfy the required answer format. The tool-failure half is $0/96$ and the decoy half is $3/96$. This weakens a simple memorized-answer explanation of the repository result but does not establish that the repositories were absent from model pretraining.

Table 9: Additional checks. Control effects are PLACEBO minus BASE.

Experiment	System	Condition	n	Effect	95% CI	Adj. p
Raw JSON	InternLM2.5 serial	clean	48	-93.8	[-100.0,-87.5]	$<.0001$
Raw JSON	InternLM2.5 serial	disrupt	48	-79.2	[-89.6,-66.7]	$<.0001$
Raw JSON	Qwen3 serial	clean	48	0.0	[0.0,0.0]	1.000
Raw JSON	Qwen3 serial	disrupt	48	-2.1	[-8.3,4.2]	1.000
File-format tasks	Gemma3 parallel	disrupt	24	-12.5	[-29.2,4.2]	1.000
File-format tasks	Gemma3 serial	disrupt	24	0.0	[0.0,0.0]	1.000
File-format tasks	InternLM2.5 parallel	disrupt	24	0.0	[-12.5,12.5]	1.000
File-format tasks	InternLM2.5 serial	disrupt	24	4.2	[0.0,12.5]	1.000
File-format tasks	Qwen3 parallel	disrupt	24	-4.2	[-12.5,0.0]	1.000
File-format tasks	Qwen3 serial	disrupt	24	-12.5	[-25.0,0.0]	1.000

Table 10: Procedural-minus-neutral effects when the three prompt conditions are evaluated together. None of the eight comparisons specified before evaluation survives Holm correction.

Model	Setting	Proc.	Neutral	Diff. pp
Gemma3	P/Clean	48/48	48/48	0.0
Gemma3	P/Disrupt	32/48	37/48	-10.4
Gemma3	S/Clean	43/48	48/48	-10.4
Gemma3	S/Disrupt	32/48	29/48	$+6.3$
Qwen3	P/Clean	47/48	47/48	0.0
Qwen3	P/Disrupt	34/48	39/48	-10.4
Qwen3	S/Clean	48/48	48/48	0.0
Qwen3	S/Disrupt	36/48	43/48	-14.6

E MATCHED PROMPT EXPERIMENTS

E.1 NEUTRAL AND PROCEDURAL PROMPTS EVALUATED TOGETHER

The first matched-control experiment evaluates BASE, NEUTRAL, and PROCEDURAL together for Gemma3 and Qwen3 under both interfaces and both task conditions, for 1,152 trajectories. The primary comparison is PROCEDURAL minus NEUTRAL.

The largest raw difference is Qwen3 SERIAL/DISRUPT at -14.583 points, with Holm-adjusted $p = 0.125$. Thus, this experiment does not establish that procedural wording is more harmful than matched neutral wording. Comparisons against BASE can still show that adding apparently harmless prompt text changes behavior, but the procedural-minus-neutral comparisons remain null after correction.

E.2 EVALUATION ON 180 REPOSITORIES

The 180-repository evaluation contains 30,240 runs from 360 tasks. Gemma3, InternLM2.5, and Qwen3 are evaluated under both interfaces and both task conditions. Each system has BASE plus matched neutral/procedural short, medium, and long prompt variants. The primary comparison averages procedural minus neutral over the wording lengths specified before evaluation.

The pattern depends strongly on model and interface. Gemma is around -33 to -36 points in all four settings, InternLM ranges from roughly -10.7 to -24.6 points, and Qwen is about -5.7 points under SERIAL but exactly zero under PARALLEL. Neutral-minus-BASE differences are small and do not survive the same correction. This does not establish equivalence between neutral and base prompts, but it shows that the large procedural differences are not reproduced by the matched neutral wording in this experiment.

The largest differences are closely tied to the required output format. Across all four Gemma settings, the longest procedural wording is $0/360$ correct, with all $360/360$ runs in each setting failing before a valid tool action. Across the full experiment, 2,413 execution errors are recorded, including 2,289 JSON-decoding errors and 124 invalid-read-action errors. These runs show that prompt

Table 11: Primary procedural-minus-neutral effects in the 180-repository evaluation. Confidence intervals resample repositories. The ten nonzero comparisons survive the Holm correction specified for this experiment.

Model	Interface	Condition	Effect pp	95% CI	Holm
Gemma3	Serial	Clean	−35.741	[−36.667,−34.907]	.00012
Gemma3	Serial	Disrupt	−35.463	[−36.667,−34.259]	.00012
Gemma3	Parallel	Clean	−33.426	[−33.611,−33.333]	.00012
Gemma3	Parallel	Disrupt	−33.333	[−33.333,−33.333]	.00012
InternLM2.5	Serial	Clean	−10.741	[−12.315,−9.167]	.00012
InternLM2.5	Serial	Disrupt	−12.593	[−14.167,−11.019]	.00012
InternLM2.5	Parallel	Clean	−24.630	[−26.204,−23.056]	.00012
InternLM2.5	Parallel	Disrupt	−23.426	[−24.722,−22.130]	.00012
Qwen3	Serial	Clean	−5.741	[−7.037,−4.537]	.00012
Qwen3	Serial	Disrupt	−5.648	[−6.944,−4.444]	.00012
Qwen3	Parallel	Clean	0.000	[0.000,0.000]	1
Qwen3	Parallel	Disrupt	0.000	[0.000,0.000]	1

Table 12: Procedural-minus-neutral effects averaged over six matched prompt pairs.

Model	Interface	Condition	Effect pp
Gemma3	Serial	Clean	−3.819
Gemma3	Serial	Disrupt	+6.944
Gemma3	Parallel	Clean	0.000
Gemma3	Parallel	Disrupt	−1.736
Qwen3	Serial	Clean	0.000
Qwen3	Serial	Disrupt	−11.458
Qwen3	Parallel	Clean	+1.042
Qwen3	Parallel	Disrupt	−4.514

wording can change whether the model successfully communicates an action through the required format. They do not establish an intrinsic loss of task reasoning.

The scope of these tasks is narrower than broad software engineering. They use public repositories but present small declared-source environments with precomputed fields and require controlled evidence retrieval or transcription. Base accuracy ranges from 98.33% to 100% across the twelve primary systems. We therefore use this experiment as a large controlled repository evaluation rather than evidence about native repository exploration or code modification.

E.3 Six matched prompt pairs

The six-prompt-pair experiment asks whether a procedural-versus-neutral difference persists across several phrasings rather than depending on one sentence. It contains 4,992 runs over six matched prompt pairs specified before evaluation. Every run reaches a valid final answer, there are zero parser rejections, and the 28 runs with a rejected action all recover and later return a valid answer.

Qwen3 Serial/Disrupt is the only primary comparison that clearly survives the correction specified before evaluation. Procedural accuracy is $223/288$ compared with $256/288$ for neutral, giving -11.458 points with 95% CI $[-16.319, -6.944]$ and adjusted $p < .001$. Repeating the analysis after removing each repository in turn gives estimates from -11.957 to -10.507 points. The six prompt-pair effects in this setting are -10.417, $+4.167$, -18.750, -20.833, $+2.083$, and -25.000 points. Thus, the average difference is substantial in this model and interface, while the individual prompt pairs vary considerably.

F Robustness and Implementation Checks

F.1 Statistical calibration check

A separate CPU simulation was intended to check interval coverage and sensitivity under controlled effects. The simulation did not reproduce the repository-bootstrap and multiplicity procedure used in the main experiments and therefore cannot validate that procedure. We retain this as an unsuccessful calibration check and do not use it to support any statistical claim in the paper.

F.2 Repeated runs across server restarts and random seeds

We repeat three selected settings over 12 tasks, four server restarts, and three request seeds per restart, for 864 runs. All requests complete successfully. Two Qwen settings remain exact null or ceiling comparisons in every restart and repetition. The Gemma setting has parser failures in almost every run and gives $0/144$ under neutral and $5/144$ under procedural, with restart-level differences of $+8.33$, $+2.78$, 0, and $+2.78$ points. The corresponding earlier setting had shown a much larger effect. These results show that repeated runs can be stable within one inference setup while the same prompt condition behaves differently after the inference setup changes.

F.3 Held-out repository evaluation

A later evaluation on 24 public repositories does not provide a clean interface comparison because the parallel-action validator incorrectly requires exactly one source identifier even though the prompt allows several. All 107 recorded read-action validation errors occur in parallel conditions. The serial results can be reported descriptively, but the parallel effects and interface interactions are not used. A corrected follow-up on a new repository set does not yield usable model outcomes because of server and parsing failures, so it provides no transfer estimate.

F.4 Alternative output formats

We also attempted to compare raw JSON with constrained structured output and native tool calls. The raw-JSON configuration reaches the normal evaluation code, but the other two configurations fail before valid model outputs are produced. This experiment therefore does not establish whether the control effect persists, disappears, or reverses under working alternative formats.

F.5 Reproducibility and scoring checks

We rerun the 180-repository analysis from a separate copy of the released files and reproduce the reported numerical outputs exactly. This checks reproducibility of the analysis rather than independent model generation. We also test 42 parsing and scoring cases together with 11 deliberately altered inputs. These tests behave as expected, subject to the documented limits of the parser checks. Completed model runs, failed runs, preliminary checks, and simulations are reported separately in the released files, and fields that were not recorded are left unavailable rather than estimated.

G Additional Analyses and Null Results

G.1 Tool-use behavior can change without accuracy

An early routing experiment varies the pressure to use tools. With a hard budget of three, $0/120$ runs reach the full declared route. Under a tool-cost condition, $112/120$ do so, and under unrestricted routing the count is $119/120$. Correct-answer counts remain similar across the three conditions. Thus, large changes in route completion or tool-call volume need not produce corresponding changes in task accuracy.

G.2 Access to tools does not guarantee correct evidence use

A separate 32-case evaluation gives $16/32$ correct under clean, $15/32$ under preserve, and $9/32$ under disrupt. The clean-minus-disrupt difference is $+21.875$ points with 8 clean-only versus 1

disrupt-only pair and exact $p = .03906$. Route coverage remains $32/32$, $32/32$, and $31/32$, and all declared tools are called in $30/32$, $30/32$, and $29/32$ cases. In the conflict and corruption subset, only $2/16$ disrupted cases are correct even though $15/16$ call all tools. Tool access and invocation are therefore not sufficient for correct use of the available information.

G.3 Previously retrieved information and source hints

On the authentic 32-case evaluation, providing the same information already retrieved during the run directly before the final answer gives $27/32$ under disrupt compared with the end-to-end $14/32$. Adding a structured comparison step on the same information gives $24/32$, so the extra comparison step does not selectively rescue the disrupted condition. These results place much of the loss before final answer generation without identifying one unique earlier cause.

A separate source-identifier intervention reaches $32/32$ on both clean and disrupted versions of a 32-case evaluation. It reveals source identifiers but not their contents or the answer. This shows that source information is sufficient to make the tasks solvable. Giving unrelated information in the same format does not produce the disruption-specific pattern predicted by a simple source-selection explanation. The more complete source-information experiment is reported in Appendix D.2.

G.4 Deterministic scoring checks

A deterministic scoring check compares literal equality with accepted lexical aliases. For one system the counts are $27/64$ under literal matching and $32/64$ with accepted aliases. For Qwen they are $49/64$ and $56/64$. These checks illustrate why answer formatting and task correctness should be separated. They do not replace the primary scorers used in the later experiments.

G.5 Additional model checks

Attempts to broaden the earliest authentic-repository comparison to additional models were limited by low clean-task accuracy, interface mismatch, or server incompatibility. We therefore do not average those systems into the reported effect or replace them after observing the results. The later model–interface experiments provide the broader model comparison used in the main paper.

G.6 Output wording and decoding

In a 192-trajectory sensitivity analysis, changing the visible output instructions from a stricter to a more flexible form changes parsing, termination, tool use, generated content, and final answers. The accuracy difference is about -10.417 points in both clean and disrupted conditions, with confidence intervals crossing zero. We treat this as supporting evidence that output instructions can affect execution rather than as a main accuracy result.

The large greedy-decoding Ministral control effect also becomes much smaller under sampling. On the same 24-case subset with temperature 0.7, top-p 0.95, and three replicates per setting, the control effects are -2.778 points in clean and -5.556 points in disrupt for Ministral, and -4.167 and $+1.389$ points for Qwen. Parser errors and maximum-step terminations also become less frequent. Thus, the effect observed under greedy decoding should not be assumed to remain unchanged under a different decoding setup.

H Trajectory Analysis

The analyses in this section use trajectories already produced by the main experiments. Unless a quantity is changed directly by the experiment, we use it to describe how the runs differ rather than to identify a mechanism.

H.1 Multi-source reading and first-action differences

For Gemma under PARALLEL, the fraction of read actions containing several sources increases from 16.0% under base to 85.7% under control in the disrupted condition. Under clean it increases from

20.5% to 100%, even though the clean accuracy difference is only -2.1 points. Large behavioral changes can therefore occur without a large change in aggregate accuracy.

The paired trajectories often differ before any tool result is observed. Gemma under PARALLEL differs on the first action in $36/48$ clean pairs and $36/48$ disrupted pairs. InternLM under PARALLEL differs on the first action in $31/48$ pairs in both conditions. Under SERIAL, first-action differences are much less common. These observations show that the control can alter the initial sequence of actions rather than acting only after evidence has been observed.

H.2 READING THE REQUIRED SOURCE

In the 24 tool-failure cases from repository set 3, whether the required source is read closely tracks correctness in the cells examined. Gemma under PARALLEL reads the source in $23/24$ cases and is correct in $23/24$ under base, compared with $7/24$ and $7/24$ under control. Gemma under SERIAL is $24/24$ versus $21/24$ for source access, while InternLM under PARALLEL is $18/24$ versus $14/24$ and Qwen under PARALLEL is $20/24$ versus $17/24$. Because these measurements are observed after the prompt changes, they are descriptive. The experiment that directly provides the required source in Appendix D.4 is the basis for the mechanism claim.

H.3 THE POSITIVE GRANITE RESULT

The positive Granite result shows a different local pattern. Under SERIAL/DISRUPT, base is $12/48$ correct with $28/48$ valid final answers and $20/48$ maximum-step terminations. The control is $29/48$ correct with $41/48$ valid final answers and $7/48$ maximum-step terminations. Total source reads decrease from 138 to 123. In this setting, the added instruction is associated with fewer long unsuccessful searches, which provides one plausible explanation for the positive accuracy difference.

H.4 TOOL-FAILURE AND DECOY CASES

Across the later repository evaluations, tool-failure cases are often more affected by the control than decoy cases, but the pattern is not universal. Across 20 model–interface comparisons, the tool-failure effect is more negative than the decoy effect in 13 cases, less negative in three, and tied in four. The mean difference is -20.208 points and an exploratory exact sign test on the non-tied comparisons gives approximately $p = .021$. Because this comparison was assembled after the main experiments, we treat it as exploratory.

H.5 A NON-SIGNIFICANT ACCURACY DIFFERENCE DOES NOT ESTABLISH IDENTICAL BEHAVIOR

For 48 primary comparisons with non-significant raw accuracy differences, a later sensitivity analysis asks how small the paired disagreement rate can be shown to be. None of the 48 comparisons has a one-sided 95% upper bound at or below 5%. The median upper bound is 16.7163%, and 33 of 48 remain compatible with more than 10% paired disagreement. These bounds treat case-pair indicators as independent and therefore do not replace the repository-level bootstrap used for the main results. Their purpose is only to show that the available data do not establish that a statistically null control behaves identically to base.

I STATISTICAL ANALYSIS

Paired outcomes. When the same task appears in two conditions, correctness is compared within task. Exact two-sided paired tests are used for binary outcomes. Invalid outputs and maximum-step terminations remain in the denominator rather than being removed after the prompt condition is applied.

Repository clustering. The authentic repository evaluation contains eight repository groups, while the model–interface evaluations contain 24. Confidence intervals for the main effects resample repositories rather than individual trajectories. Interface differences are formed within case

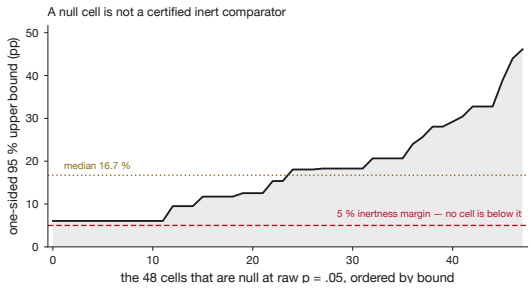

Figure 7: Sensitivity analysis for comparisons with non-significant mean accuracy differences. The analysis does not establish that the control is active. It shows that the available sample sizes generally cannot establish near-perfect agreement between paired outcomes.

before repository resampling so that the shared repository structure of the compared conditions is preserved. We also repeat selected analyses after removing each repository in turn.

Multiple comparisons. The Holm-adjusted values shown for the later model–interface and additional-check experiments reproduce numerically. The exact correction sets were not specified in advance for every early experiment, so we describe those values as corrections computed in the final analysis rather than as pre-specified families. As a sensitivity analysis, all forty primary model–interface comparisons were also adjusted together. The strongest negative Gemma result in repository set 2, the Gemma PARALLEL result in repository set 3, and the positive Granite result in repository set 1 remain below 0.01 under this pooled correction. The main conclusion that the sign and magnitude vary across systems does not depend on treating the earlier correction groups as pre-specified.

Bootstrap presentation. With 24 repositories, percentile bootstrap endpoints move in discrete increments and can change slightly with the resampling seed. We therefore report intervals to one decimal place in the paper and retain exact seeds and full-precision intervals in the released data. No main conclusion depends on the last reported decimal.

Trajectory measurements. Route coverage, multi-source reading, whether the required source was read, the first action, and task-family splits can all change after the prompt changes. We therefore use these measurements to describe differences between runs rather than to claim that they caused an accuracy difference. The required-source experiment is different because source availability itself is changed directly.

J PUBLIC BENCHMARK EXPERIMENTS

We attempted to apply the paired-control design to several established public evaluations. None produced a valid confirmatory control-effect estimate, and we do not treat these attempts as null replications. The limitations arose from incomplete execution, model checks, response handling, or the inability to construct the required paired comparison under the official evaluation setup.

In one later attempt using AgentDojo and SurfaceBench, all 106 planned model requests execute, but the model emits intended tool calls in a representation that the evaluation code does not consume. The resulting zero scores are therefore inconclusive. A separate attempt with the official τ^2-bench environment also completes the planned requests, but the inference server is not configured with the native tool-call path required by the benchmark. Those zero rewards reflect the execution setup rather than a valid benchmark comparison.

Table 13: Public benchmark attempts. None produced a valid paired control-effect estimate.

Benchmark	Evaluation completed	Status	Why no effect is reported
SWE-bench	4/80 planned runs	incomplete	Execution and quota constraints prevented a valid paired comparison.
BFCL	preliminary evaluation	model check not passed	The available model families did not meet the required official function-calling competence level.
AgentBench	15 observed runs	model check not passed	The selected checkpoint did not meet the required task-success and recovery checks.
API-Bank	30 observed runs	model check not passed	Executable-call competence was below the threshold specified for the comparison.
AgentDojo / SurfaceBench	106/106 requests	response handling failed	Model actions were emitted in a representation that the evaluation code did not consume.
τ^2-bench	24/24 requests	inference configuration failed	The server lacked the native tool-call execution path required by the official benchmark setup.

K Additional Limitations

Several results limit the scope of the conclusions. The matched procedural-versus-neutral comparison is null after correction in one experiment. A separate file-format task set is largely null. Some candidate models do not pass the preliminary interface checks. The large 180-repository evaluation contains strong effects, but many of its largest differences arise because the model fails to produce the required action format. Repeated runs also show that a prompt condition can behave differently after the inference setup changes.

Other attempted extensions remain inconclusive. A held-out interface evaluation contains a validator error in the parallel condition, and the corrected follow-up does not produce usable model outcomes. The statistical calibration simulation does not reproduce the procedure used in the main experiments. Alternative output-format and public-benchmark comparisons also fail before a valid paired estimate can be obtained. We report these as limitations rather than as null effects.

Taken together, the evidence supports a deliberately limited conclusion. A control with no task-specific information can change agent behavior, and the size and direction of that change depend on the model, interface, prompt wording, output format, and inference setup. The evidence does not show that procedural wording is universally harmful, that neutral wording is equivalent to BASE, that one interface is uniformly better, that every control effect is caused by parsing, or that the effect has been confirmed on a public benchmark. The mechanism result is also local: directly providing the required source removes the observed gap in one Gemma–PARALLEL setting, not across all models and interfaces.