International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(9s) 2026 pp. 624-633
https://doi.org/10.70917/ijcisim-2026-3460
Article

ContextShapley: Measuring How Context Components 'Helps or Hurts' Language Model Performance

Shaik Ahamad[1], V B Narsimha[2]

[1,2] Department of Computer Science Engineering, UCE(A), Osmania University, Hyderabad, India
ahamadkv17@gmail.com[1], vbnarasimha@gmail.com[2]

Abstract: Modern LLM systems combine multiple context sources—system instructions, few-shot examples, retrieved documents, conversation history, and tool outputs—but practitioners lack a principled way to measure which components help or hurt and how they interact. We present ContextShapley, a reproducible framework that formulates context construction as a cooperative game and uses exact Shapley values to quantify the marginal contribution of these five component types. For each task instance we evaluate all 32 component subsets and compute both individual Shapley values and pairwise interaction indices, and we verify that the resulting attributions satisfy the efficiency, symmetry, and null-player axioms. We evaluate 35 instances sampled from IFEval, HotpotQA, and GSM8K and report cross-model results on both a closed LLM (GPT-5) and an open LLM (Qwen3:8B via Ollama), totaling 2,240 model calls that are fully deterministic and reproducible from a single command. Results show strong task dependence in component values: system instructions dominate instruction following ($\varphi \approx 0.54$), whereas tool outputs and retrieved documents contribute most on knowledge QA; in contrast, retrieved documents are consistently harmful on non-trivial GSM8K cases. We also observe systematic interference between components, including a strong negative document–tool interaction on HotpotQA ($I \approx -0.22$) and frequent degradation from conversation history in knowledge QA. These findings show that more context is not always better and support task-specific, model-aware context optimization rather than fixed prompt templates.
Keywords: - Context Engineering, Language Model Evaluation, Context Attribution, Shapley Value, Cooperative Game Theory

1. INTRODUCTION

Large Language Model (LLM) applications increasingly rely on context engineering, where system instructions, examples, retrieved passages, conversation history, and tool outputs are combined to guide model behaviour . However, context design in most real systems is still heuristic – practitioners add or remove components through trial and error, with limited quantitative evidence about which components are truly useful[1,2].

Existing studies provide important but partial insights. Prompt compression improves efficiency , and long-context studies show that model quality can degrade as context grows or as relevant information is poorly positioned . Attribution methods such as ContextCite explain which text spans influence outputs . Yet, these approaches do not directly answer a key system question: what is the marginal contribution of each context component type, and how do components interact when used together?[3,4]

To address this gap, we introduce ContextShapley, a framework that frames context composition as a cooperative game and computes exact Shapley values for five canonical components, System Instructions (S), Few-Shot Examples (E), Retrieved Documents (D), Conversation History (H), and Tool Outputs (T). By comprehensively evaluating all component coalitions for each task instance, the framework quantifies both individual contributions and pairwise interactions (synergy or interference). This connects context engineering with established game-theoretic attribution in machine learning[5,6,7,8] .

Across a fixed sample of Instruction Following, Multi-Hop Question Answering, and Mathematical Reasoning, and across both closed and open LLM settings. Our results show that context value is strongly task-dependent, i.e.

components that help in one task can be neutral or harmful in another. We also uncovered harmful interaction patterns, demonstrating that simply appending more context does not necessarily improve results. Our work shifts context optimization from intuition-based tweaking to measurement-driven engineering, while also making clear the practical constraints, such as the combinatorial explosion in cost as the number of components increases[9,10].

Objectives

This study develops a framework to analyze how context components influence LLM performance across tasks and models. The research objectives are:

1. Develop a principled framework to quantify the marginal contribution of five context components {S, E, D, H, T} using exact Shapley values.
2. Understand how components influence each other by modeling interaction effects, quantifying both pairwise synergy and interference.
3. Generate task-specific evidence across diverse benchmarks to analyse whether each component's importance is consistent or task-dependent.
4. Assess cross-model consistency and diversity by studying component contribution patterns in both closed source and open source LLMs, and determine which results generalize across models versus which are specific to particular architectures.
5. Provide a fully reproducible research pipeline that includes comprehensive coalition assessments, detailed run logging, and axiom validation, enabling transparent replication and facilitating subsequent use by other researchers.

Taken together, these aims constitute the paper's main contributions: a game-theoretic approach to context attribution, an empirical analysis of benefits and harms at the component level, a cross-task and cross-model study of interaction effects, and a reproducible toolkit to support measurement-driven context optimization[11,12,13].

2. LITERATURE REVIEW

Context construction has become a central design variable in modern LLM systems, especially in agentic and retrieval-augmented workflows. Rather than relying on a single prompt, current pipelines combine system instructions, examples, retrieved evidence, history, and tool output. Recent work has framed this practice as context engineering and highlighted its practical impact on model quality and reliability . However, most applied workflows still rely on heuristic prompt tuning rather than principled component-level measurement[14].

A first stream of literature studies context management at the system level. Wang et al. synthesized a broad context engineering landscape, while Chen et al. discussed agentic context behavior and failure modes in long-horizon use. Li et al. introduced budget-aware context allocation strategies, showing that context tokens are a constrained resource. This stream motivates structured optimization, but it generally treats context as a global budget and does not isolate the marginal value of specific context component types[15].

A second stream focuses on degradation in long contexts. Liu et al. showed positional sensitivity in long inputs, and Chroma Research reported broad context-rot effects across frontier models. Lee et al. further demonstrated that context length can hurt performance even when retrieval quality is controlled. Together, these studies establish that adding more context is not always beneficial, but they do not identify which component classes are responsible for gains or losses in multi component pipelines[16].

A third stream examines retrieval noise and interference. Kim et al. showed that irrelevant passages can distract models in RAG settings, while Xu et al. reported non monotonic trends where more retrieved passages can eventually reduce quality. These findings are important for document selection and mainly analyze interference within retrieval. Cross-component interference (for e.g., documents with tool outputs, or history with examples) remains less quantified[17].

A fourth stream addresses attribution and explainability. Gao et al. proposed context-level attribution to identify influential evidence spans in generation. In parallel, explainable AI methods based on cooperative game theory have provided strong theoretical grounding for contribution analysis . Shapley's original formulation remains the key axiomatic basis for fair marginal contribution estimation[18].

Yet a methodological gap persists prior research rarely provides a clear, component-type level breakdown of contextual value and interaction effects spanning multiple benchmarks and model families. Instead, existing findings

typically take the form of token- or sentence-level attributions, or coarse, prompt-level aggregates, which constrains actionable engineering choices about which component types to include, exclude, or combine[19].

In conclusion, existing work shows that context quality, length, and retrieval relevance are all important, yet it falls short of explaining how different types of context elements contribute to and interfere with one another in end-to-end LLM pipelines. This project tackles that limitation by using exact Shapley-value analysis over context component types, explicitly modeling their interactions across tasks and models, to shift context design from heuristic practice to measurable, reproducible optimization.

3.MATERIALS AND METHODS

We model context composition as a cooperative game (N, v), where $N = \{S, E, D, H, T\}$ is the set of five context component types: System Instructions (S), Few-Shot Examples (E), Retrieved Documents (D), Conversation History (H), and Tool Outputs (T). The value function $v: 2^N \rightarrow [0,1]$ assigns each subset of components a task performance score. For each task instance, we evaluate all $2^5 = 32$ subsets with the LLM to obtain the full value function.

The Shapley value $\phi(c)$ for component c measures the average marginal contribution of c across all possible orderings of component addition:

$$\phi(c) = \sum_{S \subseteq N \setminus \{c\}} \frac{|S|!\ (|N| - |S| - 1)!}{|N|!} [v(S \cup \{c\}) - v(S)]$$

where N is the full set of context components, $c \in N$ is the target component, $S \subseteq N \setminus \{c\}$ is any coalition excluding c, and $v(S)$ is the task performance value of coalition S.

We programmatically verify the efficiency axiom on every task instance: $\sum_c \phi\,(c) = v(N) - v(\emptyset)$. All 70 instances across both models pass, confirming computational correctness.

The pairwise interaction $I(c_i, c_j)$ quantifies synergy ($I > 0$) or interference ($I < 0$) between two components. It is a difference-in-differences measure: how the marginal contribution of c_j changes when c_i is present versus absent. We compute exact interaction indices for all 10 component pairs per task instance.

We fix the assembly order (S → E → D → H → T → Query) to avoid ordering effects. System instructions use the system role; all other components are concatenated into the user message with section headers.

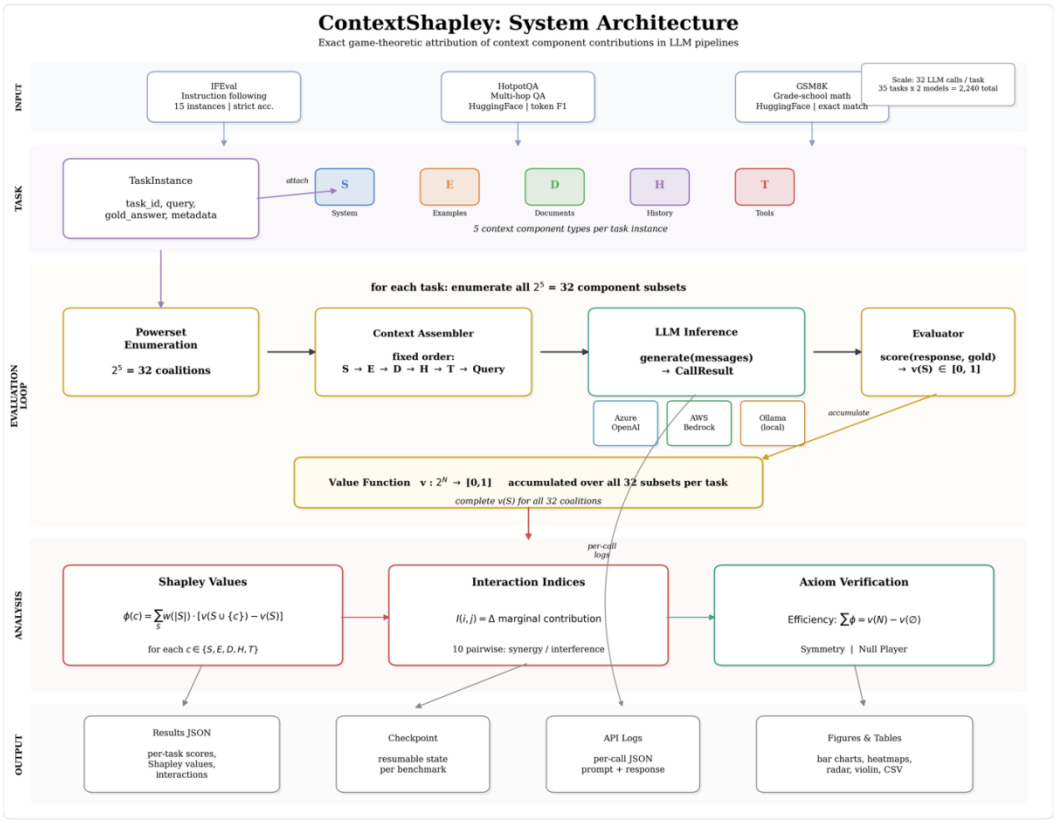

Fig. 1. ContextShapley system architecture: a pipeline from benchmark loading to powerset enumeration, context assembly, multi-provider LLM inference, evaluation, and Shapley computation with axiom verification.

Fig. 1 presents the end-to-end ContextShapley pipeline. For each task instance, all 32 component subsets are evaluated through the inner loop, producing a complete value function that feeds exact Shapley computation with axiom verification.

4. DATA COLLECTION

Data were obtained from three benchmarks selected to capture different task types: instruction following (IFEval), multi-hop knowledge question answering (HotpotQA), and mathematical reasoning (GSM8K). For each model, the study uses 15 IFEval instances, 10 HotpotQA instances, and 10 GSM8K instances per model, resulting in 35 task instances per model and a total of 70 instances across the two models.

Table 1. Benchmark Summary

Benchmark	Inst.	Task Type	Metric
IFEval	15	Instruction following	Strict binary acc.
HotpotQA	10	Multi-hop knowledge QA	Token-level F1
GSM8K	10	Math word problems	Exact numeric match

Each task instance includes five context components: (i) task-specific system instructions, (ii) few-shot examples, (iii) retrieved documents, (iv) simulated conversation history, and (v) simulated tool outputs, yielding 32 context coalitions per instance. Total model evaluations are therefore 2,240 (). Raw results are stored as per-task JSON records with coalition scores, Shapley values, interactions, and axiom pass flags. 70×32

Primary analytical tables were derived from these logs and saved as summary CSV files (including component level Shapley summaries and pairwise interaction summaries). Figures were produced from the same summarized

dataset to maintain a single source of truth across numerical and graphical outputs. Data quality validation included full coalition coverage for each task, successful scorer runs for every coalition, and confirmation of the efficiency axiom for all profiled instances.

Table 2. Context Component Construction per Benchmark

Comp.	**IFEval**	**HotpotQA**	**GSM8K**
S: System	Format constraints	"Answer using documents"	"Solve step-by-step"
E: Examples	2–3 format demos	3 solved QA pairs	3 solved math problems
D: Documents	IF-following guide	Distractor passages	Arithmetic reference
H: History	Simulated dialogue	Simulated dialogue	Simulated dialogue
T: Tools	Validator status	Search results	Calculator output

5. RESULTS

Numerical Results

Component rankings shift across task types. Table 3 (mean Shapley values by benchmark and component) shows that component importance is strongly task-dependent rather than universal. In IFEval, system instructions dominate (mean $\phi(S) = 0.541$), while all other components are near zero or slightly negative. In HotpotQA, tool outputs ($\phi = 0.126$) and retrieved documents ($\phi = 0.096$) are the strongest positive contributors, whereas conversation history is negative ($\phi(H) = -0.065$). In GSM8K, no component is strongly positive, and retrieved documents are harmful on average ($\phi(D) = -0.065$). This pattern indicates that optimal context design should be benchmark/task-specific.

Table 3. Mean Shapley Values (std) by Component Across Benchmarks (GPT-5) $\pm$

Component	IFEval	HotpotQA	GSM8K
S: System Instr.	+0.541 ± 0.493	+0.086 ± 0.102	+0.035 ± 0.036
E: Few-Shot Ex.	+0.019 ± 0.076	+0.019 ± 0.090	+0.035 ± 0.036
D: Retrieved Docs	−0.014 ± 0.036	+0.096 ± 0.137	−0.065 ± 0.095
H: Conv. History	−0.003 ± 0.018	−0.065 ± 0.085	+0.010 ± 0.034
T: Tool Outputs	−0.009 ± 0.019	+0.126 ± 0.121	−0.015 ± 0.024

Retrieved documents harm mathematical reasoning. Table 4 (per-instance $\phi(D)$ on GSM8K) confirms the document effect is not an averaging artifact: in all non-trivial GSM8K cases (6/6 with non-zero contribution magnitude), retrieved documents are negative. This is an important unexpected finding because retrieval is commonly assumed to help. In this setting, additional reference text appears to interfere with mathematical reasoning.

Table 4. Per-Instance Shapley Value of Retrieved Documents on GSM8K

GSM8K Task	**φ(D)**	**Direction**
gsm8k_0000	0.000	neutral (trivial)
gsm8k_0001	0.000	neutral (trivial)
gsm8k_0002	−0.050	NEGATIVE
gsm8k_0003	−0.250	STRONGLY NEG.
gsm8k_0004	−0.033	NEGATIVE

gsm8k_0005	−0.033	NEGATIVE
gsm8k_0006	0.000	neutral (trivial)
gsm8k_0007	−0.233	STRONGLY NEG.
gsm8k_0008	−0.050	NEGATIVE
gsm8k_0009	0.000	neutral (trivial)
Non-trivial: 6/6 negative (100%)		

Significant pairwise interference. Table 5 (top pairwise interactions) highlights non-additivity in context composition. The strongest observed effect is negative D×T interaction on HotpotQA ($I = -0.222$), indicating destructive interference when both sources are included together. Additional benchmark-specific patterns include negative E×S interaction in IFEval and GSM8K, and positive E×H or H×T interactions in HotpotQA. The key inference is that adding individually useful components can still reduce total performance when pairwise interference is present.

Table 5. Top Pairwise Interaction Effects

Benchmark	**Pair**	**Mean I**	**Type**
HotpotQA	D × T	−0.222	Interference
HotpotQA	E × H	+0.122	Synergy
HotpotQA	H × T	+0.101	Synergy
GSM8K	E × S	−0.117	Interference
IFEval	E × S	−0.044	Interference

Table 6 highlights the cross-model numerical comparison (GPT-5 vs. Qwen3-8B), showing a capability-dependent pattern. On IFEval, Qwen3-8B is more dependent on system instructions (mean $\phi(S) = 0.811$ vs. 0.541 for GPT-5), while several non system components remain near zero or mixed. This suggests that context policy transfer across model classes should be done with care rather than assumed.

Table 6. Cross-Model Comparison on IFEval: GPT-5 vs. Qwen3-8B

Comp.	**GPT-5**	**Qwen3**	**Δ**	**Interpretation**
S: System	+0.541	+0.811	+0.270	Qwen3 more S-dep.
E: Examples	+0.019	+0.050	+0.031	Similar
D: Docs	−0.014	−0.028	−0.013	Both find D unhelpful
H: History	−0.003	−0.006	−0.002	Similar
T: Tools	−0.009	−0.028	−0.019	Both find T unhelpful

Graphical Results

Fig. 2 shows score saturation by coalition size. IFEval improves roughly linearly with more components, while GSM8K remains almost flat (high baseline, little benefit from components). Confidence bands show variance across task instances.

Fig. 3 shows per-instance Shapley values for all tasks. Each cell gives one component's value for one task, revealing per-task variation and "dictator" patterns with in IFEval. S=1.0

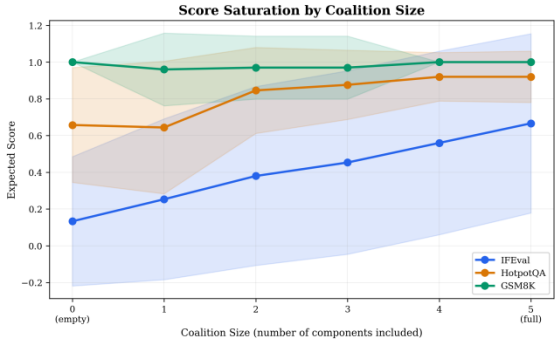

Fig. 2. Score saturation by coalition size. IFEval improves linearly; GSM8K is nearly flat with a high baseline.

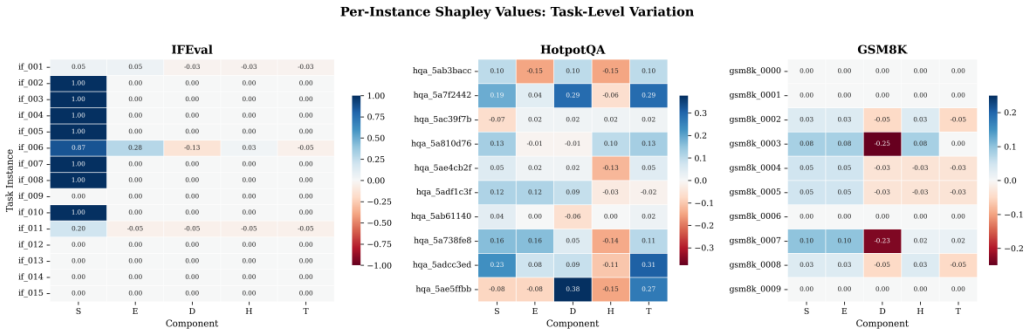

Fig. 3. Per-instance Shapley values across all task instances. Blue indicates positive contribution; red indicates negative. The IFEval "dictator" pattern () is clearly visible. $\phi(S)=1.0$

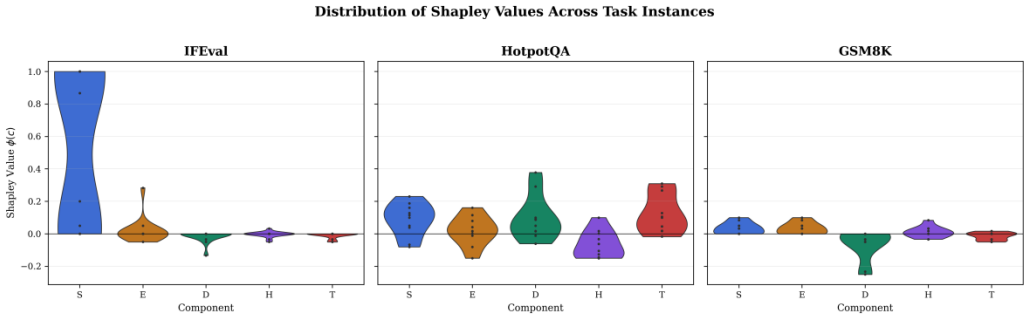

Fig. 4. Distribution of Shapley values across task instances. The bimodal distribution of S on IFEval reflects the dictator pattern; HotpotQA shows wider spreads.

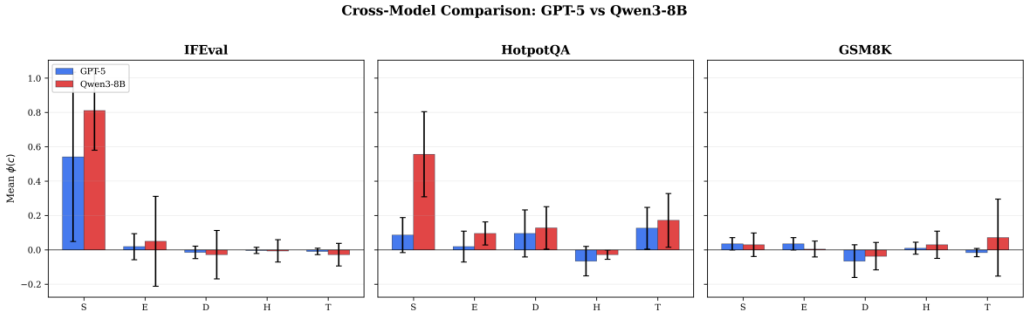

Fig. 5. Cross-model comparison: GPT-5 (blue) vs. Qwen3-8B (red). The weaker model shows dramatically higher system instruction dependence on HotpotQA.

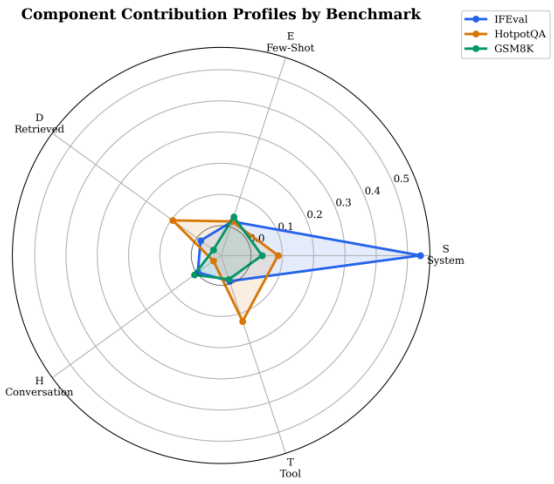

Fig. 6. Radar profiles showing each benchmark's component importance signature. IFEval is S-dominated; HotpotQA distributes across D and T; GSM8K is compact near zero.

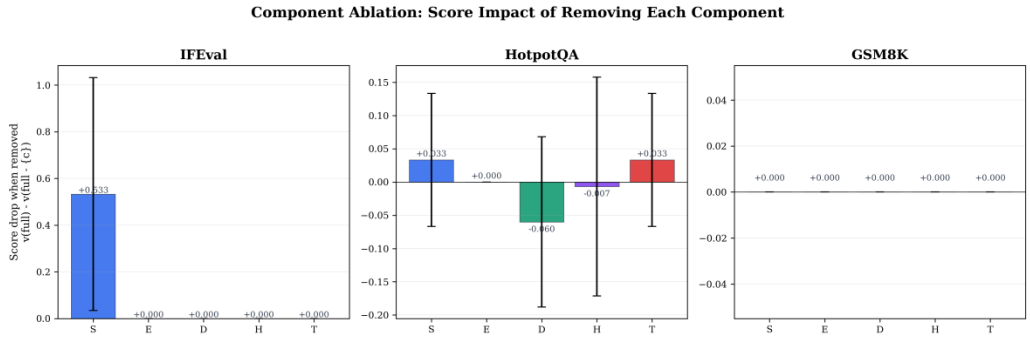

Fig. 7. Component ablation: score impact of removing each component from the full context. Removing S from IFEval causes a drop. +0.533

Fig. 4 shows violin plots of Shapley values per component. The bimodal distribution of S on IFEval, with values clustered at 0 and 1, reflects the dictator pattern. HotpotQA shows wider spreads indicating task-level variability.

Fig. 5 shows the cross-model comparison between GPT-5 and Qwen3-8B. The weaker model shows dramatically higher system instruction dependence on HotpotQA (vs.). +0.557+0.086

Fig. 6 shows radar profiles depicting each benchmark's component importance "signature." IFEval extends sharply toward S; HotpotQA distributes across D and T; GSM8K is compact near zero.

Fig. 7 highlights the component ablation effect—how the score drops when removing each component from the full context. Removing S from IFEval causes a drop; removing D from HotpotQA decreases performance by . +0.5330.060

6. DISCUSSION

Proposed Improvements

Based on the observed patterns, we propose a model-aware context policy layer:

- **Math-reasoning workflows** (GSM8K-like): Remove retrieved documents by default and include them only when problem-specific evidence suggests need.
- **Knowledge QA workflows** (HotpotQA-like): Hide non-critical conversation history to minimize distraction, and avoid injecting large amounts of D and T at the same time when their content significantly overlaps.

- **Instruction-following workflows** (IFEval-like): Prioritize clean, explicit system prompts; use few-shot/context additions only when they are empirically improve performance for the target model.
- **Deployment policy**: Maintain separate context templates for frontier models and smaller open models instead of relying on a single shared template.

To operationalize this, a practical next experiment is an ablation-controlled "policy-on/policy-off" rerun where each proposed rule is applied per benchmark and compared against baseline composition. Reported outputs should include: (i) absolute score delta by benchmark/model, (ii) Shapley redistribution after policy change, and (iii) interaction-magnitude reduction for known harmful pairs (especially DT on HotpotQA). ×

Source code and experiment logs are available at https://github.com/Ahamad-ai/ContextShapley.

7. CONCLUSIONS

This work presents ContextShapley as a reproducible framework for measuring context-component contribution and interference in LLM pipelines. The study: (i) presents exact component-level attribution based on Shapley values, (ii) quantifies pairwise interaction effects, (iii) demonstrates pronounced task sensitivity on IFEval, HotpotQA, and GSM8K, (iv) uncovers model-specific behavior in GPT-5 and Qwen3-8B, and (v) delivers an auditable experimental framework featuring complete coalition logging and axiom checking.

The unique contribution is not just identifying which components help or hurt performance, but demonstrating that context composition behaves non-additively and that detrimental interactions can arise even when each component is beneficial on its own. This shifts the field from heuristic driven toward a more measurement-driven approach to context engineering.

Practically, the findings support using benchmark and model-specific context policies instead of fixed universal templates. Methodologically, the framework offers a reusable protocol for future context studies across other model families and domains.

Future work should strengthen validation (with more instances), broaden cross-model scope (including additional frontier and open models), and assess adaptive context-selection controllers that proactively steer clear of harmful component interactions at inference time. This would transform descriptive attribution into practical, real-time optimization for deployed LLM systems.

8. ACKNOWLEDGMENT

We thank Osmania University for supporting this research.

We used OpenAI GPT-5 and Ollama Qwen3:8B as experimental subjects to generate LLM responses for all 2,240 coalition evaluations in Section V (Results and Discussion). GPT-5 was used for minor grammar and editing during manuscript preparation. All content was reviewed and edited by the authors, who take full responsibility for the final work.

9. DECLARATION OF STATEMENT

Funding

The authors received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Author Contributions

S. Ahamad conceived the study, designed and implemented the ContextShapley framework, conducted the experiments and analysis, and drafted the manuscript. Dr. V. B. Narsimha supervised the research and critically reviewed the manuscript. Both authors read and approved the final version.

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.

Data Availability Statement

The source code and full experiment logs that support the findings of this study are openly available in the ContextShapley repository at https://github.com/Ahamad-ai/ContextShapley.

REFERENCES

1. Anthropic, "Effective context engineering for AI agents," Anthropic Engineering Blog, 2025.
2. J. Wang et al., "A survey of context engineering for large language models," arXiv preprint, arXiv:2507.13334, 2025.
3. H. Jiang et al., "LLMLingua: Compressing prompts for accelerated inference of large language models," Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP), 2023.
4. N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang, "Lost in the middle: How language models use long contexts," Trans. Assoc. Comput. Linguistics, vol. 12, pp. 157–173, 2024.
5. Chroma Research, "Context rot: Evaluating long-context degradation in frontier models," Chroma Research Report, 2025.
6. J. Lee et al., "Context length alone hurts LLM performance despite perfect retrieval," Findings of the Assoc. Comput. Linguistics: EMNLP, 2025.
7. T. Gao, H. Yen, J. Yu, and D. Chen, "ContextCite: Attributing model generation to context," arXiv preprint, arXiv:2409.00729, 2024.
8. L. S. Shapley, "A value for n-person games," Contributions to the Theory of Games, vol. 2, no. 28, pp. 307–317, 1953.
9. S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," Advances in Neural Information Processing Systems, vol. 30, 2017.
10. Y. Chen et al., "Agentic context engineering," Proc. Int. Conf. Learning Representations (ICLR), 2026.
11. T. Li et al., "ContextBudget: Budget-aware context management for long-horizon agents," arXiv preprint, arXiv:2604.01664, 2026.
12. S. Kim et al., "The distracting effect: Understanding irrelevant passages in RAG," arXiv preprint, arXiv:2505.06914, 2025.
13. P. Xu et al., "Long-context LLMs meet RAG: Overcoming challenges for long-form generation," Proc. Int. Conf. Learning Representations (ICLR), 2025.
14. A. Ghorbani and J. Zou, "Data Shapley: Equitable valuation of data for machine learning," Proc. Int. Conf. Machine Learning (ICML), 2019.
15. Madhu Nakerekanti, VB Narasimha" Analysis on malware issues in online social networking sites", 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS).
16. Prabhadevi Cheruku, VB Narasimha" Optimizing identity and access management through 1D-SCNN-based anomaly detection", Journal of Applied Research on Industrial Engineering, 2024.
17. K Rajendraprasad, VB Narasimha "Steganography image detection using different steganalysis techniques with Markov chain features",2016.
18. A Rakesh Phanindra, VB Narasimha, Ch V PhaniKrishna" A review on application security management using web application security standards" Springer ,2018,
19. Madhu Nakerekanti, Dr VB Narsimha" Secured multiparty access control model for online social networking using machine learning",107,2022.

Author Biographies

Shaik Ahamad is a Research Assistant at the University College of Engineering(A), Osmania University. He holds a Master's degree in Computer Applications from Osmania University. His research interests span Context Engineering, Large Language Models (LLMs), Generative AI, and Applied AI, with a focus on building reliable, interpretable, and efficient AI systems. He is particularly interested in how modern language-model pipelines assemble and use context—including retrieval-augmented generation, prompt and context optimization, and evaluation of LLM behavior. Grounded in a strong foundation of Data Science, Statistics, and Data Analytics, he has undertaken numerous projects applying data-driven techniques to solve complex problems and enhance decision-making. He continues to contribute to the advancement of knowledge and technology across AI, LLMs, and context-aware language modeling.

Dr. V B Narsimha is an Assistant Professor in the Department of Computer Science Engineering at the University College of Engineering (A), Osmania University, Hyderabad, India. His research interests span Artificial Intelligence, Machine Learning, Large Language Models (LLMs), Data Mining, and Data Analytics, along with Computer Networks, Cloud Computing, and information security. His work focuses on applying data-driven and intelligent techniques to solve real-world problems in secure, scalable, and reliable computing systems. He has authored numerous peer-reviewed articles in reputed international journals and conferences, and actively guides research across emerging areas of AI and language technologies. He continues to contribute to the advancement of knowledge and innovation in AI, machine learning, and intelligent data systems.