When Does Task Decomposition in Root Cause Analysis LLM-Workflows Cause Performance Regressions?

Ivo Rohwer[1], Martin Straesser[2], Yannik Lubas[1], Samuel Kounev[1], André Bauer[3]
[1] University of Würzburg [2] TH Aschaffenburg [3] Illinois Institute of Technology

Abstract

LLM-workflows are increasingly used for root cause analysis (RCA) of incidents in microservice deployments. Task decomposition is crucial when designing such workflows: Complex tasks are split across multiple LLM calls to improve the workflow performance. Although task decomposition is indispensable for many complex RCAs, a fixed decomposition can limit the generalization capabilities of an LLM workflow. Therefore, decomposition steps that improve performance under some conditions may cause performance degradation under different conditions. We propose a definition for *ε-regression-freeness* to describe the relationship of a decomposition relative to the non-decomposed baseline. We investigate when such regressions can occur using four workflow variations and 2 input modes across the 51 telecom scenarios of the OpenRCA benchmark, and discuss to what extent the observed robustness differences between these variants could be due to structural reasons.

1 Introduction

Root cause analysis (RCA) of incidents in microservice deployments is both business-critical and inherently complex. Consequently, a growing body of research focuses on supporting root cause analyses through LLM-workflows. Similar to classical software workflows, LLM-workflows are also constructed following standardized design patterns [2]. One of the most important design decisions is task decomposition: Many tasks are too complex to be solved by a single LLM call, which is why they are divided into smaller subtasks, each handled through its own LLM call.

While task decomposition is indispensable for handling complex tasks using LLMs, long multi-step LLM pipelines carry the risk that numerous inter-step dependencies reduce generalization capabilities and make the pipeline sensitive to changes [5, 7, 8]. This can result in a decomposition that demonstrated advantages on a test set but leads to performance regression under changed circumstances (e.g., altered input preprocessing).

In this paper, we define *ε-regression-freeness* in Definition 1 as a criterion for assessing the robustness of a decomposition step relative to a baseline without that step. We apply this criterion to analyze the results of various variations of a minimal workflow. For performance evaluation, we test these workflow variations using two input preprocessing modes on the telecom scenarios of the OpenRCA [4] benchmark.

2 Related Work

A number of publications examine LLM-based RCA approaches [4]. Furthermore, numerous works explore domain-agnostic approaches for systematic task decomposition and LLM workflow optimization: The approaches range from the implementation of classical Hierarchical Task Networks for strict specification of action sequences [3], through graph-theoretical minimization of local complexity via constraint decomposition [6], to adaptive prompt optimization [5] and empirical, error-driven restructuring of entire workflow architectures [8]. In some cases, the transferability of optimized systems to deviating data distributions is also cited as an optimization goal. However, this is not formally defined in these works and always refers implicitly to the overall performance of a workflow.

Note that a similar regression-freeness concept has already been applied in the deep learning domain to describe the relationships between individual model generations [1], whereby we transfer this concept to workflow engineering. To the best of our knowledge, no prior work defines regression-freeness for individual LLM-workflow decomposition steps or investigates the regression-freeness of individual decomposition categories.

3 Approach

To construct examples that allow us to discuss the extent to which task decompositions can be *ε-regression-free*, we start with a baseline LLM-workflow that uses only a single LLM call to generate a root cause analysis. This workflow is executed on the OpenRCA [4] telecom benchmark, where each scenario consists of a set of telemetry data and requires three output values: the correct incident start time, the incident cause from 5 predefined categories, and the root cause entity (the concrete pod, service, or database instance where the error occurs). Our baseline workflow receives as input a summary of metric and trace anomalies within the

Definition 1 (ε-Regression-Freeness)

We define an LLM-workflow decomposition d to be *ε-regression-free* with respect to a baseline workflow implementation b over a set of input conditions C if $\text{accuracy}(d, c) \geq \text{accuracy}(b, c) - \varepsilon$ for all $c \in C$. ε-regression-freeness constrains the relative performance of the decomposed workflow compared to the baseline, but makes no statement about absolute performance: both variants may vary with input conditions, and d may outperform b; what is excluded is that d falls substantially behind the baseline when applied under the same conditions.

incident time window, then executes a single LLM call that is tasked with directly generating the complete result (time, cause, entity).

Based on this baseline, we construct a decomposition that adds an intermediate reasoning step, referred to in the following as "Decomposition A": Before the result is generated, an extra LLM call specifically examines whether the given incident is a network-related incident. Then in a second call the actual output is generated. The input is the original telemetry data summary concatenated with the result of the network-specific reasoning. This represents a decomposition from which one could plausibly expect improved performance on one hand, since distinguishing between network-related and non-network-related errors should represent an easier task than directly generating the final result. However, this simultaneously represents an (albeit implicit) additional assumption not contained in the baseline variant, since the statement "Distinguishing network-related incidents is easier and therefore improves the final result" is ultimately an assumption, and whether this is true or false depends substantially on the context of the task.

As a contrasting example, we construct a decomposition that does not introduce any new intermediate representation, which we expect to be less prone to regressions: Instead, it splits the generation of the three output elements across three different LLM calls. The three output elements are generated in the same order and for each call with the same context of already generated outputs as in the monolithic variant. This is referred to in the following as "Decomposition B". The expected improvement also rests on an uncertain assumption, namely that generating each of the required outputs individually is simpler as it allows for a more focused reasoning process of the LLM and is therefore more accurate than generating all three output elements at once. However, unlike in the previous example, this decomposition does not generate any additional, potentially incorrect intermediate result, which is why a violation of this assumption plausibly does not lead to worse performance compared to the baseline.

As a fourth variation, we combine both decompositions, i.e., first the arguments regarding whether it is a network-related error are generated, and then the three output values are generated through 3 individual LLM calls. This can be seen as Decomposition B relative to Decomposition A as baseline (instead of the monolithic baseline), or Decomposition A relative to Decomposition B.

We test these four RCA-workflow variations with two input modes: In the first mode, all metric anomalies from the telemetry data are listed in the input, but anomalies are excluded from the listing if they were also abnormal outside the incident time period. In the second mode, all metric anomalies are listed, including those where the metric showed anomalies outside the incident time period as well. This represents a trivial variation in the algorithmic preprocessing of the input data, as plausibly performed in LLM-workflow engineering, whereby neither of the two variants is "obviously better" since, depending on context, repeatedly occurring anomalous metrics can indeed have important informational content for an incident investigation.

4 Evaluation

To compare the workflow variations under the different input modes, we evaluate the results for the OpenRCA telecom scenarios with the partial accuracy, which indicates the percentage of scenarios where at least one of the three required output elements (time, reason, entity) matches the ground truth. The telecom dataset comprises 51 scenarios, so we conducted 408 workflow runs in total. We performed our experiments using gemini-2.5-flash as LLM model.

Table 1 shows the results of the workflow variations for the first input mode. As we can see, decompositions A and B individually do not represent an improvement over the monolithic baseline. However, the combined variant shows a certain improvement. None of the decompositions appears to represent a deterioration relative to their respective baseline, assuming an $\varepsilon > 2\%$ as defined 1. Table 2 shows the results of the workflow variations for the second input mode. Decomposition B again does not deviate significantly from the baseline. In contrast, Decomposition A shows a notable performance regression (33.3% vs. 47.1%). We suspect this is because the concentration on network errors, given many recurrent anomalous metrics in the input, leads the investigations down the wrong paths. This finding is also confirmed in the results of the combination: Decomposition B+A brings

Table 1: First input mode (recurrent anomalies excluded), accuracy for root cause entity, reason, time and partial accuracy (≥ 1 element correct)

Variant	N	Ent.%	Reas.%	Time%	Part.%
Monolithic	51	23.5	29.4	13.7	41.2
Dec. A	51	15.7	31.4	11.8	39.2
Dec. B	51	19.6	25.5	15.7	39.2
A+B	51	23.5	33.3	19.6	49.0

Table 2: Second input mode (all anomalies including recurrent), accuracy for root cause entity, reason, time and partial accuracy (≥ 1 element correct)

Variant	N	Ent.%	Reas.%	Time%	Part.%
Monolithic	51	13.7	39.2	11.8	47.1
Dec. A	51	13.7	19.6	19.6	33.3
Dec. B	51	19.6	31.4	21.6	49.0
A+B	51	13.7	27.5	13.7	37.3

a slight improvement relative to A (37.3% vs 33.3%). However, B+A performs significantly worse relative to B (37.3% vs. 49.0%). This demonstrates that for this input mode, adding Decomposition A causes a regression compared to both baselines. In contrast, in the first input mode, A+B was an improvement over B as baseline (49.0% vs. 39.2%).

5 Discussion

As described in Section 3, Decomposition A was constructed to generate an additional intermediate representation. This representation may be incorrect under certain conditions, which is most likely the reason for the performance regression observed in the experiment. This raises the question of whether this is a general property of chain-of-thought-like decompositions. We argue that such a fixed division of reasoning steps, in many cases, encodes assumptions that can lead to performance regressions under changed input conditions.

In Section 3, we describe that Decomposition B is constructed such that no additional output is generated compared to the baseline, and we therefore expect this decomposition to be more robust. The results of our experiments confirm this insofar as no performance regression is observed. However, since we only tested two input modes, this result should be regarded as preliminary. Moreover, additional intermediate representations are not the only possible cause of errors, e.g., Zhou et al. [6] discuss how task decompositions that do not account for inter-task dependencies lead to failures. If such unaccounted dependencies lead to worse results than the non-decomposed baseline for certain input conditions, this means, according to our definition, that the decomposition is not regression-free. However, we argue that in our concrete case, the decomposition's regression-freeness is at least plausible, since the outputs are generated with the same context and in the same order as in the baseline.

Overall, the results illustrate how difficult it is to evaluate to what extent a performance gain achieved through task decomposition is actually stable.

6 Conclusion

We introduced ε-regression-freeness as a criterion for describing the robustness of individual task decomposition steps in LLM workflows relative to a non-decomposed baseline and applied it to an RCA workflow for microservice incidents. Our experiments on the telecom scenarios of the OpenRCA benchmark with four different workflow variations show that these exhibit different levels of robustness across the input preprocessing modes we tested: For a decomposition that added an additional reasoning step, we observed a strong performance regression for one of the input modes, whereas this was not the case for a decomposition that only distributed the generation of existing representations across more LLM calls. Detailed experiments that could demonstrate to what extent these results generalize to other sets of input properties, RCA benchmarks, or LLM workflows from other domains remain for future work.

References

[1] S. Yan et al. "Positive-congruent training: Towards regression-free model updates". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.

[2] O. M. Ayala. "Task decomposition and RAG as design patterns for LLM-based systems". In: *International Conference on AI Engineering–Software Engineering for AI (CAIN)*. 2025.

[3] V. Hsiao, M. Roberts, and L. Smith. "Procedural knowledge improves agentic llm workflows". In: *arXiv preprint arXiv:2511.07568* (2025).

[4] J. Xu et al. "OpenRCA: Can large language models locate the root cause of software failures?" In: *International Conference on Learning Representations*. 2025.

[5] M. Zhao et al. "ADOPT: Adaptive Dependency-Guided Joint Prompt Optimization for Multi-Step LLM Pipelines". In: *arXiv preprint arXiv:2512.24933* (2025).

[6] T. Zhou et al. "An approach for systematic decomposition of complex LLM tasks". In: *arXiv preprint arXiv:2510.07772* (2025).

[7] E. Meyerson et al. "Solving a million-step LLM task with zero errors, 2025". In: *Empirical validation of the Babbage Principle via'Massively Decomposed Agentic Processes* ().

[8] J. Zhang et al. "Failure-driven workflow refinement". In: *arXiv preprint arXiv:2510.10035* ().