Every Response Counts: Quantifying Uncertainty of LLM-based Multi-Agent Systems through Tensor Decomposition

Anonymous ACL submission

Abstract

While Large Language Model-based Multi-Agent Systems (MAS) consistently outperform single-agent systems on complex tasks, their intricate interactions introduce critical reliability challenges arising from communication dynamics and role dependencies. Existing Uncertainty Quantification methods, typically designed for single-turn outputs, fail to address the unique complexities of the MAS. Specifically, these methods struggle with three distinct challenges: the cascading uncertainty in multi-step reasoning, the variability of inter-agent communication paths, and the diversity of communication topologies. To bridge this gap, we introduce MATU, a novel framework that quantifies uncertainty through tensor decomposition. MATU moves beyond analyzing final text outputs by representing entire reasoning trajectories as embedding matrices and organizing multiple execution runs into a higher-order tensor. By applying tensor decomposition, we disentangle and quantify distinct sources of uncertainty, offering a comprehensive reliability measure that is generalizable across different agent structures. We provide comprehensive experiments to show that MATU effectively estimates holistic and robust uncertainty across diverse tasks and communication topologies.

1 Introduction

While multi-agent systems (MAS), where multiple LLM-based agents collaborate, consistently outperform single-agent systems on complex tasks, their complex interactions introduce critical and MAS-specific reliability challenges (Li et al., 2023; Wu et al., 2024; Wang et al., 2025; Zhang et al., 2024). Uncertainty in these systems emerges not from a single agent's isolated error, but from the complex dynamics of communication, role dependencies, and consensus-building. A minor, early mistake can irrevocably cascade through the collaboration. Therefore, Uncertainty Quantification (UQ) for MAS is critical, especially when MAS is applied to high-stakes domains such as scientific discovery (Lu et al., 2024), healthcare decision support (Kim et al., 2024; Tang et al., 2023), and autonomous planning (Silva and Macharet, 2025). For example, in a medical context, an initial agent's misdiagnosis can steer the entire pipeline toward a confidently asserted but dangerously flawed treatment plan and a reliable UQ method could help to mitigate such risks.

Uncertainty estimation itself is not a new concern in machine learning. For decades, it has been a fundamental part of supervised learning tasks such as regression (Ye et al., 2024) and classification (Gal and Ghahramani, 2016; Sensoy et al., 2018). However, the landscape changes dramatically in the era of Large Language Models and their deployment as agents. Unlike traditional supervised tasks, LLMs must generate free-form text. This generative nature introduces new uncertainty factors that go beyond classical classification or regression. Recent work has therefore proposed specialized UQ methods for LLMs, focusing on semantic consistency such as semantic entropy (Kuhn et al., 2023) and graph-based methods (Lin et al., 2023; Da et al., 2024). All these methods rely on natural language inference (NLI) models (MacCartney, 2009) to capture the similarity between the answers. While these techniques have proven useful, most of them concentrate on single-turn outputs from a standalone model.

In contrast, the setting of LLM-based agents, especially multi-agent systems, raises a new class of challenges: (1) Multi-step reasoning: Many current UQ frameworks measure uncertainty by assessing outputs' semantic consistency with NLI models (Kuhn et al., 2023; Lin et al., 2023). This approach fails in the context of multi-step reasoning for two key reasons. First, applying it only to the final output ignores the rich uncertainty information embedded in the reasoning process. Second,

a naive attempt to fix this by concatenating entire reasoning trajectories into long documents makes it difficult for NLI models, which are typically designed for sentence-pair tasks and have context limitations. More importantly, in MAS, uncertainty is distributed across heterogeneous agents; an NLI model cannot distinguish whether a contradiction arises from an individual agent's hallucination or a logical misalignment between two different agents during a handoff. (2) Inter-agent communication diversity: For the same query, agents may collaborate through different sequences of interactions across runs. However, UQ methods that focus on semantic diversity are blind to this path diversity. (3) Communication topology diversity: Existing UQ methods are designed and validated for single models, which represent a fixed computational structure. In the MAS ecosystem, however, systems are built with diverse communication topologies. The effectiveness of a UQ method developed for a single model is highly unknown when applied to these varied and complex multi-agent structures.

In this paper, we take a pioneering step toward uncertainty estimation for LLM-based multi-agent systems by introducing a novel UQ framework of **M**ulti-**A**gent **T**ensor **U**ncertainty. To address the challenge of multi-step reasoning, MATU moves beyond analyzing only the final text, instead representing each agent's entire reasoning trajectory as an embedding matrix. To address the challenge of inter-agent communication diversity, we aggregate multiple runs of the same query to capture variability in how agents interact and exchange information. To address the challenge of communication topology diversity, MATU organizes all collected trajectories and runs into a higher-order tensor, which is inherently generalizable across different communication structures. This three-dimensional tensor, which is composed of agents, reasoning steps, and sampling runs, provides a holistic and generalizable way to represent the system's behavior. We can then apply tensor decomposition to disentangle and quantify the distinct sources of uncertainty, offering a comprehensive reliability measure at both the response and system levels.

- We provide the first systematic definition of uncertainty quantification for LLM-based multi-agent systems, identifying unique sources of uncertainty introduced by tool usage, multi-step reasoning, and inter-agent communication in MAS.
- We design `MATU`, a tensor decomposition-based framework that integrates multi-agent uncertainty signals at both response and run levels, enabling holistic uncertainty estimation of multi-agent systems.
- We conduct extensive experiments across diverse tasks with or without tool-usage and communication topologies, and further provide case analyses that illustrate how different dimensions of uncertainty interact, demonstrating the need for dealing with the new challenge of UQ in multi-agent systems.

2 Related Work

LLM-based Agents LLMs have evolved into agents capable of solving diverse tasks, including web search (Nakano et al., 2021; Deng et al., 2023), software development (Wang et al., 2021; Yang et al., 2024a), and complex reasoning (Gao et al., 2023; Chen et al., 2022), by leveraging tools and historical memory (Yao et al., 2023; Park et al., 2023). While single agents are effective, multi-agent systems (MAS) demonstrate superior performance through collaboration (Li et al., 2023; Wu et al., 2024). These systems employ varied communication topologies, ranging from static designs (Li et al., 2023; Qian et al., 2023; Hong et al., 2023; Holt et al., 2023; Zhou et al., 2023) to dynamic structures (Zhuge et al., 2024; Liu et al., 2023; Zhang et al., 2024; Wang et al., 2025). However, the complexity of these interactions poses new challenges for trustworthiness, necessitating uncertainty estimation methods that generalize across diverse agent topologies.

Uncertainty for Large Language Model While uncertainty quantification is established for traditional regression and classification (Ye et al., 2024; Amini et al., 2020; Sensoy et al., 2018; Ovadia et al., 2019), LLMs' open-ended generation requires distinct approaches. Semantic entropy (Kuhn et al., 2023) addresses this but necessitates access to token probabilities. For black-box settings, recent works estimate uncertainty by analyzing the semantic consistency of generated responses (Lin et al., 2023; Chen and Mueller, 2024; Da et al., 2024; Gao et al., 2024; Hou et al., 2024). These methods typically leverage NLI models to construct similarity matrices and derive uncertainty metrics from graph Laplacian eigenvalues (Lin et al., 2023; Chen and Mueller, 2024; Da et al., 2024; Catak and Kuzlu, 2024), or by integrating

multiple uncertainty sources (Chen et al., 2025).

However, research on UQ for agent systems remains sparse, with Kirchhof et al. (2025) identifying key gaps in interactive and underspecification uncertainties. Currently, SAUP (Zhao et al., 2025) stands as the primary approach, employing situational weights for step-wise analysis. However, it treats steps independently, overlooking the holistic uncertainty of complete reasoning trajectories. In this paper, our work bridges this gap by integrating both response-level and run-level dynamics for a more reliable estimation of uncertainty

3 Background

Uncertainty Quantification (UQ) for LLM-based agents extends beyond single-agent settings. In a multi-agent system (MAS), a set of K agents $\{\mathcal{M}_1, \ldots, \mathcal{M}_K\}$ collaboratively generate trajectories through communication and multi-step reasoning. For different agents, the model parameter θ_k could be the same or different according to different designs. For different collaboration styles, the input of agent M_i might also be different. For example, in a roundabout communication topology, the input might be the discussion contexts from other agents, while the input might be the assignment in a star communication topology.

Considering previous UQ works on LLMs, repeated generations are key for the black-box UQ. Therefore, here we also define the repeated generations for MAS S. Given an input x, the j-th run of the MAS produces a trajectory $\tau^{(j)} = \{y_{1:T_k}^{(j,k)}\}_{k=1}^{K}$,

where $y_{1:T_k}^{(j,k)}$ denotes the sequence of outputs from agent k during run j, and T_k is the number of steps taken by agent k. For one task x, everything will be fixed, including the role and parameters of agents and communication topology. Then, across N repeated runs of the MAS, we collect a set of trajectories $\mathcal{T} = \{\tau^{(1)}, \tau^{(2)}, \ldots, \tau^{(N)}\}$.

Problem 1 (Multi-agent Uncertainty)**.** *Given an input x and a set of trajectories $\mathcal{T}$ generated by a MAS across N runs, the goal is to compute an uncertainty score U that reflects the variability across $\mathcal{T}$. Formally,*

$$U = \mathcal{F}(x, \mathcal{T}),$$

where $\mathcal{F}$ is an aggregation functional that maps the input and the trajectory set to a scalar value measuring the overall uncertainty. A lower U indicates that the MAS consistently produces stable and reliable trajectories, while a higher U suggests divergent reasoning, unstable communication, or fragile collaboration among agents.

Note that in our definition, each trajectory $\tau^{(j)}$ consists of agent-specific sequences $y_{1:T_k}^{(j,k)}$, where the horizon length T_k may differ across agents and runs. This variability captures the intrinsic challenges of multi-step reasoning, since errors made in earlier steps can propagate differently depending on the trajectory length, and it also reflects the diversity of communication topologies, where agents may follow different interaction patterns.

4 Method

In this section, we present our method MATU in detail. MATU is designed for uncertainty quantification of any general multi-agent system in a black-box setting by analyzing the embedding matrices from running trajectories using tensor decomposition. The overall pipeline of MATU can be found at Fig. 1, and we start the introduction by embedding.

4.1 Embedding for Multi-step Reasoning

Multi-step reasoning poses a fundamental challenge in uncertainty quantification for multi-agent systems. Unlike single-turn settings, where the model outputs a single sentence, multi-agent reasoning unfolds as a sequence of intermediate steps. Errors introduced in earlier steps can cascade through subsequent ones, while different agents may take trajectories of varying lengths depending on their roles or communication topologies. This variability makes it difficult to directly compare trajectories across repeated runs.

To overcome these challenges, we encode each intermediate output, whether a natural language sentence or a tool call result, into a shared latent space using pre-trained embedding models. In detail, we treat tool call results as a string as well and use a text embedding model such as Qwen3-Embedding-0.6B. Formally, for the t-th step in trajectory $\tau^{(j)}$, we define $e_t^{(j)} \in \mathbb{R}^d$, where d is the embedding dimension. By concatenating the embeddings across all steps in trajectory j and agent k, we construct an embedding matrix $E^{(j,j)} \in \mathbb{R}^{T_{j,k} \times d}$, where $T_{j,k}$ denotes the number of steps in that trajectory and agent. This embedding construction mitigates the core difficulties of multi-step reasoning by mapping heterogeneous, variable-length, and modality-mixed outputs to a fixed-dimensional semantic space at the step level,

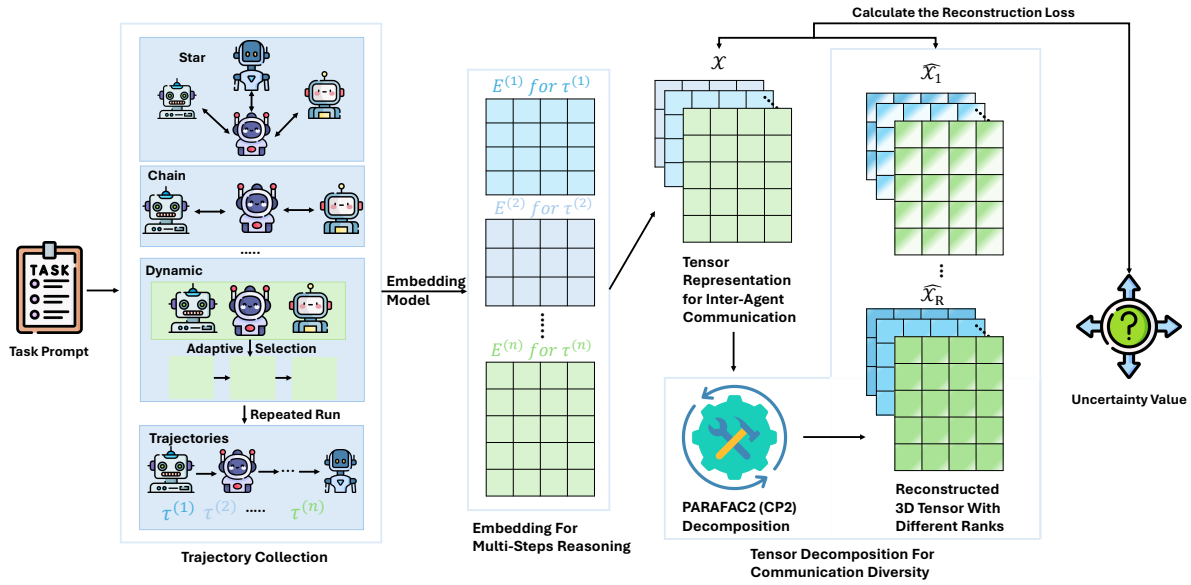


Figure 1: The overall pipeline of MATU. As shown in the figure, MATU could be applied to multi-agent systems with different communication topologies. We first collect trajectories for a fixed system and task, and then obtain embedding matrices for each trajectory. Then, we form a ragged tensor by stacking all embedding matrices and obtain the reconstructed tensor by conducting CP-2 decomposition. Finally, we use the reconstruction losses from reconstructed tensors with different ranks as the final uncertainty.

thereby decoupling semantic comparability from surface form and length. Semantically similar steps, even when expressed with different wording or produced by different agents or tools, are brought closer in the embedding space. Besides, using additional embedding models facilitates step-wise aggregation without requiring token-level probabilities and establishes the foundation for subsequent tensor representations and decomposition.

4.2 Tensor Representation for Inter-agent Communication

The second challenge comes from inter-agent communication. Even when the agent system and the task input are fixed, MAS may produce distinct communication patterns. Agents can exchange information in slightly different orders, generate intermediate responses of different lengths, or invoke tools at different points.

To capture such variability, we represent embedding matrices from repeated trajectories as a ragged tensor (Fegade et al., 2022). In run $j \in \{1, \ldots, N\}$, each agent $k \in \{1, \ldots, K\}$ produces a trajectory of length $T_{j,k}$, which we embed into a matrix $E^{(j,k)} \in \mathbb{R}^{T_{j,k} \times d}$. We define the ragged object as the doubly-indexed matrix collection

$$\mathcal{X} = \{\, E^{(j,k)} \mid j = 1, \ldots, N;\ k = 1, \ldots, K \,\},$$

where $E^{(j,k)}$ denotes the stacked embedding matrix of agent k in run j. Note that this matrix collection is a three-dimensional ragged tensor. Unlike a standard tensor in $\mathbb{R}^{N \times T \times d}$ that assumes a fixed T, the ragged tensor $\mathcal{X}$ allows T_j to vary across runs:

$$E^{(j,k)} \in \mathbb{R}^{T_{j,k} \times d}, \qquad T_{j,k} \neq T_{j',k'} \text{ in general.}$$

This representation enables us to aggregate multi-run trajectories into a single mathematical object without discarding the diversity of communication patterns. The variability of inter-agent communication is thus encoded directly into the structure of $\mathcal{X}$, laying the groundwork for decomposition methods that disentangle and quantify the uncertainty it induces.

4.3 Tensor Decomposition for Communication Diversity

The third challenge arises from communication diversity across different system topologies. Multi-agent systems may be organized in star (Wu et al., 2024), chain (Li et al., 2023), or dynamic communication structures (Wang et al., 2025), and each topology induces distinct statistical properties in the trajectories it generates. An uncertainty quantification framework must therefore be general enough to handle arbitrary topologies while remaining sensitive to their structural differences.

To address this, we apply the PARAFAC2 Decomposition for Ragged Tensors (CP-2), a factorization method specifically designed to handle irregular tensor structures (Schenker et al., 2023; Perros et al., 2017). Unlike classical tensor decomposition, CP-2 operates directly on ragged tensors by aligning latent factors across dimensions of varying lengths. This property makes CP-2 particularly well-suited to our settings with variable lengths.

Formally, CP-2 seeks a low-rank approximation of the ragged tensor $\mathcal{X}$ in the form

$$\mathcal{X} \approx \sum_{r=1}^{R} \lambda_r \, u_r^{(1)} \otimes u_r^{(2)} \otimes u_r^{(3)},$$

where R is the target rank, λ_r are scalar weights, and $u_r^{(1)}, u_r^{(2)}, u_r^{(3)}$ are latent factors that are defined so as to respect the irregular lengths in $\mathcal{X}$. Through this decomposition, CP-2 captures shared patterns across steps, agents, and runs, while preserving the diversity introduced by different communication topologies.

To quantify uncertainty, we perform CP-2 decomposition under different ranks R. For each R, we reconstruct an approximation $\hat{\mathcal{X}}_R$ and compute the reconstruction loss $\mathcal{L}_R = \|\mathcal{X} - \hat{\mathcal{X}}_R\|$.

The sequence of losses $\{\mathcal{L}_R\}$ reflects how compressible the set of trajectories is under low-rank factors. Higher losses indicate that trajectories cannot be explained by a small number of latent components, implying higher uncertainty. To obtain a single scalar score, we aggregate the reconstruction losses across all considered ranks, defining the final uncertainty value as

$$U = \sum_{R=1}^{R_{\max}} \mathcal{L}_R,$$

where $R_{\max}$ denotes the largest rank examined during decomposition. This score summarizes the degree to which variability in trajectories resists compression across different model capacities, and thus serves as the overall uncertainty estimate for the multi-agent system. By grounding the analysis in CP-2 decomposition, our framework can utilize the information from ragged tensors and generalize to arbitrary communication topologies while maintaining good uncertainty quantification.

5 Experiments

We conduct comprehensive experiments to evaluate the effectiveness of MATU. Our study is designed to answer the following research questions:

- **RQ1:** Does MATU provide more accurate uncertainty quantification for multi-agent systems with static design?
- **RQ2:** Does MATU provide more accurate uncertainty quantification for multi-agent systems with dynamic design?
- **RQ3:** Does MATU provide more accurate uncertainty quantification for multi-agent systems with tool integration?

Beyond the research questions, we also provide a detailed case study to show why MATU could work in Appendix C.

5.1 Experimental Setup

Dataset To comprehensively evaluate MATU, we use four diverse datasets: MATH (mathematical reasoning) (Hendrycks et al., 2021), MoreHopQA (multi-hop QA) (Schnitzler et al., 2024), MMLU (general knowledge) (Hendrycks et al., 2020), and HumanEval (code generation) (Chen et al., 2021). Detailed descriptions are provided in Appendix B.1.

Multi-agent System. We use multiple MAS with different designs. In detail, we consider using Camel (Li et al., 2023), which consists of an AI User and an AI Assistant with round-robin conversation, and AutoGen (Wu et al., 2024), which uses a star agent that assigns tasks to all other agents. Both frameworks use static design. On the other hand, we use AnyMac (Wang et al., 2025), which will dynamically choose the next agent based on the progress as the dynamic multi-agent system.

Models For models behind agents, we are using both open-source and closed-source models. For the open-source model, we mainly use Qwen2.5-7B (Bai et al., 2023) and Llama3.1-8B (Dubey et al., 2024), which is the representative open-source model. For closed-source models, we mainly use GPT-4o from OpenAI.

Evaluation Metrics Effective uncertainty measures should correlate with response correctness: higher uncertainty should indicate a higher likelihood of error. Following prior work (Lin et al., 2023; Da et al., 2024), we evaluate uncertainty estimates by using them to predict whether a generated answer is correct. We report Area Under Receiver Operating Characteristic (AUROC) and Area Under Accuracy Rejection Curve (AUARC) as evaluation metrics, where **a higher AUROC or AUARC demonstrates better uncertainty measures**. To compute AUROC and AUARC, the accuracy of each original response is required. To label re-

Table 1: Comparison of our methods with different baselines on various datasets and large language models on Camel. Best performance has been **highlighted**.

Methods	GPT-4o		Qwen2.5-7B		Llama3.1-8B	
	AUROC	AUARC	AUROC	AUARC	AUROC	AUARC
Dataset: MATH						
Eigv(Agre)-final	0.5698	0.5216	0.5238	0.8466	0.5243	0.6170
Eigv(Agre)-Whole	0.5632	0.5218	0.6784	0.8963	0.5622	0.6346
P(true)	0.5825	0.5592	0.6351	0.8855	0.5421	0.6303
SAUP-Single	-	-	0.5597	0.8499	0.5244	0.6374
SAUP-Multiple	-	-	0.6078	0.8722	0.5258	0.6427
MATU	**0.6797**	**0.6160**	**0.7089**	**0.9064**	**0.7354**	**0.7525**
Dataset: MoreHopQA						
Eigv(Agre)-final	0.5307	0.3374	0.5631	0.6529	0.5572	0.5644
Eigv(Agre)-Whole	0.5259	0.3319	0.5420	0.6342	0.5398	0.5585
P(true)	0.5480	0.3405	0.5766	0.6512	0.5313	0.5460
SAUP-Single	-	-	0.5103	0.6211	0.5083	0.5576
SAUP-Multiple	-	-	0.5386	0.6345	0.5668	0.5798
MATU	**0.5555**	**0.3474**	**0.6529**	**0.7226**	**0.6320**	**0.6561**
Dataset: MMLU						
Eigv(Agre)-final	0.5365	0.3304	0.5537	0.8023	0.5161	0.7270
Eigv(Agre)-Whole	0.5341	0.3236	0.5420	0.7995	0.5940	0.7646
P(true)	0.5059	0.3183	0.6846	0.8585	0.6207	0.7964
SAUP-Single	-	-	0.5233	0.7749	0.5424	0.7361
SAUP-Multiple	-	-	0.5641	0.8100	0.5289	0.7330
MATU	**0.5604**	**0.3384**	**0.7149**	**0.8656**	**0.7075**	**0.8427**

Table 2: Comparison of our methods with different baselines on various datasets and large language models on AutoGen. Best performance has been **highlighted**.

Methods	GPT-4o		Qwen2.5-7B		Llama3.1-8B	
	AUROC	AUARC	AUROC	AUARC	AUROC	AUARC
Dataset: MATH						
Eigv(Agre)-final	0.5898	0.5826	0.6355	0.4512	0.5912	0.3802
Eigv(Agre)-Whole	0.6015	0.5892	0.6111	0.4326	0.5761	0.3679
P(true)	0.6079	0.5931	0.6524	0.5102	0.6271	0.4571
SAUP-Single	-	-	0.5268	0.3990	0.6064	0.3830
SAUP-Multiple	-	-	0.5385	0.4090	0.6334	0.3933
MATU	**0.6582**	**0.6220**	**0.7146**	**0.5334**	**0.7544**	**0.4687**
Dataset: MoreHopQA						
Eigv(Agre)-final	0.5311	0.4968	0.5331	0.6678	0.5395	0.5721
Eigv(Agre)-Whole	0.5218	0.4942	0.5323	0.6689	0.5279	0.5642
P(true)	0.5598	0.5033	0.5806	0.7031	0.5515	0.5827
SAUP-Single	-	-	0.5197	0.6445	0.5422	0.5782
SAUP-Multiple	-	-	0.5342	0.6708	0.5488	0.5877
MATU	**0.5817**	**0.5237**	0.6392	0.7374	0.5989	0.6117
Dataset:MMLU						
Eigv(Agre)-final	0.5981	0.5649	0.7105	0.8617	0.5521	0.4288
Eigv(Agre)-Whole	0.5759	0.5438	0.6867	0.8516	0.5316	0.3762
P(true)	0.5802	0.5528	0.6556	0.8363	0.5775	0.4368
SAUP-Single	-	-	0.6484	0.8552	0.5138	0.3031
SAUP-Multiple	-	-	0.7193	0.8589	0.5018	0.2973
MATU	**0.6277**	**0.5841**	**0.7315**	**0.8833**	**0.5954**	**0.4745**

sponses as correct or incorrect, we use a reference LLM, GPT-5, to provide correctness scores to the final answer from MAS.

Baseline We compare MATU against three baselines: P(true) (Kadavath et al., 2022), Eigv(Agr) (Lin et al., 2023), and SAUP (Zhao et al., 2024). For the Eigv(Agr), we use the final answer or every conversation to compute the entailment matrix (Bowman et al., 2015), resulting in two different variants: Eigv(Agr)-Answer and Eigv(Agr)-Whole. SAUP is originally designed for one trajectory, while we collect multiple trajectories. Therefore, we use SAUP-Single, which uses the SAUP from the first trajectory, and SAUP-Multiple which uses the mean SAUP from all trajectories. Please note that SAUP is a **white-box** method so that it cannot be applied to closed-source models. More introduction can be found at Appendix B.2.

Implementation Detail For the embedding models, we use open-source Qwen3-embedding-0.6B to get the fast processing speed. For trajectories, we collect 10 trajectories for every task, and we use a temperature of 0.9 for every setting. All the experiments are conducted on a single Nvidia A100-80GB GPU or using an OpenAI API.

5.2 Performance for Multi-agent System with Static Design (RQ1)

Firstly, to explore how good MATU is for MAS with static design, we conduct experiments on Camel (Li et al., 2023) and AutoGen (Wu et al., 2024) and three different datasets to demonstrate the performance comprehensively. The results are shown in Table 1 and Table 2. The results show that:

• MATU consistently outperforms all baselines by capturing holistic system-level behavior rather than just final output consistency. While traditional methods like Eigv(Agre) focus on semantic similarity and SAUP measures step-wise uncertainty independently, MATU integrates the entire reasoning trajectory and multi-run communication patterns into a unified tensor. This approach allows it to identify fragile consensus in the collaborative process that response-level or single-trajectory measures fail to detect.

• MATU shows consistent reliability whether the task involves challenging mathematical reasoning in MATH, multi-hop question-answering in MoreHopQA, or broad knowledge synthesis in MMLU. By mapping heterogeneous outputs into a shared embedding space, the method provides a robust reliability measure that remains effective regardless of whether the MAS is performing logical deduction or knowledge retrieval.

5.3 Performance for Multi-agent System with Dynamic Design (RQ2)

To evaluate the performance of MATU in more complex, adaptive environments, we extend our evaluation to multi-agent systems with dynamic designs.

Table 3: Comparison of our methods with different baselines on various datasets and large language models on AnyMac.

Methods	GPT-4o		Qwen2.5-7B		Llama3.1-8B	
	AUROC	AUARC	AUROC	AUARC	AUROC	AUARC
Dataset: MATH						
Eigv(Agre)-final	0.6359	0.6133	0.6506	0.8212	0.6340	0.6059
Eigv(Agre)-Whole	0.6308	0.6115	0.6314	0.8081	0.6215	0.5953
P(true)	0.6226	0.6070	0.6602	0.8291	0.6581	0.6225
SAUP-Single	-	-	0.6261	0.7982	0.6339	0.6008
SAUP-Multiple	-	-	0.6396	0.8119	0.6477	0.6065
MATU	**0.6675**	**0.6439**	**0.6966**	**0.8585**	**0.7121**	**0.6518**
Dataset: MorehopQA						
Eigv(Agre)-final	0.5257	0.3992	0.6079	0.6741	0.6110	0.6369
Eigv(Agre)-Whole	0.5203	0.4010	0.6021	0.6681	0.6034	0.6300
P(true)	0.5455	0.4121	0.6205	0.6853	0.6158	0.6416
SAUP-Single	-	-	0.6088	0.6770	0.5918	0.6277
SAUP-Multiple	-	-	0.6242	0.6914	0.6055	0.6322
MATU	**0.5671**	**0.4336**	**0.6457**	**0.7029**	**0.6262**	**0.6493**
Dataset: MMLU						
Eigv(Agre)-final	0.5568	0.4952	0.5446	0.7650	0.5321	0.6586
Eigv(Agre)-Whole	0.5641	0.5049	0.5337	0.7433	0.5215	0.6512
P(true)	0.5594	0.4976	0.5552	0.7681	0.5297	0.6632
SAUP-Single	-	-	0.5048	0.7261	0.5340	0.6542
SAUP-Multiple	-	-	0.5382	0.7602	0.5382	0.6719
MATU	**0.5925**	**0.5152**	**0.5821**	**0.7768**	**0.5500**	**0.6797**

Unlike static topologies, dynamic systems such as AnyMac (Wang et al., 2025) adaptively select the next agent during execution based on the evolving context of the task. We conduct these experiments across the same datasets using both open-source models and closed-source architectures (GPT-4o). The results for the AnyMac system are detailed in Table 3, leading to the following observations:

• MATU demonstrates superior adaptability to unpredictable communication sequences by leveraging higher-order tensor representations. In dynamic systems where the sequence of agent interactions varies significantly between runs, traditional semantic or step-wise baselines struggle to maintain a consistent reliability measure. By organizing these varied trajectories into a ragged tensor and applying tensor decomposition, MATU successfully aligns latent factors across dimensions of varying lengths, allowing it to outperform the strongest baselines by a significant margin in AUROC and AUARC.

• Comparative analysis of self-evaluation and propagation baselines underscores the necessity of multi-run structural ensembling. Considering all experimental results and the baselines, $P(true)$ emerges as the most competitive, likely because it leverages the LLM's intrinsic ability to reflect on its own non-linear reasoning process. Furthermore, the consistent superiority of SAUPMultiple over SAUPSingle confirms that a single interaction path is a poor proxy for the system's overall reliability in a dynamic environment.

5.4 Performance for Multi-agent System with Tool Integration (RQ3)

To explore whether MATU can effectively quantify the reliability of collaborative agents when integrated with external tools, we conduct experiments on the HumanEval benchmark. This task requires agents not only to reason linguistically but also to synthesize executable code and interact with a Python interpreter, which serves as a functional tool within the multi-agent workflow. We conduct the experiments on llama3, and the results can be found at Fig. 3. The results show that MATU outperforms other baselines on both AUROC and AUARC with the Humaneval dataset, showing the robustness of MATU with code environment and tool integration.

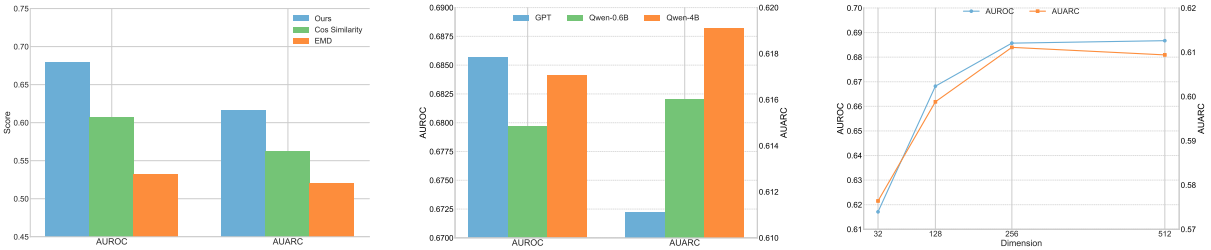
5.5 Ablation and Sensitivity Study

To further analyze the robustness and key components of our framework, we conduct a series of ablation and sensitivity experiments on the Camel and the MATH dataset with GPT-4o.

Ablation with Input Variants To verify whether raw embedding tensors are superior to traditional distance-based representations, we compare MATU against variants that use Earth Mover's Distance (EMD) and Cosine Similarity to construct the similarity matrices for decomposition instead of the step-level embedding matrices. As shown in Fig. 2a, MATU consistently yields higher AUROC and AUARC scores, while EMD and Cosine Similarity fail to capture the granular latent signals within agent trajectories. This confirms that applying tensor decomposition directly to reasoning embeddings preserves significantly richer multi-agent dynamics than distance-based metrics.

Impact of Embedding Models We examine how the choice of the underlying text embedding model affects the precision of uncertainty estimation. We evaluate three models of varying scales: GPT-Embedding, Qwen-0.6B-Embedding, and Qwen-4B-Embedding. The results are shown in Fig. 2b. While larger models like Qwen-4B and GPT-Embedding provide slight performance gains, the difference compared to the Qwen-0.6B model is minimal. We conclude that Qwen-0.6B offers the optimal balance between computational efficiency and accuracy, making it sufficient for UQ.

Sensitivity to Embedding Dimensions To determine the optimal latent space dimensionality for representing complex reasoning steps, we evaluate



(a) Ablation Study (b) Embedding Models (c) Effect of Embedding Dimension

Figure 2: Results for ablation study and sensitivity study. The results show that our design for MATU and our choices of the hyperparameter are well-suited.

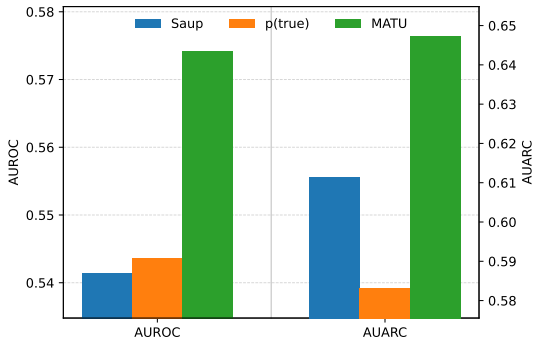


Figure 3: Comparison of MATU and baselines on llama3 and the Humaneval dataset. The results show that MATU can have better results even with tool integration, showing the robustness of MATU.

the system's performance across dimensions ranging from 32 to 512. The results in Fig. 2b indicate a sharp improvement in both AUROC and AUARC as the dimension increases to 256, after which the gains become marginal. Consequently, we select 256 as our default embedding dimension to ensure comprehensive representation without incurring redundant computational overhead.

5.6 Down-stream Task

To evaluate the practical utility of MATU in real-world deployment, we conduct a backbone selection task. This experiment explores whether uncertainty scores can serve as a reliable signal to select the most accurate answer from a pool of different MAS configurations. Specifically, for a given query, we generate multiple potential solutions across four distinct LLM backbones: Qwen2.5-7B (Yang et al., 2024b), Llama3.1-8B (Dubey et al., 2024), Qwen3-4B (Yang et al., 2025), and Gemma3-4B (Kamath et al., 2025). For each query, the system identifies the backbone that yields the lowest uncertainty score U and selects its response as the final output. We evaluate this routing strategy by comparing the resulting system accuracy when guided by MATU against selection based on $P(true)$ 4, $SAUP$-Multiple, and a random selection baseline on the Camel framework and the MATH dataset. The results are shown in Fig. 4. The results show the superior performance improvement using MATU, showing that MATU is a robust tool for backbone selection, which indicates that MATU offers a robust uncertainty value.

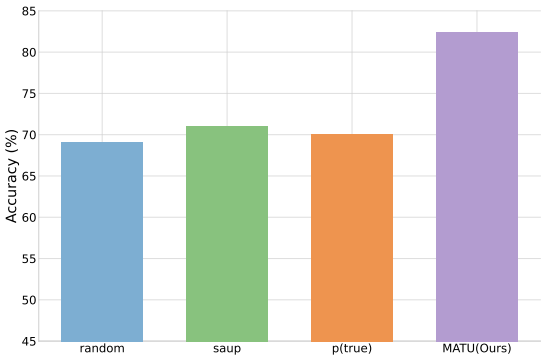


Figure 4: Comparison of backbone selection results. A higher accuracy demonstrates a better selection strategy. The results show that MATU has a superior performance improvement on accuracy, indicating that MATU offers a more robust uncertainty value.

6 Conclusion

In this work, we propose MATU, a pioneering framework for quantifying uncertainty in LLM-based multi-agent systems by leveraging tensor decomposition to capture the holistic dynamics of multi-step reasoning and inter-agent communication. By organizing reasoning trajectories into ragged tensors and analyzing them via PARAFAC2 decomposition, our method effectively disentangles sources of uncertainty across varying communication topologies and run lengths, overcoming the limitations of traditional semantic or step-wise approaches. Extensive experiments on diverse benchmarks demonstrate that MATU consistently outperforms existing baselines in both static and dynamic system designs, while also proving its practicality in down-stream tasks such as backbone model selection.

Limitations

While MATU demonstrates effectiveness in quantifying uncertainty for multi-agent systems, we acknowledge several limitations in our current work. First, the core mechanism relies on constructing a higher-order tensor from multiple reasoning trajectories (e.g., $N = 10$ runs in our experiments), meaning the inference cost scales linearly with the number of sampled trajectories. Although this multi-run paradigm is standard in black-box uncertainty estimation like Self-Consistency, it inevitably consumes more computational resources compared to single-pass methods. Second, since MATU decouples semantic meaning from surface form by mapping reasoning steps into a latent space, the sensitivity and accuracy of our uncertainty quantification are bounded by the quality of the underlying embedding model. In highly specialized domains where general-purpose embedding models may fail to capture subtle semantic nuances, MATU's performance might degrade unless domain-specific embeddings are employed.

References

Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. 2020. Deep evidential regression. *Advances in neural information processing systems*, 33:14927–14937.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Ferhat Ozgur Catak and Murat Kuzlu. 2024. Uncertainty quantification in large language models through convex hull analysis. *Discover Artificial Intelligence*, 4(1):90.

Jiuhai Chen and Jonas Mueller. 2024. Quantifying uncertainty in answers from any language model and enhancing their trustworthiness. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5186–5200.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Tiejin Chen, Xiaoou Liu, Longchao Da, Jia Chen, Vagelis Papalexakis, and Hua Wei. 2025. Uncertainty quantification of large language models through multi-dimensional responses. *arXiv preprint arXiv:2502.16820*.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. 2022. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*.

Longchao Da, Tiejin Chen, Lu Cheng, and Hua Wei. 2024. Llm uncertainty quantification through directional entailment graph and claim level response augmentation. *arXiv preprint arXiv:2407.00994*.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Pratik Fegade, Tianqi Chen, Phillip Gibbons, and Todd Mowry. 2022. The cora tensor compiler: Compilation for ragged tensors with minimal padding. *Proceedings of Machine Learning and Systems*, 4:721–747.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR.

Xiang Gao, Jiaxin Zhang, Lalla Mouatadid, and Kamalika Das. 2024. SPUQ: Perturbation-based uncertainty quantification for large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2336–2346, St. Julian's, Malta. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Samuel Holt, Max Ruiz Luyten, and Mihaela van der Schaar. 2023. L2mac: Large language model automatic computer for extensive code generation. *arXiv preprint arXiv:2310.02003*.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, and 1 others. 2023. Metagpt: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*.

Bairu Hou, Yujian Liu, Kaizhi Qian, Jacob Andreas, Shiyu Chang, and Yang Zhang. 2024. Decomposing uncertainty for large language models through input clarification ensembling. In *International Conference on Machine Learning*, pages 19023–19042. PMLR.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Gemma Team Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ram'e, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean-Bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gael Liu, and 191 others. 2025. Gemma 3 technical report. *ArXiv*, abs/2503.19786.

Yubin Kim, Chanwoo Park, Hyewon Jeong, Yik S Chan, Xuhai Xu, Daniel McDuff, Hyeonhoon Lee, Marzyeh Ghassemi, Cynthia Breazeal, and Hae W Park. 2024. Mdagents: An adaptive collaboration of llms for medical decision-making. *Advances in Neural Information Processing Systems*, 37:79410–79452.

Michael Kirchhof, Gjergji Kasneci, and Enkelejda Kasneci. 2025. Position: Uncertainty quantification needs reassessment for large-language model agents. *arXiv preprint arXiv:2505.22655*.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.

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

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2023. Generating with confidence: Uncertainty quantification for black-box large language models. *arXiv preprint arXiv:2305.19187*.

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

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. 2024. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*.

Bill MacCartney. 2009. *Natural language inference*. Stanford University.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, and 1 others. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. 2019. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22.

Ioakeim Perros, Evangelos E Papalexakis, Fei Wang, Richard Vuduc, Elizabeth Searles, Michael Thompson, and Jimeng Sun. 2017. Spartan: Scalable parafac2 for large & sparse data. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 375–384.

Chen Qian, Xin Cong, Cheng Yang, Weize Chen, Yusheng Su, Juyuan Xu, Zhiyuan Liu, and Maosong Sun. 2023. Communicative agents for software development. *arXiv preprint arXiv:2307.07924*, 6(3):1.

Carla Schenker, Xiulin Wang, and Evrim Acar. 2023. Parafac2-based coupled matrix and tensor factorizations. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Julian Schnitzler, Xanh Ho, Jiahao Huang, Florian Boudin, Saku Sugawara, and Akiko Aizawa. 2024. Morehopqa: More than multi-hop reasoning. *ArXiv*, abs/2406.13397.

Murat Sensoy, Lance Kaplan, and Melih Kandemir. 2018. Evidential deep learning to quantify classification uncertainty. *Advances in neural information processing systems*, 31.

João Vitor de Carvalho Silva and Douglas G Macharet. 2025. Can llm agents solve collaborative tasks? a study on urgency-aware planning and coordination. *arXiv preprint arXiv:2508.14635*.

Xiangru Tang, Anni Zou, Zhuosheng Zhang, Ziming Li, Yilun Zhao, Xingyao Zhang, Arman Cohan, and Mark Gerstein. 2023. Medagents: Large language models as collaborators for zero-shot medical reasoning. *arXiv preprint arXiv:2311.10537*.

Song Wang, Zhen Tan, Zihan Chen, Shuang Zhou, Tianlong Chen, and Jundong Li. 2025. Anymac: Cascading flexible multi-agent collaboration via next-agent prediction. *arXiv preprint arXiv:2506.17784*.

Yue Wang, Weishi Wang, Shafiq Joty, and Steven CH Hoi. 2021. Codet5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. *arXiv preprint arXiv:2109.00859*.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, and 1 others. 2024. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *First Conference on Language Modeling*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *ArXiv*, abs/2505.09388.

John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. 2024a. Swe-agent: Agent-computer interfaces enable automated software engineering. *Advances in Neural Information Processing Systems*, 37:50528–50652.

Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, and 25 others. 2024b. Qwen2.5 technical report. *ArXiv*, abs/2412.15115.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Kai Ye, Tiejin Chen, Hua Wei, and Liang Zhan. 2024. Uncertainty regularized evidential regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 16460–16468.

Guibin Zhang, Yanwei Yue, Xiangguo Sun, Guancheng Wan, Miao Yu, Junfeng Fang, Kun Wang, Tianlong Chen, and Dawei Cheng. 2024. G-designer: Architecting multi-agent communication topologies via graph neural networks. *arXiv preprint arXiv:2410.11782*.

Qiwei Zhao, Dong Li, Yanchi Liu, Wei Cheng, Yiyou Sun, Mika Oishi, Takao Osaki, Katsushi Matsuda, Huaxiu Yao, Chen Zhao, and 1 others. 2025. Uncertainty propagation on llm agent. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6064–6073.

Qiwei Zhao, Xujiang Zhao, Yanchi Liu, Wei Cheng, Yiyou Sun, Mika Oishi, Takao Osaki, Katsushi Matsuda, Huaxiu Yao, and Haifeng Chen. 2024. Saup: Situation awareness uncertainty propagation on llm agent. *arXiv preprint arXiv:2412.01033*.

Zihao Zhou, Bin Hu, Chenyang Zhao, Pu Zhang, and Bin Liu. 2023. Large language model as a policy teacher for training reinforcement learning agents. *arXiv preprint arXiv:2311.13373*.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. 2024. Gptswarm: Language agents as optimizable graphs. In *Forty-first International Conference on Machine Learning*.

A Code

Sample code can be found at https://anonymous.4open.science/r/MATU-5864/README.md.

B Detailed Experimental Settings

B.1 Detailed Introduction to Datasets

- MATH (Hendrycks et al., 2021): A dataset for **mathematical reasoning** that consists of challenging competition-level problems across algebra, geometry and number theory.
- MoreHopQA (Schnitzler et al., 2024): A widely used question-answering dataset requiring **multi-hop text reasoning** over Wikipedia passages.
- MMLU (Hendrycks et al., 2020): The Massive Multitask Language Understanding benchmark, covering 57 subjects. It assesses broad knowledge and problem-solving abilities, making it a strong indicator of **general-domain reasoning**. To avoid the overlap between the MATH dataset, when using MMLU, we exclude subjects about math.
- HumanEval (Chen et al., 2021): A code generation benchmark consisting of programming problems with unit tests. Models are required to synthesize correct and executable code solutions, and we provide a code environment for all multi-agent systems as a tool integration.

B.2 Detailed Introduction to Baselines

As far as we know, we are the first method that targets the uncertainty quantification for MAS. To compare our method, we mainly adopt the existing methods for LLM or single-agent to multi-agent settings. In detail, we consider using Eigv(Agr) (Lin et al., 2023), which is the sum of eigenvalues for graph normalized Laplacian matrix and the graph is formed by the entailment matrix (Bowman et al., 2015) and P(true) (Kadavath et al., 2022), which obtains the uncertainty by directly asking the LLM itself. For the Eigv(Agr), we use the final answer or every conversation to compute the entailment matrix, resulting in two different variants: Eigv(Agr)-answer and Eigv(Agr)-whole. Besides, we also use SAUP (Zhao et al., 2024), which is a white-box UQ method for a single agent by calculating the weighted sum of entropy for each step. We will treat the step from a different agent as each step in SAUP to transfer SAUP to a multi-agent setting. SAUP is originally designed for one trajectory, while we collect multiple trajectories. Therefore, we use SAUP-Single which uses the SAUP from the first trajectory, and SAUP-Multiple that uses the mean SAUP from all trajectories.

C Case Study

To qualitatively demonstrate the robustness of MATU against the structural variability of multi-agent interactions, we analyze a representative example from the MATH dataset with qwen2.5, as illustrated in Table 4. In detail, we have:
Question: "What is the distance between the two intersections of $y = x^2$ and $x + y = 1$?"
Ground Truth: "$\sqrt{10}$"

In this experiment, we collected 10 independent reasoning trajectories. The multi-agent system demonstrated perfect performance, achieving a 100% accuracy rate by deriving the correct answer $\sqrt{10}$ in all runs. However, the trajectories exhibited significant diversity in their communication patterns. While some runs produced concise and direct derivations (Type A), others involved self-correction mechanisms where agents identified and fixed calculation errors (Type B), or contained heavy steps (Type C), resulting in varying trajectory lengths with similar core logic when solving the problem. We report the normalized uncertainty values for all methods so that we might compare the uncertainty directly.

Analysis of Baselines. Despite the consistency in the final outcome, baseline methods failed to accurately reflect the system's reliability. SAUP assigned a misleadingly high uncertainty score of 0.88. This false positive occurs because SAUP calculates uncertainty by accumulating entropy step-by-step. The heavy-step trajectories (Type C), despite being logically sound, contained more intermediate steps, which artificially inflated the cumulative entropy. Consequently, SAUP misinterpreted the surface-level verbosity, which is a byproduct of the communication topology, as semantic instability. Similarly, Eigv-Whole yielded a moderate uncertainty score of 0.35. This suggests that the NLI models used for entailment checking struggled to handle the long contexts and the noise introduced by self-correction steps, failing to fully recognize the logical entailment between the diverse reasoning paths.

Case Info	Agent Reasoning Trajectories (Key Steps Only)	Uncertainty Quantification
Problem: Find the distance between intersections of $y = x^2$ and $x + y = 1$. **True Answer:** $\sqrt{10}$ **System Accuracy:** 100% (10/10 runs correct)	**Trajectory Type A: Direct & Concise** • Determine intersection coordinates → Calculate distance *Assessment: Ideal path, minimal token generation.*	**SAUP (Baseline):** 0.88 (High) *Issue: High cumulative entropy from verbose steps.*
	Trajectory Type B: Self-Correction • Determine coordinates → Correct y-axis calculation error → Get correct coordinates → Calculate distance *Assessment: Agent successfully recovers from an error.*	**Eigv-Whole (Baseline):** 0.35 (Medium) *Issue: Long contexts dilute NLI entailment accuracy.*
	Trajectory Type C: Verbose (High Step Count) • Determine coordinates → Get coordinates → . . . *(intermediate steps)* . . . → Calculate distance *Assessment: Logically identical to Type A, but higher step count increases cumulative entropy.*	**MATU (Ours):** 0.05 (Low) *Result: Correctly aligns semantic intent across diverse paths.*

Table 4: **Case Study on Mathematical Reasoning.** Despite diverse communication patterns, all agents consistently reach the correct solution ($\sqrt{10}$). Baselines like SAUP fail due to sensitivity to trajectory length (step count), and Eigv-Whole struggles with long-context entailment. MATU effectively disentangles surface-level variations from semantic stability, correctly assigning low uncertainty.

Analysis of MATU. In contrast, **MATU** correctly quantified the system's high reliability with a low uncertainty score of 0.05. By leveraging tensor decomposition on the reasoning embeddings, MATU effectively disentangles surface-level variations from the underlying semantic content. The tensor structure allows our method to align latent factors across trajectories of different lengths, recognizing that the corrective steps in Type B and the verbose explanations in Type C semantically converge to the same reasoning path as the concise Type A. This case highlights MATU's unique ability to filter out the noise caused by communication diversity, providing a more robust and holistic uncertainty measure for multi-agent systems.