AGENT THAT MATTERS: AN ATTRIBUTION FRAMEWORK FOR MULTI-AGENT LLMS

Mingyu Lu[*], **Yushan Hung**[*], **Su-In Lee**
Paul G. Allen School of Computer Science & Engineering
University of Washington
{mingyulu, yushan13, suinlee}@cs.washington.edu

ABSTRACT

Quantifying individual agent contributions in LLM-based multi-agent systems (MAS) is critical for optimizing architectural efficiency and mitigating functional redundancy. We therefore introduce a game-theoretic attribution framework that formalizes MAS collaboration as a cooperative game, enabling the decomposition of system utility into principled contribution scores. Through the empirical evaluation of MetaGPT on the HumanEval and MBPP benchmarks, we identify the Product Manager as a dominant veto player, while the QA Engineer exhibits negligible or negative marginal impact. Our results show that in hierarchical MAS, Leave-One-Out (LOO) attribution serves as a reliable proxy for more complex axiomatic estimators like Shapley and Banzhaf values. Finally, we demonstrate that in-context removal fails as a faithful substitute for exact removal and increases token usage by a significant amount. Together, these findings demonstrate that framing MAS as a cooperative game is a promising direction for credit assignment, providing a rigorous foundation for diagnosing architectural bottlenecks and optimizing resource allocation in complex multi-agent workflows.

1 INTRODUCTION

In recent years, the reframing of large language models (LLMs) as agents, along with the development of multi-agent systems (MAS) composed of interactive, LLM-powered agents working toward shared goals, has attracted significant attention (Hong et al., 2023; Li et al., 2023). These agentic systems have shown strong potential across a range of domains, such as role-based software engineering workflows for code generation (Hong et al., 2023). However, as these systems grow in complexity, understanding the importance of individual components becomes a critical prerequisite for system optimization (Liu et al., 2023a;b). While recent work estimates component contributions on an individual input level (Liu et al., 2023b; Yang et al., 2025), quantifying each agent's marginal contribution to overall system utility (e.g., accuracy) over the task distribution remains underexplored.

To address this problem, recent studies (Hong et al., 2023; Cui et al., 2025) have turned to removal-based frameworks, primarily utilizing Leave-One-Out (LOO) ablation (Cook et al., 1982). LOO measures an agent's marginal contribution by observing system performance degradation upon its removal from the full coalition. To improve evaluation efficiency, Cui et al. (2025) proposed approximating these removals via in-context introspective simulation, prompting the system to mimic an agent's absence without exact removal. However, these evaluations remain largely confined to LOO removals, leaving more principled measures grounded in cooperative game theory, such as the Shapley value (Shapley et al., 1953) or Banzhaf value (Dubey & Shapley, 1979), unexplored. Consequently, it remains unknown if introspective simulation maintains counterfactual fidelity when multiple agents are removed simultaneously, as required for rigorous coalitional attribution.

To answer these questions, we establish a removal-based framework for agent attribution grounded in cooperative game theory. By formalizing MAS evaluation as a characteristic function game, we provide a rigorous mathematical foundation for decomposing aggregate utility into discrete individual contributions. This approach enables a systematic assessment of agent utility across diverse removal

[*]Equal contribution.

distributions, including LOO, Banzhaf, and Shapley values. We apply this framework to MetaGPT, a software engineering MAS, to evaluate agent utility and assess whether in-context removal can effectively approximate exact removal. Our empirical analysis compares these protocols across multiple performance metrics and computational costs. Our empirical analysis reveals that in-context learning often fails to maintain counterfactual fidelity; furthermore, it can, paradoxically, increase computational latency and resource overhead compared to standard removal.

2 Problem Setup & Method

2.1 The MAS Game

We propose a formal framework that treats Multi-Agent System (MAS) evaluation as a removal game. This formulation allows us to leverage axiomatic attribution methods to rigorously quantify agent importance by decomposing global outcomes into individual contributions.

Definition 1. *(MAS Game): A Multi-Agent System is defined by the tuple $(\mathcal{A}, \nu)$, where $\mathcal{A} = \{a_1, \ldots, a_n\}$ is the set of n agents and $\nu : 2^{\mathcal{A}} \to \mathbb{R}$ is the characteristic utility function. Given a task distribution $\mathcal{D}$, the utility of a subset (coalition) $S \subseteq \mathcal{A}$ is defined as:*

$$\nu(S) = \mathbb{E}_{x \sim \mathcal{D}}[\mathcal{G}(\tau_{S,x})]$$

where $\tau_{S,x}$ represents the execution trace (or system output) generated by the subset of agents S on input x, and $\mathcal{G} : \mathcal{T} \to \mathbb{R}$ is a global behavior mapping that quantifies any arbitrary system property, such as task success, computational latency, or resource consumption.

2.2 Quantifying Agent Importance via Cooperative Game Theory

The objective of our framework is to derive an attribution vector $\boldsymbol{\phi} \in \mathbb{R}^n$, where each component ϕ_i quantifies the contribution of agent a_i to the aggregate system utility $\nu(\mathcal{A})$. Central to this quantification is the marginal contribution of an agent a_i to a specific coalition S:

$$\Delta_i(S) = \nu(S \cup a_i) - \nu(S), \quad a_i \notin S. \tag{1}$$

To provide a robust evaluation of agent importance, we define a unified attribution framework where various game-theoretic approaches are expressed as a weighted sum of marginal contributions across the power set of coalitions:

$$\phi_i = \sum_{S \subseteq \mathcal{A} \setminus a_i} w(|S|)\Delta_i(S) \tag{2}$$

The weighting function $w(|S|)$ allows us to interpret agent utility through distinct distribution:

- **Leave-One-Out (LOO)** ($w(|S|) = 1$ if $S = \mathcal{A} \setminus \{i\}$, else 0) serves as a computationally efficient baseline that measures the impact of an agent on the full coalition $\mathcal{A}$. While computationally efficient, this method may fail to capture interaction among players.
- **Shapley Value** (Shapley et al., 1953): ($w(|S|) = \frac{|S|!(n-|S|-1)!}{n!}$) is the unique distribution satisfying the efficiency axiom, ensuring the total utility is fully allocated among agents ($\sum \phi_i = \nu(\mathcal{A})$). Despite its theoretical guarantees, its exact computation is often prohibitively expensive due to the combinatorial complexity of the power set.
- **Banzhaf Value** (Dubey & Shapley, 1979) ($w(|S|) = \frac{1}{2^{n-1}}$) treats every possible coalition with equal importance. As a semi-value (Harsanyi & Harsanyi, 1982), it provides an average-case expectation of an agent's marginal utility. While it effectively identifies "critical" agents, it lacks the efficiency property, meaning the sum of individual attributions does not necessarily reconstruct the total system utility.

Computational consideration for MAS While computing the full power set 2^n is generally intractable for large systems, modern MAS architectures often rely on structurally indispensable orchestrators. Our framework formalizes this dependency by defining a set of mandatory agents $M \subset \mathcal{A}$, such that $\nu(S) = 0$ for any coalition where $M \not\subseteq S$. This constraint prunes the evaluation space to $2^{n-|M|}$ coalitions. By focusing on these operationally viable subsets, we maintain computational tractability across the task distribution $\mathcal{D}$ while strictly respecting the system's underlying architectural logic.

2.3 REMOVAL PROTOCOL IN MAS

A central challenge in MAS attribution is the operationalization of the removal operator, specifically, how to evaluate $\nu(S)$ when $S \subset \mathcal{A}$. While game theory provides the weights $w(|S|)$, the validity of the resulting attribution depends entirely on the counterfactual fidelity of the removal protocol. We distinguish between two paradigms:

Exact Removal (Ground Truth): This protocol physically excludes agents $\mathcal{A} \setminus S$ from the environment, re-initializing and executing the system with only the subset S, (Figure 1). While it yields ground-truth marginal contributions, the requirement for repeated re-executions across the power set makes it computationally intensive.

Introspective Removal (Simulation): Following Cui et al. (2025), this protocol leverages in-context learning (ICL) to simulate agent absence. Instead of exact removal, agents are prompted to ignore or rethink the contributions of specified peers within the existing execution context (Figure 2). While this avoids re-execution overhead, its counterfactual fidelity remains unverified and is a core focus of our investigation.

By defining the MAS game and removal protocols, our framework provides a principled approach for decomposing any utility function of interests into individual agent contributions.

3 EXPERIMENTS SETUP

We leverage our framework to investigate three core research questions: (i) Synergy: Do the Shapley and Banzhaf values uncover critical agent dependencies and redundancies that remain latent under standard LOO ablation? (ii) Fidelity: To what extent can introspective removal faithfully approximate the ground-truth marginal contributions of exact removal? (iii) Efficiency: Does the introspective simulation yield a significant net gain in computational efficiency?

3.1 METAGPT

Agent Configuration We evaluate our framework on MetaGPT (Hong et al., 2023), a role-based multi-agent framework designed for collaborative software engineering. The system simulates a professional development pipeline by assigning LLM agents to specialized roles: Product Manager, Architect, Project Manager, Engineer, and QA Engineer. These agents interact through a structured message-passing interface to transform natural language requirements into executable code.

Task and utility function We evaluate system performance on the HumanEval (Chen, 2021) and MBPP (Austin et al., 2021) benchmarks. We define the characteristic function $\nu(S)$ as the pass@1 score, the probability that a generated solution passes all unit tests on the first attempt, attained by a specific coalition $S \subseteq \mathcal{A}$. To investigate the relationship between performance and resource allocation, we further define an auxiliary cost function based on token consumption.

For all experiments, we utilize Gemini 2.0 Flash from OpenRouter [1] as the backbone model. To ensure statistical significance, we conduct three independent trials for each coalition and report the average performance across runs.

4 RESULTS: AGENT ATTRIBUTION ANALYSIS OF METAGPT

Here, we provide a detailed analysis of the agent attribution of MetaGPT.

4.1 EXACT REMOVAL ATTRIBUTION RESULTS

Results under the exact removal reveal a highly concentrated importance distribution within MetaGPT, with the Product Manager (PM) emerging as a dominant veto player. As shown in Table 1, the PM's marginal contribution to Pass@1 consistently eclipses that of the second-highest contributor, e.g., Engineer in HumanEval or Architect in MBPP, across all axiomatic distributions. Conversely, the

[1] `https://openrouter.ai/google/gemini-2.0-flash-001`

Table 1: Agent attribution of MetaGPT computed via Shapley, Banzhaf, and LOO. The rightmost column reports the Spearman rank correlation between introspective and exact removal.

Method	ProductManager	Architect	ProjectManager	Engineer	QaEngineer	Spearman corr.
HumanEval (pass@1)						
Shapley	0.564	0.040	0.033	0.043	0.030	0.30
Banzhaf	0.661	0.049	0.043	0.051	0.040	0.30
LOO	0.709	0.012	-0.004	0.022	-0.012	0.35
MBPP (pass@1)						
Shapley	0.536	0.051	-0.004	0.009	0.016	0.25
Banzhaf	0.518	-0.001	-0.011	-0.005	0.002	0.20
LOO	0.566	0.030	0.040	0.070	0.086	0.40

Rank: 1st, 2nd, 3rd, 4th, 5th

near-zero or marginally negative scores observed for the QA Engineer suggest either functional redundancy or significant coordination overhead when operating within the full coalition.

Our analysis also shows that the agent importance ranking remains invariant across LOO, Banzhaf, and Shapley distributions in HumanEval: PM $>$ Engineer $>$ Architect $>$ Project Manager $>$ QA Engineer. This stability implies that within the rigid hierarchical workflows typical of systems like MetaGPT, the computationally efficient LOO baseline may serve as a reliable proxy for more complex power-set attributions.

Finally, we investigate the alignment between task utility and resource expenditure Table 2. We calculated the Spearman rank correlation (r_s) between attribution vectors for Pass@1 and token usage, yielding high consistency across all indices ($r_s = 0.7, 0.5$, and 0.9 for Shapley, Banzhaf, and LOO, respectively). This strong monotonic relationship confirms that MetaGPT's architectural design effectively concentrates computational effort on performance-critical roles, ensuring that the primary drivers of system success are also the primary drivers of resource usage.

4.2 Fidelity and Computational Costs of Introspective Simulation

We next evaluate the fidelity of Cui et al. (2025) by comparing utilities from in-context removal (Figure 2) to exact removal (Figure 1). Surprisingly, our results reveal a weak correlation between the two methods, with average Spearman rank coefficients (r_s) of 0.27, 0.25, and 0.37 for the Shapley value, Banzhaf value, and LOO, respectively, across all datasets (Table 1). These findings suggest that in-context removal fails to faithfully mimic the counterfactual state of agent absence when evaluating global system behavior. Furthermore, we identify a computational paradox: the Introspective protocol incurs significantly higher token consumption than Exact Removal by 169 times Table 3. This overhead is driven by increased prompt complexity and the necessity of re-processing historical traces during the "rethinking" phase. Consequently, while non-invasive, introspective simulation currently lacks the fidelity and cost-efficiency required for robust agent attribution in complex MAS workflows.

5 Conclusion

In this work, we propose a formal attribution framework for LLM-based multi-agent systems (MAS) grounded in cooperative game theory. By formalizing agent collaboration as a characteristic function game, our approach enables a rigorous, axiomatic decomposition of system utility into individual contributions. Empirical evaluation on MetaGPT reveals a concentrated importance distribution where the Product Manager acts as a dominant veto player, while roles like the QA Engineer exhibit negligible or negative marginal utility. We find that while agent rankings remain consistent across LOO, Banzhaf, and Shapley values within MetaGPT. Our results also show that introspective removal fails to faithfully replicate ground-truth removal and even introduces an increase in token usage. Ultimately, our framework provides a robust tool for identifying influential agents, offering a principled methodology for investigating MAS. Future work will focus on developing high-fidelity acceleration methods that can approximate exact removal without the prohibitive computational costs of current in-context approaches.

REFERENCES

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Mark Chen. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

R Dennis Cook, Norton Holschuh, and Sanford Weisberg. A note on an alternative outlier model. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 44(3):370–376, 1982.

Yue Cui, Liuyi Yao, Zitao Li, Yaliang Li, Bolin Ding, and Xiaofang Zhou. Efficient leave-one-out approximation in llm multi-agent debate based on introspection. *arXiv preprint arXiv:2505.22192*, 2025.

Pradeep Dubey and Lloyd S Shapley. Mathematical properties of the banzhaf power index. *Mathematics of Operations Research*, 4(2):99–131, 1979.

John C Harsanyi and John C Harsanyi. A simplified bargaining model for the n-person cooperative game. *Papers in game theory*, pp. 44–70, 1982.

Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, et al. Metagpt: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*, 3(4):6, 2023.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in Neural Information Processing Systems*, 36:51991–52008, 2023.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv:2308.03688*, 2023a.

Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. Dynamic llm-agent network: An llm-agent collaboration framework with agent team optimization. *arXiv preprint arXiv:2310.02170*, 2023b.

Lloyd S Shapley et al. A value for n-person games. 1953.

Yingxuan Yang, Bo Huang, Siyuan Qi, Chao Feng, Haoyi Hu, Yuxuan Zhu, Jinbo Hu, Haoran Zhao, Ziyi He, Xiao Liu, et al. Who's the mvp? a game-theoretic evaluation benchmark for modular attribution in llm agents. *arXiv preprint arXiv:2502.00510*, 2025.

Table 2: Agent token attribution in HumanEval (Millions). Values represent the marginal contribution of each agent to total token expenditure; negative values indicate that an agent's presence reduces overall system cost.

Method	ProductManager	Architect	ProjectManager	Engineer	QaEngineer
Exact Removal					
Shapley	3.59M	0.26M	0.28M	0.27M	0.24M
Banzhaf	4.18M	0.33M	0.37M	0.36M	0.34M
LOO	4.64M	0.06M	0.01M	0.01M	-0.11M
Introspective Removal					
Shapley	9,641M	579M	585M	590M	596M
Banzhaf	11,311M	752M	760M	765M	770M
LOO	11,993M	-145M	-135M	-124M	-114M

Table 3: Total token usage across all 2^5 subsets.

Removal protocol	Total token usage	Increase vs. exact
Exact removal	66M	–
Introspective removal	180,988M	2704.35×

A Appendix

A.1 Weighting Functions for Marginal Contribution

Leave-One-Out (LOO) LOO attribution measures the marginal contribution of an agent by comparing the utility of the full coalition with the utility of the coalition from which the agent is removed. Formally, the LOO score for agent $a_i \in \mathcal{A}$ is defined as:

$$\phi_{\mathrm{LOO}}(\nu, \mathcal{A})_i = \nu(\mathcal{A}) - \nu(\mathcal{A} \setminus \{a_i\}), \tag{3}$$

Banzhaf Value (Dubey & Shapley, 1979) The Banzhaf value is a *semivalue* that measures an agent's average marginal contribution over all possible coalitions that exclude it, assigning equal weight to each. It satisfies all Shapley axioms except *efficiency*. Formally:

$$\phi_{\mathrm{Banzhaf}}(\nu, \mathcal{A})_i = \mathbb{E}_{S \sim w_{\mathrm{Banzhaf}}} \left[\nu(S \cup \{a_i\}) - \nu(S) \right], \tag{4}$$

where w_{Banzhaf} denotes the uniform distribution over subsets $S \subseteq \mathcal{A} \setminus \{a_i\}$.

Shapley Value (Shapley et al., 1953) The Shapley value is the unique distribution satisfying *linearity*, *dummy player*, *symmetry*, and *efficiency*. Formally, for an agent a_i and utility function ν, the value $\phi_{\mathrm{Shapley}}(\nu, \mathcal{A})_i$ is defined as:

$$\phi_{\mathrm{Shapley}}(\nu, \mathcal{A})_i = \frac{1}{n} \sum_{S \subseteq \mathcal{A} \setminus \{a_i\}} \binom{n-1}{|S|}^{-1} \left(\nu(S \cup \{a_i\}) - \nu(S) \right) \tag{5}$$

where the weighting term $\binom{n-1}{|S|}^{-1}$ accounts for all possible coalition sizes. Intuitively, it captures the average marginal contribution of an agent across all potential subsets.

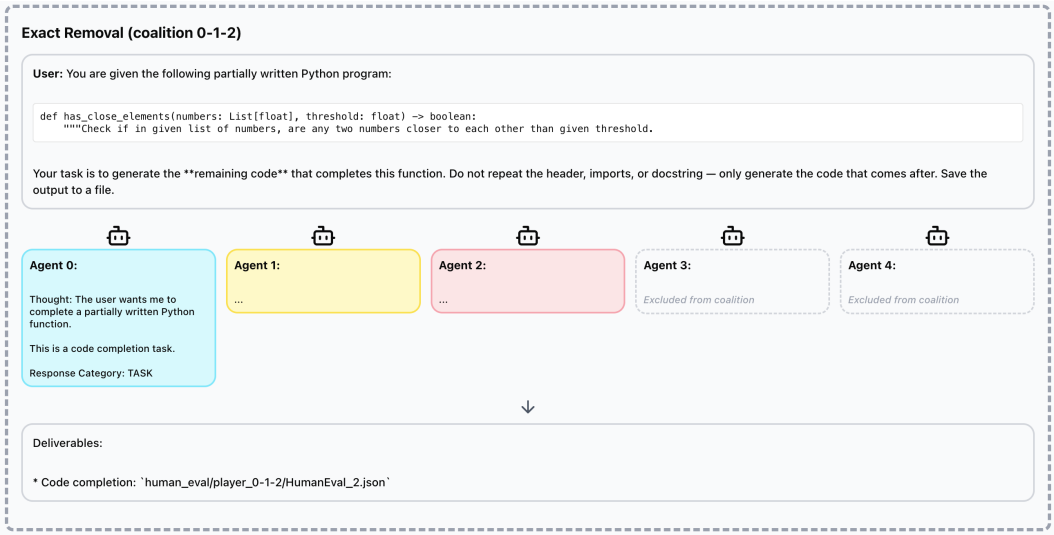

Figure 1: Exact Removal Protocol

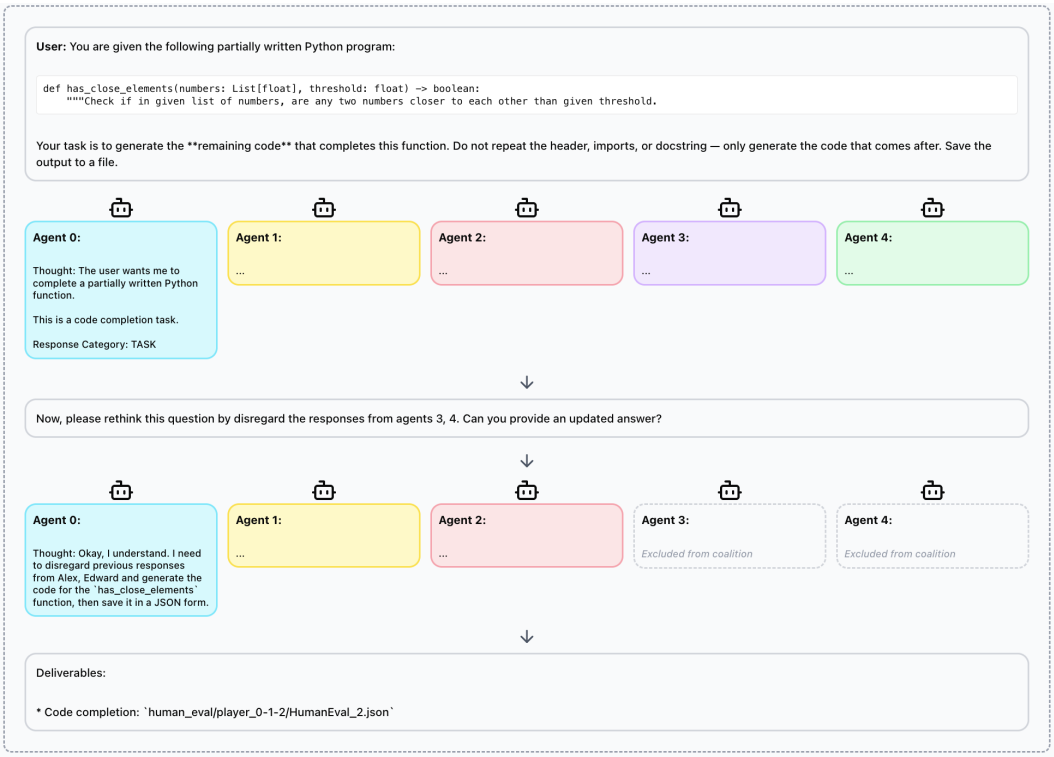

Figure 2: Introspective Removal Protocol