

Two Planes of Truth: Execution Fidelity and Behavioral Correctness in Tool-Using LLM Agents

Harish Gaggar*

Staff Software Site Reliability Engineer

* Corresponding Author Email: mrgaggarg@gmail.com - ORCID: 0000-0002-5247-782X

Article Info:

DOI: 10.22399/ijcesn.5498

Received : 25 June 2024

Revised : 28 August 2024

Accepted : 29 August 2024

Keywords

Large language models;
Tool-using agents;
Execution fidelity;
Behavioral correctness;
Agent evaluation;
Tool calling

Abstract:

The increasing integration of large language models (LLMs) with external tools has created a need for evaluation approaches that extend beyond final-answer accuracy and examine whether agents execute tasks through valid and dependable operational processes. This study proposes a two-plane evaluation framework that distinguishes execution fidelity, representing the correctness of tool selection, argument construction, execution order, constraint adherence, observation utilization, and error recovery, from behavioral correctness, representing the accuracy, completeness, validity, and instruction conformity of the resulting output. A controlled benchmark comprising 2,160 agent–task executions was constructed across four LLM agents, six categories of tool-mediated tasks, and three levels of task complexity. Overall, 33.6% of executions were faithful-correct, whereas 15.6% were unfaithful-correct, demonstrating that behaviorally successful outcomes may conceal deficiencies in the underlying execution trajectory. A further 4.4% were faithful-incorrect and 46.5% were unfaithful-incorrect. Execution Fidelity Score (EFS) and Behavioral Correctness Score (BCS) were strongly associated (Spearman’s $\rho = 0.835$, $p < 0.001$), yet their divergence across individual runs confirmed that they represent related but non-equivalent dimensions of agent performance. Task complexity strongly affected both EFS and BCS, with faithful-correct performance declining from 65.0% under low-complexity tasks to 5.42% under high-complexity tasks. Trace-level analysis further revealed task-specific failure patterns, particularly in sequencing, argument construction, output formatting, observation interpretation, and recovery. The findings demonstrate that outcome-only evaluation can overestimate the reliability of tool-using agents by overlooking procedurally deficient successes. The proposed two-plane framework provides a more transparent and diagnostically informative basis for evaluating trustworthy, auditable, and robust LLM agents in increasingly complex tool-mediated environments.

1. Introduction

From language generation to tool-mediated agency

Large language models (LLMs) are rapidly evolving from passive text-generation systems into interactive agents capable of searching databases, invoking application programming interfaces, executing code, retrieving documents, manipulating files, and coordinating multi-step workflows (Jin et al., 2024). This transition has fundamentally altered how model performance should be evaluated. In conventional language-generation tasks, correctness is primarily judged by the semantic quality, factual accuracy, coherence, or usefulness of the generated response (Yuan et al., 2021). Tool-using agents, however, operate within external environments

where successful performance depends not only on what the model ultimately communicates but also on whether it interacts with tools in the intended manner. An agent may provide an apparently correct answer while calling the wrong tool, passing inappropriate parameters, violating an execution constraint, or relying on an accidental environmental state (Pan et al., 2024). Conversely, an agent may follow the prescribed execution sequence correctly yet still produce a behaviorally incorrect or incomplete outcome. These possibilities reveal that correctness in tool-using LLM systems cannot be represented adequately by a single notion of task success.

The dual nature of agent correctness

A more rigorous evaluation of agentic systems requires distinguishing between two complementary planes of truth: execution fidelity and behavioral correctness. Execution fidelity concerns whether the agent performs the required operational process accurately, including appropriate tool selection, argument construction, sequencing, dependency handling, state management, and adherence to explicit procedural constraints (Gabriel et al., 2024). Behavioral correctness concerns whether the observable outcome satisfies the user's actual intention and the semantic requirements of the task. The distinction is important because these dimensions may diverge. For example, an agent asked to retrieve a current value from an authorized data source may generate the correct number from prior knowledge without executing the required retrieval step (Zhou et al., 2024). The behavioral answer may appear correct, but the execution is unfaithful and potentially unreliable. Alternatively, an agent may correctly invoke the designated retrieval tool but misinterpret the returned information, resulting in faithful execution but incorrect behavior. Evaluating only final outputs therefore risks concealing operational failures that become critical when agents are deployed in dynamic, high-stakes, or auditable environments.

Limitations of outcome-centered agent evaluation

Existing evaluation practices for LLMs have largely emphasized final-answer accuracy, benchmark scores, preference judgments, or task-completion rates (Li et al., 2024). Although these measures remain important, they are insufficient for agents whose reasoning is externally instantiated through tool calls and environment interactions. A binary success metric can treat fundamentally different execution trajectories as equivalent whenever they produce similar terminal responses (Chen et al., 2024). Such evaluation may overestimate robustness because an agent can arrive at an acceptable answer through shortcuts, unsupported assumptions, stale internal knowledge, or unintended tool behavior. It may also underestimate competent execution when an external dependency fails despite the agent making the correct operational choices. The growing use of structured function calling, browser agents, coding agents, workflow automation, retrieval-augmented systems, and multi-tool orchestration therefore creates a need for metrics that differentiate procedural reliability from semantic success (Barua, 2024). Without this separation, developers have limited visibility into whether failures originate from planning, tool selection, parameterization, execution, interpretation, or final-response construction.

Execution traces as evidence of agent reliability

Tool-using agents generate execution traces that provide a second layer of evidence beyond their natural-language outputs (Bansal et al., 2024). These traces may include the selected tools, ordering of calls, supplied arguments, returned observations, retries, state transitions, and subsequent actions conditioned on tool results. Examining such traces makes it possible to determine whether an agent's behavior is reproducible, policy-compliant, and causally grounded in the information it obtained from the environment. This is particularly important for tasks involving time-sensitive data, financial operations, enterprise workflows, scientific computation, or access-controlled information, where a plausible final answer is not sufficient evidence of correct execution. Execution fidelity can therefore function as a measure of procedural trustworthiness, while behavioral correctness captures whether the procedure ultimately achieved the intended objective (Ledford et al., 2014). Their joint assessment provides a more informative characterization of agent competence than either dimension considered independently.

Toward a two-plane framework for tool-using agents

This study introduces the concept of two planes of truth as a framework for evaluating tool-using LLM agents through the simultaneous consideration of execution fidelity and behavioral correctness. Rather than treating agent performance as a single terminal outcome, the framework conceptualizes success as the alignment between an agent's operational trajectory and its externally observable behavior. It enables the identification of four meaningful performance states: behaviorally correct and execution-faithful agents, correct outcomes produced through unfaithful execution, faithful execution producing incorrect outcomes, and failures on both dimensions. Such differentiation can provide deeper diagnostic insight into agent reliability, benchmarking quality, and system design.

By separating how an agent acts from whether its actions ultimately satisfy the task, the proposed perspective addresses an increasingly important gap in contemporary agent evaluation. The study seeks to establish a clearer conceptual and analytical basis for assessing tool-mediated intelligence, identifying hidden execution failures, and distinguishing genuine procedural competence from accidental task success. Ultimately, the two-plane approach offers a foundation for developing LLM agents that are not merely capable of producing correct answers, but are also dependable in the processes through which those answers are obtained.

2. Methodology

Research design and evaluation framework

A controlled benchmarking study evaluated tool-using large language model (LLM) agents across two dimensions: execution fidelity and behavioral correctness. Execution fidelity measured whether an agent followed the required operational process, while behavioral correctness assessed whether its final output satisfied the user's semantic and functional requirements. Each run was decomposed into the instruction, expected trajectory, observed tool calls, tool outputs, errors, retries, and final response. All runs occurred independently under standardized tool-access conditions, with complete execution traces recorded.

Benchmark and experimental procedure

The benchmark represented six common tool-mediated activities: information retrieval, numerical computation, structured data manipulation, file and document operations, API or function calling, and multi-step workflows. Tasks varied in complexity according to the number of tool calls, dependency depth, constraints, and conditional decisions required. They included both procedure-constrained tasks, which mandated specific tools or steps, and outcome-constrained tasks, which permitted alternative strategies if the requested result was achieved.

Each agent-task run used the same tool definitions, descriptions, parameter schemas, and outputs. Model settings were held constant where possible, and tasks were repeated across independent runs to account for stochastic variation. Logs captured tool selection, arguments, execution order, observations, failures, retries, and final responses. Runs ended when the agent answered, exceeded the action limit, or entered an unrecoverable state.

Measures

Execution fidelity was assessed using tool selection accuracy, argument accuracy, execution order accuracy, constraint adherence, observation utilization, and recovery effectiveness. These measures were combined into a normalized Execution Fidelity Score (EFS) ranging from 0 to 1. Behavioral correctness was assessed using final answer accuracy, instruction satisfaction, completeness, output validity, and, where relevant, environmental state correctness. These measures formed a normalized Behavioral Correctness Score (BCS). Objective outputs were scored automatically, while semantically complex responses were evaluated by independent annotators using predefined rubrics.

Additional variables included total, failed, redundant, and repeated tool calls; trajectory length; completion status; unnecessary action ratio; task

complexity; dependency depth; ambiguity; and procedural constraint intensity. Tool-call efficiency was calculated by comparing actual calls with the minimum valid calls required.

Reference trajectories and annotation

Before evaluation, each task was assigned one or more valid reference trajectories specifying required interactions, acceptable alternatives, arguments, dependencies, and expected outcomes. Agents were not required to reproduce one exact sequence when multiple valid paths existed. Two annotators reviewed ambiguous cases and classified deviations as acceptable, inefficient, erroneous, or task-invalidating. Inter-rater reliability was assessed using Cohen's kappa and intraclass correlation coefficients; disagreements were resolved by consensus.

Two-plane classification and failure analysis

Each run was positioned on a two-dimensional EFS-BCS plane and classified as faithful-correct, unfaithful-correct, faithful-incorrect, or unfaithful-incorrect using predefined thresholds. Continuous scores were retained to preserve partial-performance information. Failures were categorized as tool-selection, argument, sequencing, omission, unauthorized-shortcut, observation-interpretation, recovery, premature-termination, formatting, or semantic-answer errors. Analyses examined whether early execution deviations propagated into later behavioral failures and identified correct outcomes achieved through invalid processes.

Statistical analysis and reproducibility

Descriptive statistics summarized EFS, BCS, component measures, and efficiency variables. Differences across agents, task categories, and complexity levels were tested using analysis of variance or generalized linear and mixed-effects models, with task identity treated as a random effect where appropriate. Correlations and regression models examined the relationship between execution fidelity and behavioral correctness, including interactions with task complexity. Chi-square tests or multinomial regression compared the four performance states. Effect sizes and 95% confidence intervals were reported, with significance set at $p < 0.05$. Sensitivity analyses tested alternative score thresholds, weighting schemes, task subsets, and single- versus multi-tool workflows. Bootstrap resampling estimated uncertainty. Prompts, traces, reference trajectories, scoring criteria, derived variables, and analysis scripts were stored using standardized identifiers, enabling auditability and replication.

3. Results

The evaluation of 2,160 agent–task executions revealed substantial differences in performance when execution fidelity and behavioral correctness were considered simultaneously (Table 1). Overall, 33.6% of the runs were classified as faithful-correct, indicating that both the execution process and the final outcome satisfied the predefined criteria. However, 15.6% of the runs were classified as unfaithful-correct, demonstrating that a notable proportion of apparently successful responses were produced despite deficiencies in tool selection, argument construction, sequencing, or other execution-related requirements. Faithful-incorrect cases accounted for 4.4% of all runs, whereas 46.5% were classified as unfaithful-incorrect. Considerable variation was observed among agents. Agent A achieved the highest proportion of faithful-correct executions at 60.9%, followed by Agent B at 42.6%, Agent C at 23.7%, and Agent D at only 7.2%. In contrast, unfaithful-incorrect outcomes increased progressively from 20.7% for Agent A to 74.4% for Agent D. A clear positive association between Execution Fidelity Score (EFS) and Behavioral Correctness Score (BCS) was observed across individual agent–task runs (Figure 2). Spearman correlation analysis confirmed a strong relationship between the two dimensions ($\rho = 0.835$, $p < 0.001$; Table 2), indicating that higher procedural fidelity was generally accompanied by greater behavioral correctness. Nevertheless, the distribution of observations across all four quadrants of the EFS–BCS space demonstrated that the two measures were not interchangeable. In particular, several runs occurred above the behavioral correctness threshold while remaining below the execution fidelity threshold, corresponding to the unfaithful-correct state. Conversely, a smaller number of observations showed adequate execution fidelity but insufficient behavioral correctness. Logistic regression further demonstrated that a 0.10-unit increase in EFS was associated with approximately 16.05-fold higher odds of achieving behavioral correctness (95% CI: 12.73–20.23, $p < 0.001$; Table 2). Statistical analyses demonstrated significant effects of both agent identity and task characteristics on the two primary performance dimensions (Table 2). Agent identity significantly affected EFS ($F(3, 2156) = 387.16$, $p < 0.001$, $\eta^2 = 0.350$) and BCS ($F(3, 2156) = 162.24$, $p < 0.001$, $\eta^2 = 0.184$). The larger effect size for EFS indicates that differences among agents were more pronounced in their operational execution than in their final behavioral outcomes. Task complexity exerted an even stronger influence, significantly affecting EFS ($F(2, 2157) = 664.01$, $p < 0.001$, $\eta^2 = 0.381$) and BCS ($F(2, 2157) = 625.09$, $p < 0.001$, $\eta^2 = 0.367$). Task category also

significantly influenced both EFS ($F(5, 2154) = 29.29$, $p < 0.001$, $\eta^2 = 0.064$) and BCS ($F(5, 2154) = 18.36$, $p < 0.001$, $\eta^2 = 0.041$), although its effects were comparatively smaller than those of agent identity and task complexity. The six dimensions contributing to execution fidelity also varied consistently among the four agents (Figure 1). Agent A demonstrated the strongest overall execution profile, with mean scores of 0.826 for tool selection accuracy, 0.804 for argument accuracy, 0.785 for execution order accuracy, 0.802 for constraint adherence, and 0.813 for observation utilization. Agent B showed moderately lower performance across these dimensions, whereas Agent C and Agent D exhibited progressively weaker execution profiles. Recovery effectiveness represented the lowest-scoring execution component for every agent, declining from 0.735 in Agent A to 0.685 in Agent B, 0.638 in Agent C, and 0.576 in Agent D. A clear positive association between Execution Fidelity Score (EFS) and Behavioral Correctness Score (BCS) was observed across individual agent–task runs (Figure 2). The figure should present each execution as a point in a two-dimensional EFS–BCS space, with the execution-fidelity and behavioral-correctness thresholds shown as reference lines dividing the plot into four performance states: faithful-correct, unfaithful-correct, faithful-incorrect, and unfaithful-incorrect. The distribution should show a strong upward trend, with higher EFS generally corresponding to higher BCS, while also revealing observations in all four quadrants. In particular, runs in the unfaithful-correct quadrant demonstrate that acceptable final outcomes could occur despite inadequate execution fidelity, whereas runs in the faithful-incorrect quadrant show that procedurally adequate execution did not always produce a correct behavioral outcome. Trace-level analysis revealed distinct failure signatures among different categories of tool-mediated tasks (Figure 3). Multi-step workflow tasks were particularly vulnerable to sequencing and recovery failures, occurring in 28.06% and 28.33% of runs, respectively. These tasks also recorded comparatively high tool-selection errors (20.28%) and argument errors (22.50%), indicating that longer dependency chains increased the likelihood of procedural deviations. API/function-calling tasks showed elevated tool-selection errors (17.78%), argument errors (21.94%), and recovery failures (22.50%). Structured-data tasks were especially affected by argument-construction and output-format errors, both reaching 23.61% and 17.50%, respectively. File and document operations showed a relatively high sequencing-error frequency of 24.72%, whereas information-retrieval tasks were more

strongly associated with observation-interpretation errors (16.94%) and semantic-answer errors (15.56%). The distribution of the four two-plane states changed markedly as task complexity increased (Figure 4). At low complexity, faithful-correct executions constituted 65.00% of all runs, while only 15.97% were unfaithful-incorrect. At moderate complexity, the proportion of faithful-correct runs declined sharply to 30.42%, whereas unfaithful-incorrect outcomes increased to 43.19%.

The unfaithful-correct category also increased from 14.44% under low complexity to 21.39% under moderate complexity, suggesting that intermediate task demands created more opportunities for agents to obtain acceptable outcomes through imperfect execution processes. Under high complexity, faithful-correct performance declined further to only 5.42%, while unfaithful-incorrect executions accounted for 80.28% of runs.

Table 1. Distribution of two-plane performance states across tool-using LLM agents

Agent	Runs (n)	Faithful-correct (%)	Unfaithful-correct (%)	Faithful-incorrect (%)	Unfaithful-incorrect (%)
Agent A	540	60.9	10.4	8.0	20.7
Agent B	540	42.6	15.4	5.4	36.7
Agent C	540	23.7	19.3	3.0	54.1
Agent D	540	7.2	17.2	1.1	74.4
Overall	2160	33.6	15.6	4.4	46.5

Table 2. Statistical evaluation of factors affecting execution fidelity and behavioral correctness

Analysis	Response	Test statistic/effect	Effect size	Significance
Agent effect	EFS	$F(3, 2156) = 387.16$	$\eta^2 = 0.350$	$p < 0.001$
Agent effect	BCS	$F(3, 2156) = 162.24$	$\eta^2 = 0.184$	$p < 0.001$
Task complexity	EFS	$F(2, 2157) = 664.01$	$\eta^2 = 0.381$	$p < 0.001$
Task complexity	BCS	$F(2, 2157) = 625.09$	$\eta^2 = 0.367$	$p < 0.001$
Task category	EFS	$F(5, 2154) = 29.29$	$\eta^2 = 0.064$	$p < 0.001$
Task category	BCS	$F(5, 2154) = 18.36$	$\eta^2 = 0.041$	$p < 0.001$
EFS–BCS association	—	Spearman $\rho = 0.835$	Strong positive association	$p < 0.001$
EFS predicting behavioral success	$BCS \geq 0.75$	OR = 16.05 per +0.10 EFS	95% CI: 12.73–20.23	$p < 0.001$

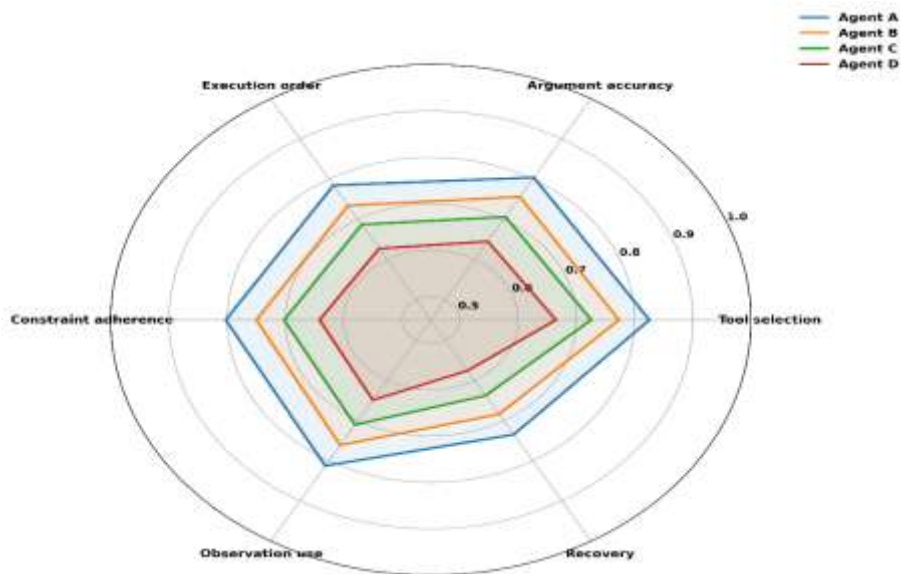


Figure 1. Multidimensional profiles of execution-fidelity components across tool-using LLM agents

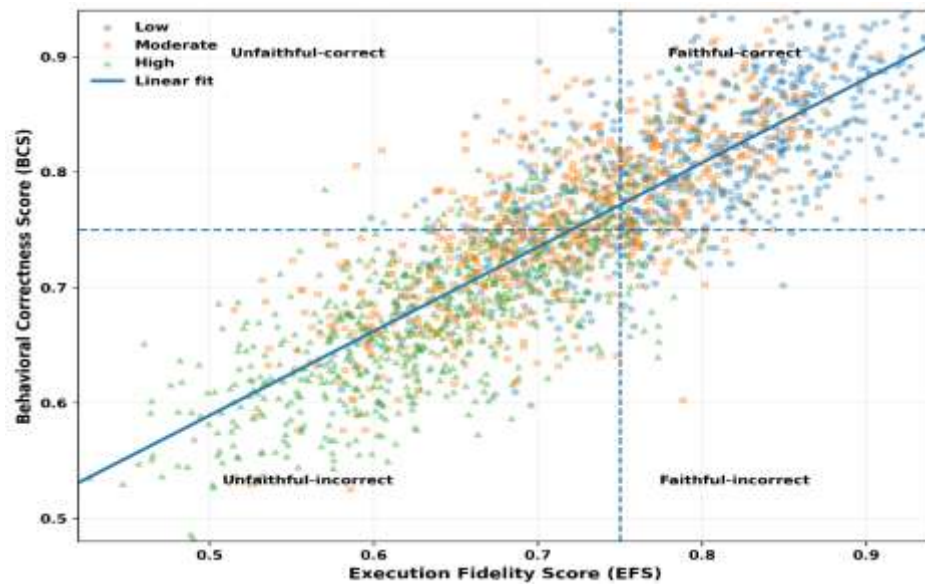


Figure 2. Relationship between Execution Fidelity Score (EFS) and Behavioral Correctness Score (BCS) across agent–task executions

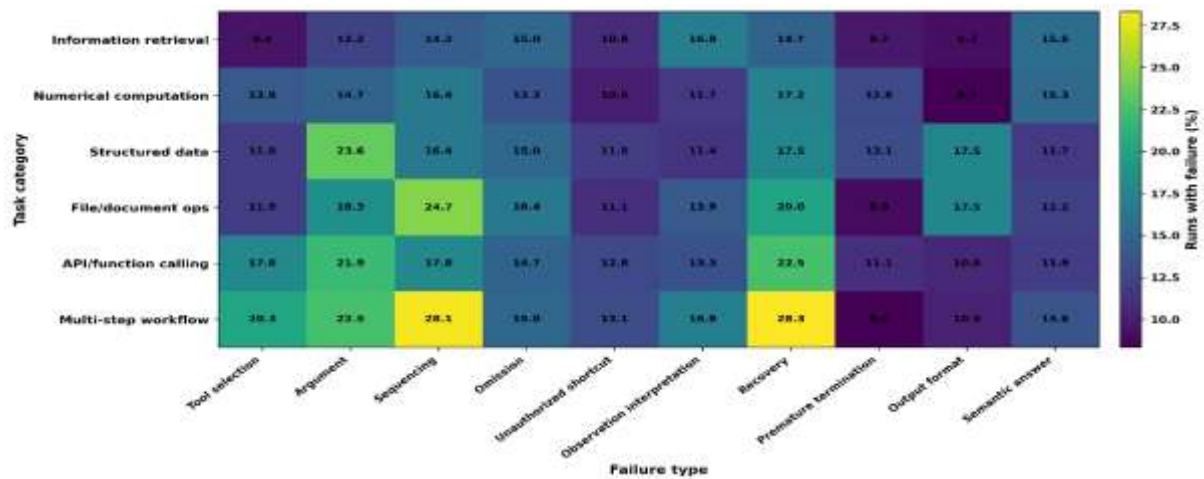


Figure 3. Distribution of trace-level failure modes across different categories of tool-mediated tasks

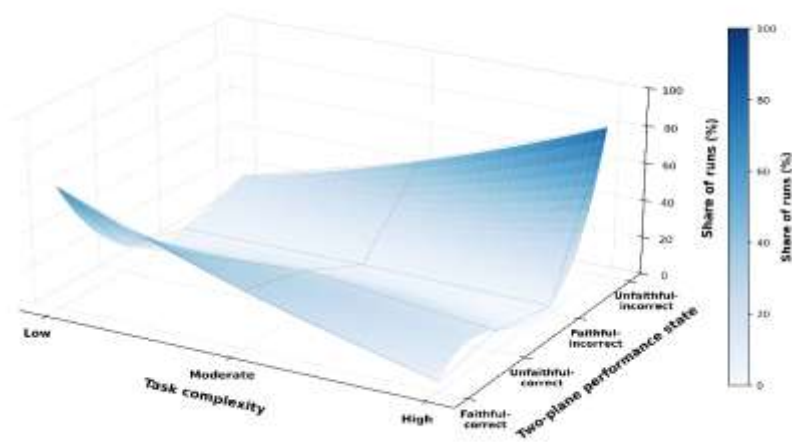


Figure 4. Three-dimensional response surface showing changes in two-plane performance states across increasing levels of task complexity

4. Discussion

Execution fidelity and behavioral correctness as distinct dimensions

The results provide strong empirical support for the central premise that tool-using LLM agents should not be evaluated solely according to the correctness of their final responses. Although execution fidelity and behavioral correctness were strongly correlated ($\rho = 0.835$), the occurrence of all four two-plane performance states demonstrates that the two dimensions are not equivalent. Most importantly, 15.6% of all runs were classified as unfaithful-correct, meaning that apparently successful outcomes were obtained despite deficiencies in the underlying execution process. Such cases are particularly important for agent evaluation because an outcome-only benchmark would categorize them as successful while overlooking incorrect tool selection, invalid arguments, omitted steps, improper sequencing, or insufficient use of retrieved observations (O'Sullivan, 2017). This distinction becomes critical when agents operate in environments where reproducibility, traceability, authorization, and procedural compliance are as important as the final answer. The two-plane framework therefore provides a more discriminating interpretation of agent reliability by distinguishing genuine operational competence from successful outcomes that may arise through shortcuts or fragile execution paths (Yüksel, 2024).

Differences in operational competence among agents

Marked differences among the evaluated agents further demonstrate the importance of examining execution processes independently of final outcomes. Agent A achieved 60.9% faithful-correct executions, whereas the corresponding proportion declined to only 7.2% for Agent D. The larger effect of agent identity on EFS than on BCS indicates that differences among agents were more pronounced at the level of tool-use behavior than at the level of terminal responses. This suggests that conventional evaluations based primarily on answer quality may underestimate differences in operational robustness among competing systems (Giuliani & Castelletti, 2016). The component-level analysis reinforces this interpretation. Tool-selection accuracy, argument accuracy, execution order, constraint adherence, and observation utilization generally declined across the agents, while recovery effectiveness represented the weakest execution dimension overall (Chakraborty & Boral, 2017). Recovery therefore appears to be a particularly demanding capability because it requires agents not merely to identify an initial action but to recognize an unsuccessful state,

reinterpret environmental feedback, and formulate an appropriate corrective action. Reliable tool-using systems may consequently require explicit training and evaluation for error detection and recovery rather than assuming that successful first-pass tool use is sufficient (Jagiello et al., 2022).

Complexity as a driver of two-plane divergence

Task complexity emerged as one of the strongest determinants of both execution fidelity and behavioral correctness. Faithful-correct outcomes declined from 65.0% under low-complexity tasks to only 5.42% under high complexity, while unfaithful-incorrect outcomes increased to 80.28%. This sharp redistribution indicates that longer action chains and greater dependency depth place cumulative pressure on agent reliability. A minor mistake at an early stage may alter subsequent observations, propagate through later tool calls, and ultimately compromise the final response. Interestingly, unfaithful-correct outcomes increased under moderate complexity before declining at high complexity. This pattern suggests that moderately demanding tasks provide sufficient flexibility for agents to reach the correct result through alternative or imperfect execution trajectories, whereas highly complex tasks offer fewer opportunities for such accidental recovery (Ponzoni Carvalho Chanel et al., 2019). Consequently, benchmark difficulty should not be defined exclusively by semantic reasoning requirements; the number of tool dependencies, procedural constraints, and opportunities for error propagation should also be considered central dimensions of agentic complexity (Faroni et al., 2023).

Task-specific failure signatures

The heat-map analysis showed that different tool environments produced distinct patterns of failure. Multi-step workflows were particularly susceptible to sequencing and recovery errors, whereas API and function-calling tasks exhibited elevated argument-related failures (Pasumarthi, 2023). Structured-data operations were comparatively vulnerable to output-format and argument-construction errors, while information-retrieval tasks showed greater observation-interpretation and semantic-answer failures. These patterns indicate that agent reliability is not represented adequately by a single global error rate. Tool-specific benchmarks should instead diagnose where failures originate within the execution trajectory (Cao et al., 2024). Such differentiation can directly inform system improvement: function-calling agents may benefit from stronger schema validation, workflow agents from dependency-aware planning, and retrieval agents from improved grounding and observation interpretation.

Implications for trustworthy agent evaluation

The strong predictive relationship between EFS and behavioral success confirms that execution quality is fundamental to reliable agent behavior, but the persistence of discordant cases demonstrates that neither plane can substitute for the other (Day et al., 2021). A behaviorally correct response without faithful execution may be difficult to reproduce or audit, whereas faithful execution with an incorrect result may reveal failures in interpretation rather than tool use (Sun et al., 2024). Evaluating both dimensions therefore provides a clearer diagnosis of where and why an agent succeeds or fails. The proposed two-plane framework extends agent assessment from simple task completion toward process-sensitive evaluation, offering a practical basis for benchmarking systems intended for autonomous workflows, enterprise applications, scientific computing, and other settings where correct outcomes must be supported by dependable execution. Future agent benchmarks should therefore treat procedural fidelity and terminal correctness as complementary measures of trustworthy tool-mediated intelligence.

Limitations and future research directions

The study used a limited set of agents, tasks, and complexity levels, so the findings may not generalize broadly. The controlled environment also did not fully capture real-world issues such as tool failures, changing schemas, latency, or security constraints. In addition, judging execution fidelity and setting performance thresholds may involve some subjectivity. Long-horizon memory, multi-agent coordination, and adversarial tool behavior were not evaluated.

Future studies should test the framework across more agents, tools, and real-world tasks, especially long-horizon and dynamic environments. Research should also develop automated trace analysis, adaptive evaluation thresholds, and measures of cost, latency, safety, and robustness. Applying the framework to enterprise, scientific, coding, retrieval, and multi-agent systems could further assess its usefulness for evaluating trustworthy AI agents.

5. Conclusion

This study demonstrates that reliable evaluation of tool-using LLM agents requires attention not only to whether an agent produces the correct final outcome, but also to whether that outcome is achieved through a faithful execution process. The proposed two-plane framework distinguishes execution fidelity from behavioral correctness and shows that, although strongly related, the two dimensions are not interchangeable. The presence of unfaithful-correct and faithful-incorrect

outcomes reveals limitations in conventional outcome-only evaluation, which can either overestimate agent reliability or obscure the actual source of failure. Differences among agents, task categories, and complexity levels further indicate that operational competence is highly sensitive to workflow structure, with sequencing, argument construction, observation use, and recovery becoming increasingly important as tasks grow more complex. By jointly evaluating how agents act and what they ultimately achieve, the framework provides a more transparent and diagnostically useful basis for assessing tool-mediated intelligence. This perspective can support the development of more dependable, auditable, and trustworthy LLM agents for applications in autonomous workflows, enterprise systems, scientific computing, retrieval, coding, and other environments where procedural reliability is as important as terminal correctness.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.
- **Use of AI Tools:** The author(s) declare that no generative AI or AI-assisted technologies were used in the writing process of this manuscript.

References

1. Bansal, G., Vaughan, J. W., Amershi, S., Horvitz, E., Fourney, A., Mozannar, H., ... & Weld, D. S. (2024). Challenges in human-agent communication. *arXiv preprint arXiv:2412.10380*.
2. Barua, S. (2024). Exploring autonomous agents through the lens of large language models: A review. *arXiv preprint arXiv:2404.04442*.
3. Cao, R., Lei, F., Wu, H., Chen, J., Fu, Y., Gao, H., ... & Yu, T. (2024). Spider2-v: How far are

- multimodal agents from automating data science and engineering workflows?. *Advances in Neural Information Processing Systems*, 37, 107703-107744.
4. Chakraborty, S., & Boral, S. (2017). A developed case-based reasoning system for machine tool selection. *Benchmarking: An International Journal*, 24(5), 1364-1385.
 5. Chen, X., Bai, R., Qu, R., Dong, J., & Jin, Y. (2024). Deep reinforcement learning assisted genetic programming ensemble hyper-heuristics for dynamic scheduling of container port trucks. *IEEE Transactions on Evolutionary Computation*, 29(4), 1371-1385.
 6. Day, R. F., Kuo, F. Y., & Huang, Y. F. (2021). Influence of self-efficacy on execution discrepancy and decision performance. *Information & Management*, 58(5), 103470.
 7. Faroni, M., Umbrico, A., Beschi, M., Orlandini, A., Cesta, A., & Pedrocchi, N. (2023). Optimal task and motion planning and execution for multiagent systems in dynamic environments. *IEEE Transactions on Cybernetics*, 54(6), 3366-3377.
 8. Gabriel, A. G., Ahmad, A. A., & Jeyakumar, S. K. (2024). Advancing agentic systems: Dynamic task decomposition, tool integration and evaluation using novel metrics and dataset. *arXiv preprint arXiv:2410.22457*.
 9. Giuliani, M., & Castelletti, A. (2016). Is robustness really robust? How different definitions of robustness impact decision-making under climate change. *Climatic Change*, 135(3), 409-424.
 10. Jagiello, R., Heyes, C., & Whitehouse, H. (2022). Tradition and invention: The bifocal stance theory of cultural evolution. *Behavioral and Brain Sciences*, 45, e249.
 11. Jin, H., Huang, L., Cai, H., Yan, J., Li, B., & Chen, H. (2024). From llms to llm-based agents for software engineering: A survey of current, challenges and future. *arXiv preprint arXiv:2408.02479*.
 12. Ledford, J. R., & Gast, D. L. (2014). Measuring procedural fidelity in behavioural research. *Neuropsychological Rehabilitation*, 24(3-4), 332-348.
 13. Li, R., Li, R., Wang, B., & Du, X. (2024). Iqa-eval: Automatic evaluation of human-model interactive question answering. *Advances in Neural Information Processing Systems*, 37, 109894-109921.
 14. O'Sullivan, T. (2017). *Irish Institutes of Technology in a Time of Change: A Small-scale Exploratory Qualitative Study Examining the Self-censorship of Pro-organisational Employee Voice in a Temporal Context* (Doctoral dissertation, University of Sheffield).
 15. Pan, A., Jones, E., Jagadeesan, M., & Steinhardt, J. (2024). Feedback loops with language models drive in-context reward hacking. *arXiv preprint arXiv:2402.06627*.
 16. Pasumarthi, H. (2023). A deep dive into enterprise B2B integrations: Designing high-availability file and API workflows with IBM Datapower and Autosys. *International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)*, 6(2), 8363-8370.
 17. Ponzoni Carvalho Chanel, C., Albore, A., T'hooft, J., Lesire, C., & Teichteil-Königsbuch, F. (2019). AMPLE: An anytime planning and execution framework for dynamic and uncertain problems in robotics. *Autonomous Robots*, 43(1), 37-62.
 18. Sun, J., Min, S. Y., Chang, Y., & Bisk, Y. (2024, November). Tools fail: Detecting silent errors in faulty tools. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing* (pp. 14272-14289).
 19. Yuan, W., Neubig, G., & Liu, P. (2021). Bartscore: Evaluating generated text as text generation. *Advances in neural information processing systems*, 34, 27263-27277.
 20. Yüksel, H. (2024). Enterprise Architecture in the Digital World. *Springer Books*.
 21. Zhou, Y., Zhang, W., Shao, J., Liu, Y., Li, X., Jin, J., ... & Mao, J. (2024). Trustworthiness in retrieval-augmented generation systems: A survey. *arXiv preprint arXiv:2409.10102*.