PropUQ-MAS: Propagation-Aware Uncertainty Quantification for LLM Multi-Agent Systems

Yaokun Liu[1*] **Yifan Liu**[1*] **Daniel Yue Zhang**[2] **Ruichen Yao**[1]
Zelin Li[1] **Dong Wang**[1]
[1]University of Illinois Urbana-Champaign
[2]Scale AI
{yaokunl2, yifan40, ryao8, zelin3, dwang24}@illinois.edu
yue.zhang@scale.com

Abstract

LLM-based multi-agent systems (MAS) solve complex tasks through communication among role-specialized agents. However, inter-agent dependencies introduce reliability risks beyond isolated agent failures. For instance, errors in intermediate messages could be inherited and amplified by downstream agents. Existing uncertainty quantification (UQ) methods mainly target isolated responses or single-agent reasoning, and therefore fail to capture uncertainty propagation in MAS. To this end, we propose PROPUQ-MAS, an error propagation-aware UQ framework that represents MAS execution as a communication-structured graph and estimates each step's reliability by combining local uncertainty with uncertainty inherited from upstream messages. Extensive experiments demonstrate that PROPUQ-MAS consistently improves UQ in MAS, with average relative gains of $+6.10\%$ in AUROC and $+47.58\%$ in PRR.[1]

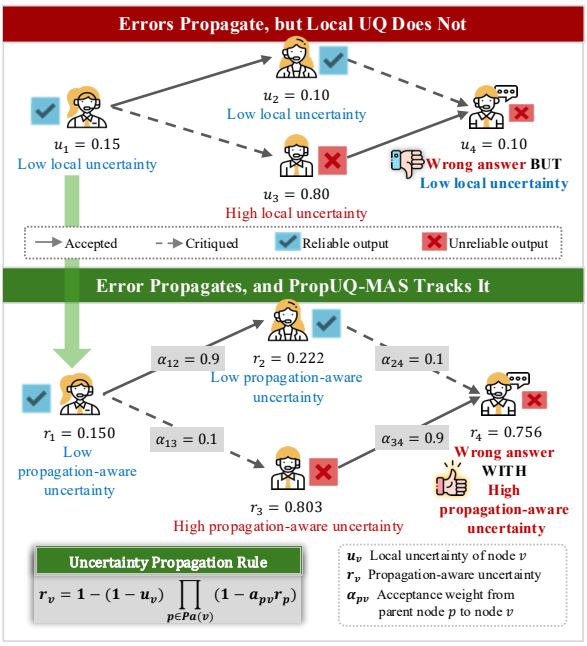

Figure 1: Local UQ misses propagated risk, while PROPUQ-MAS tracks propagation-aware uncertainty over the MAS execution graph.

1 Introduction

Large language model (LLM)-based multi-agent systems (MAS) have emerged as a powerful paradigm for solving complex tasks by orchestrating role-specialized agents across multi-step workflows (Wu et al., 2024; Li et al., 2023; Hong et al., 2024). Through inter-agent communication and strategic collaboration, MAS exhibit superior flexibility in demanding domains such as autonomous software engineering and multi-step web navigation (Liu et al., 2024). However, the same inter-agent dependencies that enable such flexibility can also shift the reliability bottleneck from individual agent failures to system-wide error propagation.

Unlike isolated agents whose errors stem primarily from intrinsic limitations (e.g., hallucinations), MAS reliability is a collective property governed by interaction dynamics: An erroneous intermediate message from an upstream agent may be interpreted as a valid context by downstream agents, incorporated into subsequent reasoning, and propagated or even amplified through inter-agent communication (Cemri et al., 2026; Hu et al., 2025).

To mitigate the cascading risks, we argue that MAS requires a step-wise propagation-aware uncertainty quantification (UQ) framework, which estimates the marginal failure risks of each intermediate output during the MAS execution trajectory. Such a framework can serve as a diagnostic signal for MAS by supporting real-time monitoring of risky intermediate states, adaptive intervention before errors accumulate, and post-hoc attribution of failure-contributing steps (Shinn et al., 2023; Zhang et al., 2025). These target applications re-

[*]Equal contribution.
[1]Code is released at https://github.com/yaokunliu/PropUQ-MAS.git.

quire the UQ framework to be topology-agnostic and online-computable: it should handle diverse MAS structures and update uncertainty as each intermediate output is generated, without waiting for the full trajectory to complete.

However, existing UQ frameworks are not designed to track step-wise uncertainty propagation in MAS. As illustrated in Figure 1, traditional LLM UQ methods estimate local uncertainty for isolated single-turn outputs using signals such as token probabilities (Fomicheva et al., 2020; Fadeeva et al., 2024), self-evaluation (Tian et al., 2023; Kadavath et al., 2022), or semantic consistency (Manakul et al., 2023; Kuhn et al., 2023). Recent agentic UQ methods extend local uncertainty estimation from single responses to multi-step reasoning trajectories (Zhao et al., 2025b; Duan et al., 2025). These methods assume a sequential single-agent trajectory, whereas MAS introduces more diverse communication topologies. These differences introduce MAS-specific UQ challenges that are not addressed by agentic UQ, yet MAS-specific UQ remains largely under-explored. More recently, MATU (Chen et al., 2026) estimates trajectory-level uncertainty by the consistency of multiple sampled executions. This post-mortem design neither models uncertainty propagation nor supports real-time node-wise UQ for online monitoring.

To this end, we propose PROPUQ-MAS, a propagation-aware MAS UQ framework for step-wise reliability estimation. By unfolding MAS trajectories into a directed acyclic graph, PROPUQ-MAS bypasses structural constraints to deliver a topology-agnostic framework. Building on this graph view, we derive a recursive propagation rule with a probabilistic interpretation: each node's uncertainty is updated by combining its local uncertainty with propagated uncertainty inherited from upstream agents. The inherited component explicitly modulates the uncertainty propagation strength over communication channels, formalizing the likelihood of error contamination across different agent behaviors, including acceptance and critique. This establishes a tractable propagation rule that quantifies uncertainty seamlessly in a single forward pass. Extensive experiments demonstrate that PROPUQ-MAS consistently improves step-wise UQ in MAS, with average relative gains of $+6.10\%$ in AUROC and $+47.58\%$ in PRR, while showing strong generalization and interpretability.

2 Related Work

2.1 LLM Uncertainty Quantification

Existing LLM UQ methods estimate the reliability of an individual generation using token-level likelihoods (Fomicheva et al., 2020; Fadeeva et al., 2024; Vashurin et al., 2025), verbalized confidence or self-evaluation (Kadavath et al., 2022; Tian et al., 2023; Xiong et al., 2024), and sampling-based semantic consistency (Kuhn et al., 2023; Manakul et al., 2023). These methods provide useful local reliability signals, which we use as the local uncertainty of each node. However, they treat each generation as an isolated prediction target and do not model how errors can be inherited through later interactions. As a result, a downstream output may appear locally confident while still being unreliable due to a contaminated upstream context.

Recent agentic UQ methods further extend local uncertainty estimation to multi-step single-agent trajectories, either by learning situation-aware step weights or decomposing sequential decision uncertainty (Zhao et al., 2025b; Duan et al., 2025). Although these methods model uncertainty over sequential trajectories, they cannot directly transfer to complex MAS topologies, such as hierarchical structures with multi-parent or multi-recipient communication. In contrast, PROPUQ-MAS models uncertainty propagation over execution graphs and applies to arbitrary interaction topologies.

2.2 MAS Uncertainty Quantification

Recent work shows that MAS reliability depends on communication dynamics (Cemri et al., 2026; Hu et al., 2025; Zhang et al., 2025; Kirchhof et al., 2025), motivating step-wise reliability monitoring. However, MAS-level UQ remains largely under-explored, and existing studies do not explicitly model how uncertainty propagates across agents during execution. MATU estimates trajectory-level uncertainty by measuring consistency across multiple sampled MAS executions (Chen et al., 2026). While useful for holistic reliability assessment, such a post-hoc design requires repeated executions and does not explicitly model how uncertainty is transmitted through inter-agent communication, making it less suitable for real-time node-wise monitoring. In contrast, PROPUQ-MAS is a real-time, training-free propagation layer that computes node-wise UQ in a single MAS forward pass, enabling online monitoring of intermediate outputs.

3 Methodology

3.1 Problem Setup

The execution trajectory of a multi-agent system with arbitrary topology can be represented as a directed acyclic execution graph (DAG) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Each node $v = (i, t) \in \mathcal{V}$ denotes an output (e.g., an intermediate message, tool-use result, or final response) produced by agent i at step t. Each directed edge $(p, v) \in \mathcal{E}$ indicates that the generation of node v conditions on the information from node p. The parent set of v is denoted as $\mathrm{Pa}(v) = \{p : (p, v) \in \mathcal{E}\}$. For MAS with feedback or repeated communication, we unroll the observed execution in temporal order: each new agent output becomes a separate node, and edges point from earlier outputs used as context to the later output they condition. Thus, even recurrent communication patterns form a DAG over the realized execution trace. Unlike a single-agent trajectory, which usually follows one temporal chain, an MAS execution graph allows a node to have multiple parents when an agent conditions on several upstream messages.

In this paper, our goal is to quantify propagation-aware uncertainty as a reliability signal for each intermediate output in the MAS execution graph. For each node v, let $E_v \in \{0, 1\}$ denote its error event, where $E_v = 1$ indicates the output is incorrect. The propagation-aware uncertainty of node v is then defined as the marginal error probability:

$$r_v := \Pr(E_v = 1), \quad \forall v \in \mathcal{V}.$$

3.2 Uncertainty Propagation in MAS

In MAS execution, node uncertainty does not arise only from local agent generation, but can also be inherited from upstream agents through inter-agent communication. We therefore focus on the propagated component of node-wise uncertainty, which depends on the accumulated uncertainties of parent nodes as a recursive mapping $\mathcal{F}$:

$$r_v = \mathcal{F}\left(\{r_p\}_{p \in \mathrm{Pa}(v)}\right),$$

where r_p represents the accumulated uncertainty from parent node p.

3.2.1 Error Event Decomposition

To instantiate the recursive mapping $\mathcal{F}$, we introduce an event-level decomposition over the execution graph $\mathcal{G}$, separating errors caused by local agent generation from errors transmitted through incoming interactions. For each node v, we decouple its uncertainty into a local component and a cascading component inherited from parent nodes through inter-agent interactions.

Local Errors. The local error event is denoted by $I_v \in \{0, 1\}$, where $I_v = 1$ represents that node v produces an incorrect output by the inherent reasoning or generation flaws of agent i at step t, isolated from the correctness of its parent inputs. We define the local uncertainty of node v as:

$$u_v := \Pr\left(I_v = 1\right),$$

where $u_v \in [0, 1]$ can be estimated by off-the-shelf single-agent uncertainty quantification methods, including verbalized self-evaluation (Tian et al., 2023; Kadavath et al., 2022), token-level predictive probabilities (Fomicheva et al., 2020; Fadeeva et al., 2024), and sampling-based self-consistency (Manakul et al., 2023; Kuhn et al., 2023).

Propagated Errors. To formalize error propagation in an MAS, we treat each edge (p, v) as a potential channel of error transmission. For each edge $(p, v) \in \mathcal{E}$, node v processes information from parent node p in one of two mutually exclusive modes: acceptance (A) or critique (C). Let $M_{pv} \in \{\mathrm{A}, \mathrm{C}\}$ denote this edge-level mode. Conditioned on parent error $E_p = 1$, we define:

$$\begin{aligned} \alpha_{pv} &:= \Pr(M_{pv} = \mathrm{A} \mid E_p = 1), \\ \beta_{pv} &:= \Pr(M_{pv} = \mathrm{C} \mid E_p = 1), \\ &\text{s.t. } \alpha_{pv} + \beta_{pv} = 1, \end{aligned}$$

where $\alpha_{pv}, \beta_{pv} \in [0, 1]$ denote the probabilities that node v accepts or critiques an erroneous output from parent node p, respectively. In online execution, the true error-conditioned parameter α_{pv} is not directly observable as E_p is unknown. We therefore instantiate α_{pv} with a self-reported acceptance score $\hat{\alpha}_{pv}$, which serves as a plug-in estimate of the edge-level transmission strength. More implementation details are provided in Appendix A.

Based on these two interaction modes, we define two edge-level events that characterize how errors are transmitted or blocked along an interaction edge. The contamination event Z_{pv} occurs when the parent node p is erroneous and v accepts its information. The correction event Y_{pv} occurs when parent node p is erroneous and v critique it:

$$\begin{aligned} Z_{pv} &:= \mathbf{1}\{E_p = 1,\ M_{pv} = \mathrm{A}\}, \\ Y_{pv} &:= \mathbf{1}\{E_p = 1,\ M_{pv} = \mathrm{C}\}. \end{aligned}$$

Given $r_p = \Pr(E_p = 1)$, the corresponding contamination and correction probabilities are:

$$\Pr(Z_{pv} = 1) = \alpha_{pv} r_p, \quad \Pr(Y_{pv} = 1) = \beta_{pv} r_p.$$

Structurally, the overall error event E_v at node v occurs if and only if the agent either suffers from a local reasoning failure or is contaminated by at least one erroneous parent:

$$E_v = I_v \vee \left(\bigvee_{p \in \mathrm{Pa}(v)} Z_{pv} \right). \tag{1}$$

The correction event Y_{pv} does not appear in E_v because it is not an error-generating event.

3.2.2 Recursive Uncertainty Propagation

The event decomposition in Eq. (1) directly implies that node v is correct if and only if it does not suffer from a local error and none of its incoming edges transmit contamination:

$$\Pr(E_v = 0) = \Pr\left(I_v = 0, \bigwedge_{p \in \mathrm{Pa}(v)} Z_{pv} = 0 \right).$$

However, computing this joint probability exactly would require the joint distribution over local errors and incoming contamination events, which is not available during online MAS execution. We therefore introduce a propagation layer that only requires marginal parent uncertainties and edge-level acceptance probabilities, leading to the following conditional independence assumptions.

Assumption 1 (Edge-wise conditional independence)**.** *At node v, the incoming contamination events are represented by their marginal probabilities and treated as independent:*

$$\Pr\left(\bigwedge_{p \in \mathrm{Pa}(v)} Z_{pv} = 0 \right) = \prod_{p \in \mathrm{Pa}(v)} \Pr(Z_{pv} = 0).$$

Assumption 2 (Local-propagation independence)**.** *The local error event I_v is independent of the incoming contamination events $\{Z_{pv}\}_{p \in \mathrm{Pa}(v)}$ for each node v.*

Under Assumptions 1 and 2, the joint probability factorizes as:

$$\Pr(E_v = 0) = \Pr(I_v = 0) \prod_{p \in \mathrm{Pa}(v)} \Pr(Z_{pv} = 0).$$

Algorithm 1 Propagation-aware MAS UQ

Require: Execution graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.
Ensure: Propagated uncertainties $\{r_v\}_{v \in \mathcal{V}}$.

```
Obtain a topological ordering of G.
for each node v in topological order do
    Estimate local uncertainty u_v using a single-agent UQ method.
    for each parent p ∈ Pa(v) do
        Estimate α̂_pv by model self-reporting.
    end for
    if Pa(v) = ∅ then
        r_v ← u_v.
    else
        r_v ← 1 − (1 − u_v) ∏_{p∈Pa(v)} (1 − α̂_pv r_p).
    end if
end for
return {r_v}_{v∈V}.
```

Therefore, we obtain the node-wise uncertainty recurrence:

$$r_v = 1 - (1 - u_v) \prod_{p \in \mathrm{Pa}(v)} (1 - \alpha_{pv} r_p). \tag{2}$$

A formal derivation of Eq. (2) is provided in Appendix B.1. Eq. (2) gives the node-wise uncertainty propagation rule for MAS execution graphs. This recurrence updates the uncertainty of each agent at each step by combining its local uncertainty with the propagated risks.

3.3 Online Inference and Key Properties

As illustrated in Algorithm 1, PROPUQ-MAS instantiates the recursive propagation rule as an online reliability tracking procedure over the MAS execution graph. Since arbitrary MAS interaction structures can be unfolded into a DAG-structured execution trace, the same procedure applies across different MAS topologies. Given $\{u_v\}_{v \in \mathcal{V}}$ and $\{\hat{\alpha}_{pv}\}_{(p,v) \in \mathcal{E}}$, node-wise uncertainties are evaluated on-the-fly using Eq. (2). Crucially, since the uncertainty recurrence depends solely on immediate predecessors, r_v can be computed in a streaming fashion during MAS execution without requiring full trajectory completion. These propagated uncertainties serve as real-time reliability signals directly attached to each agent's intermediate messages at each step, which empower proactive downstream control policies, such as dynamic localized replanning, routing low-confidence steps to external verifiers, or executing early stopping to terminate unpromising paths for reliable MAS.

Properties of PROPUQ-MAS. Beyond online computability and topology-agnostic applicability, Eq. (2) also satisfies several desirable provable properties demonstrating its probabilistic validity, non-accumulative under critique, and computational efficiency, which support its use as a propagation-aware UQ layer. Detailed proofs are provided in Appendix B.

Proposition 1 (Boundedness). For any node $v \in \mathcal{V}$, if $u_v, r_p, \alpha_{pv} \in [0,1]$ for all $p \in \mathrm{Pa}(v)$, then the propagated uncertainty is bounded:

$$0 \leq r_v \leq 1.$$

In particular, if $\mathrm{Pa}(v) = \emptyset$, the uncertainty reduces to the boundary condition $r_v = u_v$.

Proposition 2 (Uncertainty Attenuation). Let $R_v = \max_{p \in \mathrm{Pa}(v)} r_p$ denote the maximum uncertainty among all parent nodes. If $u_v < R_v$ and

$$1 - \prod_{p \in \mathrm{Pa}(v)} (1 - \alpha_{pv} r_p) < \frac{R_v - u_v}{1 - u_v},$$

then $r_v < R_v$. Therefore, the propagation rule exhibits non-monotonicity, meaning a downstream agent can have lower uncertainty than its parent through low local uncertainty or rigorous critique (i.e., small α_{pv}).

Proposition 3 (Linear Scalability). For any MAS execution graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the node-wise uncertainties $\{r_v\}_{v \in \mathcal{V}}$ can be computed exactly in a single topological forward pass. The total time complexity scales linearly with the graph size:

$$\mathcal{O}(|\mathcal{V}| + |\mathcal{E}|).$$

Consequently, the linear complexity guarantees that Eq. (2) is computationally efficient, enabling real-time UQ with minimal overhead across arbitrary and dynamic MAS topologies.

4 Empirical Evaluations

Tasks and Datasets. We evaluate PROPUQ-MAS across three benchmarks spanning two reasoning-intensive domains: (i) **Math and Science Reasoning**, including GSM8K (Cobbe et al., 2021) and MedQA (Yang et al., 2025b); and (ii) **Code Generation**, including MBPP-Plus (Liu et al., 2023). Detailed descriptions of each benchmark are provided in Appendix C.1.

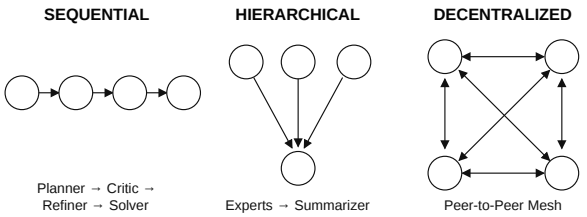

Figure 2: Illustration of the three evaluated MAS interaction topologies.

Base Models. We evaluate PROPUQ-MAS on open-source instruction-tuned models from two families: `Qwen3-4B`, `Qwen3-8B`, and `Qwen3-14B` from the Qwen3 family (Yang et al., 2025a), and `gemma-3-12b-it` (Gemma Team, 2025). Qwen3 models test robustness across model scales, while Gemma-3 evaluates cross-family generalization.

MAS Structures. To evaluate whether PROPUQ-MAS generalizes across different interaction topologies, we consider three representative MAS structures: *sequential* (Zhang et al., 2024; Zhao et al., 2025a), *hierarchical* (Zhao et al., 2026; Zhuge et al., 2024), and *decentralized* (Du et al., 2024; Liang et al., 2024). Unless otherwise specified, each MAS contains four agents following the common practice of LLM-MAS studies (Wang et al., 2025; Bajaj et al., 2026). This scale provides a controlled yet non-trivial setting, offering sufficient structural expressivity to map each topology while maintaining computational tractability. The evaluated MAS interaction topologies are illustrated in Figure 2, with detailed structure descriptions provided in Appendix C.2.

Evaluation Metrics. We evaluate uncertainty quality using AUROC and prediction rejection ratio (PRR). For each node output, we convert its correctness annotation into a binary error label, setting the error label to 1 if the output is incorrect and 0 otherwise. AUROC measures the ranking quality of uncertainty scores, testing whether incorrect outputs receive higher uncertainty than correct ones. In contrast, PRR evaluates the usefulness of uncertainty for selective prediction: outputs are rejected from most to least uncertain, and a better estimator should remove errors earlier, leading to higher retained accuracy. Thus, AUROC measures threshold-free error ranking, while PRR measures decision-level utility under uncertainty-based rejection. Detailed definitions are provided in Appendix C.3. Higher AUROC and PRR indicate better uncertainty estimation.

Table 1: Final-output uncertainty estimation across MAS topologies. PROPUQ-MAS is applied on top of each local UQ baseline; shaded cells indicate relative changes over the corresponding baseline, where green/red denotes gains/drops and darker colors indicate larger absolute changes.

Dataset	Metric	Qwen3-4B				Qwen3-8B				Gemma-3-12B			
		MSP	+PROPUQ	Verb.	+PROPUQ	MSP	+PROPUQ	Verb.	+PROPUQ	MSP	+PROPUQ	Verb.	+PROPUQ
Sequential MAS													
GSM8K	AUROC	0.578	**0.617**	0.466	**0.501**	0.605	**0.625**	0.505	**0.680**	0.471	**0.517**	0.581	**0.651**
	PRR	0.160	**0.238**	-0.073	-0.068	0.224	**0.257**	-0.028	**0.340**	0.005	**0.102**	0.156	**0.250**
MBPP+	AUROC	0.679	**0.722**	0.577	**0.607**	0.583	**0.648**	0.562	**0.577**	0.544	**0.583**	0.684	**0.742**
	PRR	0.340	**0.378**	0.090	**0.173**	0.155	**0.282**	0.064	**0.120**	0.079	**0.128**	0.277	**0.465**
MedQA	AUROC	0.649	**0.742**	0.428	**0.607**	0.754	**0.842**	0.600	**0.791**	0.635	**0.668**	0.627	**0.674**
	PRR	0.254	**0.410**	-0.167	**0.236**	0.496	**0.624**	0.217	**0.482**	0.232	**0.240**	0.237	**0.322**
Hierarchical MAS													
GSM8K	AUROC	0.564	0.541	0.535	**0.597**	0.664	**0.717**	0.516	**0.577**	0.535	0.498	0.549	**0.606**
	PRR	0.100	0.094	0.082	**0.222**	0.321	**0.446**	-0.001	**0.153**	0.063	0.060	0.080	**0.204**
MBPP+	AUROC	0.688	**0.718**	0.632	**0.661**	0.619	**0.719**	0.580	**0.630**	0.630	**0.674**	0.682	**0.730**
	PRR	0.335	**0.432**	0.201	**0.289**	0.172	**0.401**	0.081	**0.245**	0.204	**0.207**	0.382	**0.423**
MedQA	AUROC	0.628	**0.760**	0.621	**0.705**	0.658	**0.711**	0.604	**0.722**	0.567	**0.604**	0.660	**0.669**
	PRR	0.238	**0.417**	0.216	**0.296**	0.277	**0.349**	0.223	**0.362**	0.115	**0.184**	0.358	0.351
Decentralized MAS													
GSM8K	AUROC	0.657	**0.704**	0.544	**0.574**	0.728	**0.768**	0.486	**0.563**	0.563	**0.567**	0.634	**0.665**
	PRR	0.311	**0.428**	0.128	**0.185**	0.445	**0.534**	-0.004	**0.181**	0.128	**0.162**	0.271	**0.294**
MBPP+	AUROC	0.695	**0.710**	0.588	**0.604**	0.701	**0.726**	0.511	**0.583**	0.557	**0.645**	0.670	**0.720**
	PRR	0.382	**0.426**	0.140	**0.210**	0.332	**0.367**	0.014	**0.117**	0.096	**0.262**	0.342	**0.393**
MedQA	AUROC	0.749	**0.824**	0.693	**0.731**	0.812	**0.821**	0.718	**0.854**	0.599	**0.636**	0.576	**0.611**
	PRR	0.416	**0.563**	0.355	0.337	0.539	**0.568**	0.429	**0.626**	0.155	**0.208**	0.160	**0.200**

Baselines. To assess compatibility with different single-agent UQ methods, we select two representative local uncertainty estimators for LLMs: verbalized self-evaluation (Verb.) (Tian et al., 2023) and Maximum Sequence Probability (MSP) (Vashurin et al., 2025) for token-level predictive probability. Each estimator is evaluated as a standalone baseline and is also used to provide the local uncertainty u_v for PROPUQ-MAS. Because PROPUQ-MAS operates with a single MAS forward pass, we treat sampling-based UQ methods that require multiple sampled executions as complementary rather than local UQ baselines. We nevertheless compare with concurrent sampling-based methods, including MATU (Chen et al., 2026) and UProp (Duan et al., 2025), in Appendix D.3.

Implementation Details. All main experiments are run on NVIDIA GH200 120GB GPUs. Unless otherwise specified, generation uses temperature 0.6, top-p 0.95, and repetition penalty 1.05. The maximum generation length is set to 8192 tokens for math and science reasoning tasks and 32768 tokens for code-generation tasks.

4.1 Final-Output UQ Performance

We first evaluate whether PROPUQ-MAS improves uncertainty estimation for the final answer produced by a MAS. For each instance, we compare the uncertainty score of the final output produced by a standalone local UQ baseline with the propagated uncertainty score computed by PROPUQ-MAS. This comparison tests whether modeling uncertainty propagation better detects final-answer errors.

Table 1 shows that PROPUQ-MAS improves final-output UQ in most settings, with median relative gains of $+7.32\%$ in AUROC and $+41.36\%$ in PRR across datasets, model families, and MAS topologies. The gains are especially clear when the local UQ baseline is weak or noisy, suggesting that final-agent confidence alone can miss errors inherited from upstream agents. By incorporating parent uncertainty and edge-level acceptance, PROPUQ-MAS captures this interaction-induced risk and yields more informative final-answer uncertainty. The results hold across model families and scales, suggesting that PROPUQ-MAS works as a training-free, model-agnostic propagation layer rather than a model-specific calibration method. The overall performance gain supports the importance of explicitly modeling uncertainty propagation in MAS reliability assessment. Additional results on `Qwen3-14B` are provided in Appendix D.

Table 2: Intermediate-agent uncertainty estimation under decentralized MAS. PROPUQ cells are shaded by relative change over each baseline: green/red indicates gains/drops, and darker colors indicate larger absolute changes.

Dataset	Agent	Metric	Qwen3-4B				Qwen3-8B				Gemma-3-12B			
			MSP	+PROPUQ	Verb.	+PROPUQ	MSP	+PROPUQ	Verb.	+PROPUQ	MSP	+PROPUQ	Verb.	+PROPUQ
GSM8K	Agent 2	AUROC	0.667	**0.695**	0.586	**0.622**	0.754	**0.769**	0.544	**0.601**	0.589	**0.591**	0.640	**0.664**
		PRR	0.372	**0.421**	0.144	**0.224**	0.499	**0.528**	0.056	**0.201**	0.185	0.184	0.248	**0.284**
	Agent 3	AUROC	0.668	**0.692**	0.549	**0.601**	0.724	**0.760**	0.556	**0.625**	0.561	**0.566**	0.629	**0.663**
		PRR	0.369	**0.421**	0.120	**0.221**	0.416	**0.509**	0.084	**0.245**	0.125	**0.151**	0.257	**0.276**
MBPP+	Agent 2	AUROC	0.692	**0.734**	0.569	**0.590**	0.690	**0.729**	0.490	**0.546**	0.687	0.642	0.666	**0.691**
		PRR	0.368	**0.454**	0.126	**0.169**	0.325	**0.373**	-0.026	**0.098**	0.355	0.353	0.361	**0.395**
	Agent 3	AUROC	0.743	**0.761**	0.627	**0.630**	0.699	0.698	0.566	**0.602**	0.639	**0.665**	0.663	**0.705**
		PRR	0.415	**0.484**	0.253	0.252	0.349	**0.355**	0.158	**0.190**	0.241	**0.290**	0.339	**0.395**
MedQA	Agent 2	AUROC	0.770	**0.799**	0.715	**0.752**	0.796	**0.810**	0.667	**0.816**	0.656	**0.667**	0.622	**0.625**
		PRR	0.481	**0.513**	0.408	**0.423**	0.477	**0.509**	0.338	**0.509**	0.284	**0.298**	0.297	0.290
	Agent 3	AUROC	0.781	**0.812**	0.629	**0.701**	0.776	**0.822**	0.695	**0.824**	0.581	**0.634**	0.553	**0.610**
		PRR	0.479	**0.554**	0.239	**0.285**	0.459	**0.539**	0.390	**0.562**	0.174	**0.212**	0.169	**0.196**

Table 3: Intermediate-agent uncertainty estimation under sequential MAS on MedQA with Qwen3-8B, using GPT-5.5 judgments.

Agent	UQ Method	AUROC	PRR
Critic	Verb.	0.7516	0.4264
	+PROPUQ	**0.7872**	**0.5779**
Refiner	Verb.	0.6709	0.3869
	+PROPUQ	**0.7886**	**0.5012**

4.2 Intermediate-Step UQ Performance

We further evaluate whether PROPUQ-MAS provides reliable uncertainty signals for intermediate MAS outputs, rather than only for the final answer. This experiment tests a key requirement of MAS reliability assessment: the capacity to pinpoint early-stage failures that precipitate final system errors.

4.2.1 Verifiable Intermediate Steps

We first conduct quantitative analysis within the decentralized MAS setting, where each intermediate agent output is a complete candidate solution and can be checked against the ground-truth label. This configuration enables a direct intermediate-step evaluation of uncertainty scores. We compare PROPUQ-MAS with local UQ baselines to measure the benefit of modeling interaction-aware propagation at each step.

Table 2 reports intermediate-agent UQ results, with node-averaged results in Appendix Table 7. We omit Agent 1 because it has no incoming messages, so its propagated uncertainty is identical to its local uncertainty, i.e., $r_v = u_v$. We also omit Agent 4 in the intermediate-agent table because its output corresponds to the final MAS answer, whose results are already reported in Table 1. Across the evaluated intermediate agents, PROPUQ-MAS achieves average relative gains of $+6.10\%$ in AUROC and $+47.58\%$ in PRR, showing that modeling propagated risk helps identify unreliable intermediate outputs, not only unreliable final answers. The PRR gains indicate that high-risk intermediate steps can be prioritized for rejection or intervention before they affect later agents.

4.2.2 LLM-Judged Reasoning Steps

For intermediate reasoning steps without direct ground-truth labels, we conduct an LLM-judged evaluation on 300 randomly sampled MedQA instances under the sequential MAS setting using Qwen3-8B. We use GPT-5.5 as an LLM judge (Zheng et al., 2023) to annotate Critic and Refiner outputs as erroneous when they contain substantive medical or reasoning errors likely to mislead downstream agents; the judge prompt is provided in Appendix E.4. We exclude the Planner because it is a source node with no upstream inputs and therefore does not involve uncertainty propagation. For hierarchical MAS, the summarizer is the only node that aggregates upstream uncertainty, and its output is the final answer already evaluated in Table 1. The expert agents are source nodes, leaving no additional propagated intermediate node for evaluation.

Table 3 shows consistent improvements from PROPUQ-MAS for both intermediate agents. On 50 randomly selected examples, human annotations agree with the GPT-5.5 judgments in 92.0% of cases, supporting the reliability of the judge-derived labels. These results show that propagation-aware UQ extends beyond complete candidate answers to intermediate reasoning states.

Overall, these results support PROPUQ-MAS as an online reliability monitor for MAS execution.

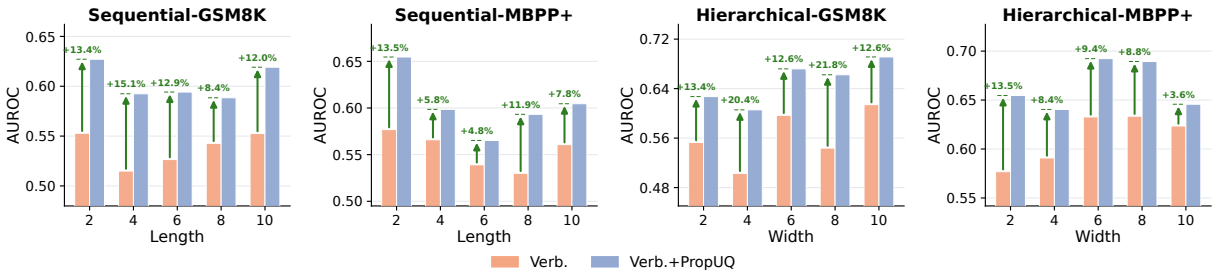

Figure 3: Structural scale generalization results measured by AUROC. We evaluate the robustness of uncertainty propagation across varying sequential chain lengths and hierarchical expert-layer widths.

Table 4: Ablation on self-reported acceptance weights using Qwen3-8B. The w/o $\hat{\alpha}$ variant fixes all edge weights to one, while w/ $\hat{\alpha}$ uses self-reported acceptance scores. Bold marks the better value in each pair.

Dataset	Metric	Sequential		Hierarchical		Decentralized	
		w/o $\hat{\alpha}$	w/ $\hat{\alpha}$	w/o $\hat{\alpha}$	w/ $\hat{\alpha}$	w/o $\hat{\alpha}$	w/ $\hat{\alpha}$
GSM8K	AUROC	0.672	**0.680**	0.569	**0.577**	0.519	**0.563**
	PRR	0.331	**0.340**	0.148	**0.153**	0.142	**0.181**
MBPP+	AUROC	0.573	**0.577**	0.615	**0.630**	0.576	**0.583**
	PRR	0.114	**0.120**	0.239	**0.245**	0.093	**0.117**
MedQA	AUROC	0.785	**0.791**	0.719	**0.722**	0.848	**0.854**
	PRR	0.476	**0.482**	0.345	**0.362**	0.612	**0.626**

By attaching propagated uncertainty to intermediate steps, it can identify early-stage errors and support interventions such as localized replanning, external verification, additional critique, or early stopping before these errors affect the final answer. Additional results on `Qwen3-14B` are provided in Appendix D.

4.3 MAS Scale Analysis

We further examine whether PROPUQ-MAS remains effective as MAS scale changes. Following prior work on MAS scaling (Qian et al., 2025), we vary two factors that directly affect uncertainty propagation: the chain length in sequential systems and the number of parent inputs in hierarchical systems. Each agent acts as both a critic and a solver, keeping agent behavior comparable across scales while preserving reasoning capability. Prompts are provided in Appendix E.

Figure 3 reports AUROC results on GSM8K and MBPP-Plus, with additional PRR results in Appendix D. PROPUQ-MAS consistently improves over the local verbalized UQ baseline across all tested sequential lengths and hierarchical widths. These results indicate that the propagation rule can track uncertainty over longer chains and aggregate risks from larger parent sets, supporting its scalability under different MAS sizes.

4.4 Ablation Study

We conduct an ablation study to examine whether self-reported acceptance scores provide useful edge-level signals for uncertainty propagation. The ablated variant, w/o $\hat{\alpha}$, sets all edge weights to 1, uniformly propagating parent uncertainty regardless of downstream acceptance or critique. The full model, w/ $\hat{\alpha}$, uses the self-reported acceptance score as the plug-in transmission weight for each edge. Table 4 shows that w/ $\hat{\alpha}$ consistently improves over w/o $\hat{\alpha}$ on both AUROC and PRR across three MAS topologies with Qwen3-8B, indicating that self-reported $\hat{\alpha}$ provides useful interaction signals beyond uniform uncertainty propagation. The results support the correction-sensitive design of PROPUQ-MAS, where accepted upstream risks contribute more than critiqued ones.

4.5 Validation of Self-Reported Edge Weights

The error-conditioned edge parameter $\alpha_{pv} = \Pr(M_{pv} = A \mid E_p = 1)$ is not directly accessible during online execution, as it depends on the correctness of the parent output. We therefore examine whether the self-reported proxy $\hat{\alpha}_{pv}$ preserves the relative strength of error acceptance. Under the decentralized MAS topology, we conduct an edge-level analysis across three datasets and four models. Specifically, we isolate execution edges whose parent is incorrect ($E_p = 1$), bin them by the child node's self-reported adoption score $\hat{\alpha}_{pv}$, and measure the error-inheritance rate, i.e., how often the child duplicates the parent's error.

Figure 4 shows a monotonic alignment between the self-reported score and the empirical error-inheritance rate, increasing from 0.050 in the lowest bin to 0.973 in the highest. This monotonicity demonstrates that the self-reported $\hat{\alpha}_{pv}$ serves as a reliable proxy for error-conditioned acceptance strength. Crucially, it provides this signal online without additional training or inference passes.

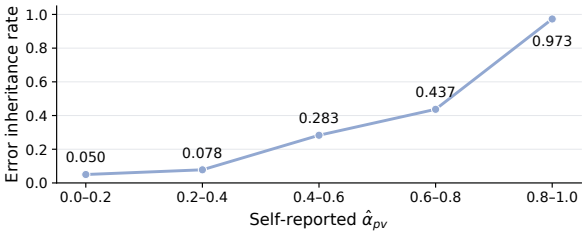

Figure 4: Child error-inheritance rate across adoption-score bins for incorrect-parent cases, aggregated over three datasets and four models.

More generally, the PROPUQ-MAS propagation recurrence in Eq. (2) is agnostic to the specific choice of edge-weight estimator. When labeled calibration data are available, the self-reported $\hat{\alpha}pv$ can therefore be replaced by calibrated or learned estimators that more accurately approximate the error-conditioned edge parameter α_{pv}.

4.6 Case Study

Figure 5 presents a GSM8K case under the hierarchical MAS topology. The summarizer is locally confident but follows two erroneous upstream agents while assigning low acceptance to the correct one, leading to an incorrect final answer. This illustrates a common MAS failure mode: local confidence can underestimate risk when downstream agents selectively rely on unreliable upstream information. By combining parent uncertainty with edge-level acceptance, PROPUQ-MAS assigns higher propagation-aware uncertainty to the final output, making the propagated error more detectable. A complementary sequential MAS case study is provided in Appendix D.5.

Case Study from GSM8K under Hierarchical MAS

Question. A team of 4 painters worked on a mansion for $3/8$ of a day every day for 3 weeks. How many hours of work did each painter put in?

Gold Answer: 189 **Final Prediction:** 47.25 (incorrect)

Math Agent. *Answer:* compute 3 weeks as 21 days and $3/8$ of a day as 9 hours, giving $21 \times 9 = 189$ hours. Then divide by 4 painters, concluding $189/4 = 47.25$ hours per painter.

Local uncertainty: 0.20
Propagation-aware uncertainty: 0.20
Acceptance by Summarizer: 0.90

Science Agent. *Answer:* treat $21 \times 3/8 = 63/8$ days as total team time, divide by 4 painters to get $63/32$ days per painter, and convert to hours: $(63/32) \times 24 = 47.25$.

Local uncertainty: 0.30
Propagation-aware uncertainty: 0.30
Acceptance by Summarizer: 0.90

Code Agent. *Answer:* each painter works $3/8$ of a day, i.e., $(3/8) \times 24 = 9$ hours per day. Over 3 weeks, this is 21 days, so each painter works $9 \times 21 = 189$ hours.

Local uncertainty: 0.05
Propagation-aware uncertainty: 0.05
Acceptance by Summarizer: 0.30

Summarizer. *I follow the Math and Science agents:* first compute 21 days and 9 hours per day, giving 189 hours. Since there are 4 painters, divide by 4 and output 47.25. The Code Agent's answer 189 is treated as total team hours rather than per-painter hours.

Local uncertainty: 0.10
Propagation-aware uncertainty: 0.469

Figure 5: Case study from GSM8K under the hierarchical MAS topology. The summarizer follows erroneous agents with low local uncertainty, while PROPUQ-MAS captures the propagated risk.

5 Conclusion

We introduce PROPUQ-MAS, a propagation-aware uncertainty quantification framework for LLM-based multi-agent systems. Unlike standard UQ methods that estimate the reliability of isolated responses, PROPUQ-MAS models MAS execution as a communication-structured graph and estimates node-wise reliability by combining intrinsic local uncertainty with cascading uncertainty inherited from upstream messages. Its recursive propagation rule enables efficient uncertainty tracking in a single topological forward pass without additional training. Experiments across reasoning and code-generation tasks show consistent improvements over single-agent UQ baselines for both final-answer and intermediate-step UQ, with gains across topologies and backbone models.

Acknowledgments

This research is supported in part by the National Science Foundation under Grant No. CNS-2427070, IIS-2331069, IIS-2202481, IIS-2130263, CNS-2131622. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

Limitations

Despite its empirical effectiveness, PROPUQ-MAS has several limitations.

First, the theoretical propagation rule uses the latent error-conditioned transmission parameter α_{pv}, while our implementation instantiates it with a self-reported plug-in estimate $\hat{\alpha}_{pv}$. Since the parent error event is unobserved during online execution, this self-reported score may be imperfectly calibrated as a probability. We therefore view $\hat{\alpha}_{pv}$ as an online proxy for edge-level transmission strength rather than a direct observation of the true error-conditioned parameter. Future work could replace this proxy with calibrated, verifier-based, or learned transmission estimators.

Second, our propagation rule relies on edge-wise conditional independence and local-propagation independence to obtain a tractable factorization. These assumptions may be violated when upstream messages jointly affect downstream reasoning or when misleading context simultaneously increases local generation error and acceptance behavior, such as under severe prompt injection or context overload. Despite these approximations, the factorized rule enables efficient single-pass uncertainty estimation and empirically improves over local UQ baselines across tasks, models, and MAS topologies. Future work could relax these assumptions by modeling correlations among local errors, acceptance behavior, and propagated uncertainty.

Third, PROPUQ-MAS depends on the quality of the local UQ estimator at each node. While the framework is agnostic to the specific local UQ estimator and can readily incorporate calibrated alternatives, such as temperature scaling or isotonic regression, these methods typically require labeled calibration data and may be less practical for online MAS execution. We therefore use lightweight, single-pass MSP and verbalized uncertainty to support real-time inference. Incorporating stronger local UQ estimators is complementary to our propagation framework and may further improve propagation-aware uncertainty estimation.

References

Tanav Singh Bajaj, Nikhil Singh, Karan Anand, and Eishkaran Singh. 2026. Position: Safety and fairness in agentic ai depend on interaction topology, not on model scale or alignment. *arXiv preprint arXiv:2605.01147*.

Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya G. Parameswaran, Dan Klein, Kannan Ramchandran, Matei A. Zaharia, Joseph E. Gonzalez, and Ion Stoica. 2026. Why do multi-agent llm systems fail? *Advances in Neural Information Processing Systems*, 38.

Tiejin Chen, Huaiyuan Yao, Jia Chen, Evangelos E Papalexakis, and Hua Wei. 2026. Every response counts: Quantifying uncertainty of llm-based multi-agent systems through tensor decomposition. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16204–16218.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first international conference on machine learning*.

Jinhao Duan, James Diffenderfer, Sandeep Madireddy, Tianlong Chen, Bhavya Kailkhura, and Kaidi Xu. 2025. Uprop: Investigating the uncertainty propagation of llms in multi-step agentic decision-making. *arXiv preprint arXiv:2506.17419*.

Ekaterina Fadeeva, Aleksandr Rubashevskii, Artem Shelmanov, Sergey Petrakov, Haonan Li, Hamdy Mubarak, Evgenii Tsymbalov, Gleb Kuzmin, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, and Maxim Panov. 2024. Fact-checking the output of large language models via token-level uncertainty quantification. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9367–9385.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Gemma Team. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2024. MetaGPT: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*.

Jinwei Hu, Yi Dong, Shuang Ao, Zhuoyun Li, Boxuan Wang, Lokesh Singh, Guangliang Cheng, Sarvapali D

Ramchurn, and Xiaowei Huang. 2025. Position: Towards a responsible llm-empowered multi-agent systems. *arXiv preprint arXiv:2502.01714*.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, and 17 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Michael Kirchhof, Gjergji Kasneci, and Enkelejda Kasneci. 2025. Position: Uncertainty quantification needs reassessment for large language model agents. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025 - Position Paper Track*.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. CAMEL: Communicative agents for "mind" exploration of large language model society. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 17889–17904.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. 2023. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in neural information processing systems*, 36:21558–21572.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, and 3 others. 2024. Agentbench: Evaluating LLMs as agents. In *The Twelfth International Conference on Learning Representations*.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 9004–9017.

Chen Qian, Zihao Xie, Yifei Wang, Wei Liu, Kunlun Zhu, Hanchen Xia, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, Zhiyuan Liu, and Maosong Sun. 2025. Scaling large language model-based multi-agent collaboration. In *International Conference on Learning Representations*, volume 2025, pages 41488–41505.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442.

Roman Vashurin, Ekaterina Fadeeva, Artem Vazhentsev, Lyudmila Rvanova, Daniil Vasilev, Akim Tsvigun, Sergey Petrakov, Rui Xing, Abdelrahman Boda Sadallah, Kirill Grishchenkov, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, Maxim Panov, and Artem Shelmanov. 2025. Benchmarking uncertainty quantification methods for large language models with lm-polygraph. *Transactions of the Association for Computational Linguistics*, 13:220–248.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Y Zou. 2025. Mixture-of-agents enhances large language model capabilities. In *International Conference on Learning Representations*, volume 2025, pages 33944–33963.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. 2024. Autogen: Enabling next-gen LLM applications via multi-agent conversations. In *First Conference on Language Modeling*.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2024. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. In *International Conference on Learning Representations*, volume 2024, pages 23650–23678.

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. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Hang Yang, Hao Chen, Hui Guo, Yineng Chen, Ching-Sheng Lin, Shu Hu, Jinrong Hu, Xi Wu, and Xin Wang. 2025b. Llm-medqa: Enhancing medical question answering through case studies in large language models. In *2025 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Shaokun Zhang, Ming Yin, Jieyu Zhang, Jiale Liu, Zhiguang Han, Jingyang Zhang, Beibin Li, Chi Wang, Huazheng Wang, Yiran Chen, and Qingyun Wu. 2025. Which agent causes task failures and when? on automated failure attribution of LLM multi-agent systems. In *Forty-second International Conference on Machine Learning*.

Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Ö Arık. 2024. Chain of agents: Large language models collaborating on long-context tasks. *Advances in Neural Information Processing Systems*, 37:132208–132237.

Jiaxing Zhao, Hongbin Xie, Yuzhen Lei, Xuan Song, Zhuoran Shi, Lianxin Li, Shuangxue Liu, and Haoran Zhang. 2025a. Connecting the dots: A chain-of-collaboration prompting framework for llm agents. *arXiv preprint arXiv:2505.10936*.

Qiwei Zhao, Dong Li, Yanchi Liu, Wei Cheng, Yiyou Sun, Mika Oishi, Takao Osaki, Katsushi Matsuda, Huaxiu Yao, Chen Zhao, Haifeng Chen, and Xujiang Zhao. 2025b. 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, Vienna, Austria. Association for Computational Linguistics.

Wanjia Zhao, Mert Yuksekgonul, Shirley Wu, and James Zou. 2026. Sirius: Self-improving multi-agent systems via bootstrapped reasoning. *Advances in Neural Information Processing Systems*, 38:124475–124504.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. 2024. Language agents as optimizable graphs. *arXiv preprint arXiv:2402.16823*.

A Acceptance Probability Elicitation

We instantiate the edge-level transmission parameter α_{pv} using a training-free self-reporting procedure. In the probabilistic formulation, α_{pv} denotes the error-conditioned acceptance probability, i.e., the probability that node v accepts information from parent node p when the parent output is erroneous. During online MAS execution, this true error-conditioned quantity is not directly observable, since the parent error variable E_p is unknown. We therefore use a self-reported adoption score as a plug-in estimate of the edge-level transmission strength.

Specifically, after node v is generated, the corresponding agent is prompted to report an adoption score $\hat{\alpha}_{pv} \in [0, 1]$ for each parent $p \in \mathrm{Pa}(v)$. The score indicates the degree to which the current output relies on the information from parent p, rather than critiquing, revising, or rejecting it. Let s_{pv} denote the reported adoption score. We set

$$\hat{\alpha}_{pv} = s_{pv}, \qquad \hat{\beta}_{pv} = 1 - \hat{\alpha}_{pv}.$$

In the online recurrence, $\hat{\alpha}_{pv}$ is used as the empirical edge weight in Eq. (2), serving as a proxy for the latent transmission parameter α_{pv}.

This procedure introduces no additional training and can be applied to arbitrary MAS topologies. Since the elicited score is a self-reported proxy rather than a direct observation of the error-conditioned probability, its calibration may be imperfect. Nevertheless, the mathematical properties in Section 3.3, such as boundedness, attenuation, and linear scalability, only require edge weights in $[0, 1]$ and therefore also hold for the plug-in weights used during online inference. The prompt template is shown in Figure 10.

B Detailed Proofs and Derivations

This appendix provides the formal derivation of the uncertainty recurrence presented in Eq. (2), followed by detailed proofs of the key properties outlined in Section 3.3.

B.1 Derivation of the Uncertainty Recurrence

We derive the node-wise uncertainty recurrence in Eq. (2). Recall from the structural error composition in Eq. (1) that the overall error event E_v at node v is defined as:

$$E_v = I_v \vee \left(\bigvee_{p \in \mathrm{Pa}(v)} Z_{pv} \right).$$

Applying De Morgan's laws, the complement event, representing a successful and error-free output at node v, occurs if and only if the agent avoids both local intrinsic reasoning failures and cascading external contamination:

$$E_v = 0 \iff I_v = 0 \quad \text{and} \quad Z_{pv} = 0, \quad \forall p \in \mathrm{Pa}(v).$$

Expressing this structural relationship in terms of joint probability yields:

$$\Pr(E_v = 0) = \Pr\left(I_v = 0 \ \wedge \bigwedge_{p \in \mathrm{Pa}(v)} Z_{pv} = 0 \right).$$

By invoking Assumption 2 (Local-Propagation Independence), the local error event I_v is conditionally independent of the incoming edge-level propagation events. This conditional independence allows us to factorize the joint probability as follows:

$$\Pr(E_v = 0) = \Pr(I_v = 0) \cdot \Pr\left(\bigwedge_{p \in \mathrm{Pa}(v)} Z_{pv} = 0 \right).$$

Next, by applying Assumption 1 (Edge-Wise Conditional Independence), the contamination events are mutually independent across distinct incoming edges. This tracking condition reduces the joint probability of the intersection to a product of marginal probabilities:

$$\Pr\left(\bigwedge_{p \in \mathrm{Pa}(v)} Z_{pv} = 0 \right) = \prod_{p \in \mathrm{Pa}(v)} \Pr(Z_{pv} = 0).$$

Combining these two factorizations yields the fully decoupled expression for the success probability:

$$\Pr(E_v = 0) = \Pr(I_v = 0) \prod_{p \in \mathrm{Pa}(v)} \Pr(Z_{pv} = 0). \tag{3}$$

By definition, the success probability of isolated local reasoning is $\Pr(I_v = 0) = 1 - u_v$. For each edge-level contamination event Z_{pv}, its marginal probability can be expanded by conditioning on the parent error state E_p:

$$\begin{aligned} \Pr(Z_{pv} = 1) &= \Pr(E_p = 1 \ \wedge \ M_{pv} = \mathrm{A}) \\ &= \Pr(M_{pv} = \mathrm{A} \mid E_p = 1) \Pr(E_p = 1) \\ &= \alpha_{pv} r_p, \end{aligned}$$

which directly implies $\Pr(Z_{pv} = 0) = 1 - \alpha_{pv} r_p$. Substituting these probability terms back into Eq. (3), we obtain:

$$\Pr(E_v = 0) = (1 - u_v) \prod_{p \in \mathrm{Pa}(v)} (1 - \alpha_{pv} r_p).$$

Finally, utilizing the fundamental complement relation for total node uncertainty, $r_v = \Pr(E_v = 1) = 1 - \Pr(E_v = 0)$, we arrive at the exact closure:

$$r_v = 1 - (1 - u_v) \prod_{p \in \mathrm{Pa}(v)} (1 - \alpha_{pv} r_p),$$

which completes the proof of Eq. (2).

Boundary Condition for Source Nodes. By mathematical convention, for any source node v with no predecessors (i.e., $\mathrm{Pa}(v) = \emptyset$), the empty product evaluates to 1. Under this condition, the recurrence naturally simplifies to $r_v = u_v$, preserving structural consistency across arbitrary graph entries.

B.2 Proof of Proposition 1 (Boundedness)

We verify that the local uncertainty recurrence relation preserves the algebraic constraints of a valid probability measure. Given the assumption that $u_v, r_p, \alpha_{pv} \in [0, 1]$ for all $p \in \mathrm{Pa}(v)$, it follows that $0 \leq \alpha_{pv} r_p \leq 1$ for each incoming dependency edge $(p, v) \in \mathcal{E}$. This directly implies:

$$0 \leq 1 - \alpha_{pv} r_p \leq 1.$$

Since the unit interval $[0, 1]$ is structurally closed under finite multiplication, the joint product over the parent configuration satisfies:

$$0 \leq \prod_{p \in \mathrm{Pa}(v)} (1 - \alpha_{pv} r_p) \leq 1.$$

Multiplying this product by the bounded local success probability $1 - u_v \in [0, 1]$ preserves the interval containment:

$$0 \leq (1 - u_v) \prod_{p \in \mathrm{Pa}(v)} (1 - \alpha_{pv} r_p) \leq 1.$$

Finally, substituting this term back into the complementation formula of Eq. (2) yields:

$$r_v = 1 - (1 - u_v) \prod_{p \in \mathrm{Pa}(v)} (1 - \alpha_{pv} r_p) \implies 0 \leq r_v \leq 1.$$

In the boundary case where $\mathrm{Pa}(v) = \emptyset$, the empty product evaluates to 1, and the expression smoothly collapses to $r_v = 1 - (1 - u_v) = u_v$, ensuring perfect alignment with the boundary condition. This proves Proposition 1.

B.3 Proof of Proposition 2: Uncertainty Attenuation

Consider a non-source node v such that $\mathrm{Pa}(v) \neq \emptyset$, and let $R_v = \max_{p \in \mathrm{Pa}(v)} r_p$ denote the maximum uncertainty among its parent nodes. To facilitate the algebraic analysis, we define the aggregate incoming contamination probability as:

$$B_v := 1 - \prod_{p \in \mathrm{Pa}(v)} (1 - \alpha_{pv} r_p).$$

By rearranging the terms of the core recurrence relation in Eq. (2), the node-wise uncertainty r_v can be equivalently expressed as a linear combination of local uncertainty and external contamination:

$$r_v = u_v + (1 - u_v) B_v. \tag{4}$$

By hypothesis, we assume that the node possesses a lower local risk than its parents, $u_v < R_v$, and that its aggregate contamination is bounded by:

$$B_v < \frac{R_v - u_v}{1 - u_v}. \tag{5}$$

Since $R_v \leq 1$, the condition $u_v < R_v$ guarantees that $u_v < 1$, which ensures that the denominator $(1 - u_v)$ is strictly positive. Consequently, we can multiply both sides of the inequality in Eq. (5) by $(1 - u_v)$ without reversing the inequality sign:

$$(1 - u_v) B_v < R_v - u_v.$$

Adding u_v to both sides yields:

$$u_v + (1 - u_v) B_v < R_v.$$

Finally, substituting the rewritten recurrence relation from Eq. (4) into the left-hand side of the inequality directly produces:

$$r_v < R_v.$$

This demonstrates that a downstream node can achieve a lower uncertainty score than its predecessors, formally establishing that risk does not monotonically accumulate along the MAS execution trajectory. This completes the proof of Proposition 2.

B.4 Proof of Proposition 3: Linear Scalability

By construction, the MAS execution trajectory is represented as a directed acyclic graph (DAG) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, which inherently admits a topological sorting of its vertices, denoted by the sequence

Table 5: Qwen3-14B final-output uncertainty estimation across MAS topologies. PROPUQ cells are shaded by relative change over each baseline: green/red indicates gains/drops, and darker colors indicate larger absolute changes.

Dataset	Metric	Qwen3-14B			
		MSP	+PROPUQ	Verb.	+PROPUQ
Sequential MAS					
GSM8K	AUROC	0.715	0.677	0.580	**0.651**
	PRR	0.403	0.374	0.159	**0.271**
MBPP+	AUROC	0.690	0.655	0.544	**0.569**
	PRR	0.291	**0.321**	-0.005	**0.111**
MedQA	AUROC	0.734	**0.835**	0.588	**0.756**
	PRR	0.378	**0.601**	0.178	**0.393**
Hierarchical MAS					
GSM8K	AUROC	0.649	**0.723**	0.543	**0.557**
	PRR	0.237	**0.375**	0.076	**0.105**
MBPP+	AUROC	0.674	**0.734**	0.611	**0.647**
	PRR	0.278	**0.399**	0.221	**0.255**
MedQA	AUROC	0.662	**0.737**	0.681	0.664
	PRR	0.239	**0.403**	0.249	0.237
Decentralized MAS					
GSM8K	AUROC	0.608	**0.659**	0.586	**0.612**
	PRR	0.200	**0.307**	0.154	**0.181**
MBPP+	AUROC	0.710	**0.732**	0.575	**0.608**
	PRR	0.350	**0.381**	0.128	**0.174**
MedQA	AUROC	0.753	**0.813**	0.684	**0.686**
	PRR	0.401	**0.535**	0.372	0.346

$v_1, v_2, \ldots, v_{|\mathcal{V}|}$. For any directed dependency edge $(p, v) \in \mathcal{E}$, the predecessor p strictly precedes the successor v within this sequence. This structural dependency guarantees that evaluating the recursive mapping in Eq. (2) sequentially according to the topological order always ensures that all requisite parent uncertainties $\{r_p : p \in \mathrm{Pa}(v)\}$ are fully computed prior to visiting node v.

The local calculation for an individual node v requires a single pass over its immediate parent set $\mathrm{Pa}(v)$:

$$r_v = 1 - (1 - u_v) \prod_{p \in \mathrm{Pa}(v)} (1 - \alpha_{pv} r_p).$$

The computational overhead for evaluating this node-wise recurrence scales linearly with its in-degree, yielding a local time complexity of $\mathcal{O}(|\mathrm{Pa}(v)|)$. Summing these individual operational costs across the entire vertex set $\mathcal{V}$ leverages a fundamental graph-theoretic identity:

$$\sum_{v \in \mathcal{V}} \mathcal{O}(|\mathrm{Pa}(v)|) = \mathcal{O}\left(\sum_{v \in \mathcal{V}} |\mathrm{Pa}(v)|\right) = \mathcal{O}(|\mathcal{E}|).$$

Table 6: Qwen3-14B intermediate-agent uncertainty estimation under decentralized MAS. PROPUQ cells are shaded by relative change over each baseline: green/red indicates gains/drops, and darker colors indicate larger absolute changes.

Dataset	Agent	Metric	Qwen3-14B			
			MSP	+PROPUQ	Verb.	+PROPUQ
GSM8K	Agent 2	AUROC	0.645	**0.656**	0.564	**0.593**
		PRR	0.272	**0.309**	0.100	**0.125**
	Agent 3	AUROC	0.673	**0.689**	0.516	**0.556**
		PRR	0.305	**0.352**	0.037	**0.064**
MBPP+	Agent 2	AUROC	0.698	**0.742**	0.584	**0.593**
		PRR	0.343	**0.371**	0.157	**0.171**
	Agent 3	AUROC	0.751	**0.764**	0.621	0.620
		PRR	0.434	**0.438**	0.206	**0.226**
MedQA	Agent 2	AUROC	0.812	**0.837**	0.674	**0.711**
		PRR	0.548	**0.581**	0.370	**0.376**
	Agent 3	AUROC	0.789	**0.834**	0.719	**0.765**
		PRR	0.509	**0.568**	0.452	0.435

Given that finding or verifying the topological ordering via standard algorithms (e.g., Kahn's algorithm or Depth-First Search) incurs an execution cost of $\mathcal{O}(|\mathcal{V}| + |\mathcal{E}|)$, the cumulative computational time for tracking uncertainties across the entire MAS trajectory is tightly bounded by:

$$\mathcal{O}(|\mathcal{V}| + |\mathcal{E}|).$$

Furthermore, the auxiliary space complexity is $\mathcal{O}(|\mathcal{V}|)$ to store the scalar uncertainty values $\{r_v\}_{v \in \mathcal{V}}$, plus $\mathcal{O}(|\mathcal{V}|+|\mathcal{E}|)$ to maintain the underlying graph adjacency lists and edge-level interaction parameters. Consequently, both the time and space overheads scale linearly with the scale of MAS execution, completing the proof of Proposition 3.

C Supplementary Experimental Setup

C.1 Benchmark Descriptions

GSM8K. GSM8K (Cobbe et al., 2021) contains 8.5K grade-school math word problems designed to evaluate multi-step numerical reasoning. Each problem requires translating a natural-language question into a sequence of arithmetic operations and producing a final numeric answer.

MedQA. MedQA (Yang et al., 2025b) consists of medical licensing exam questions that test biomedical knowledge, clinical reasoning, and diagnostic decision-making. The benchmark requires models to integrate textual context with domain-specific medical knowledge under a multiple-choice setting.

MBPP-Plus. MBPP-Plus (Liu et al., 2023) extends the original MBPP benchmark with additional test cases and stricter execution-based evaluation. Each problem asks the model to generate a self-contained Python function that satisfies a functional specification.

C.2 MAS Structure Details

We provide additional details for the three MAS interaction structures used in our experiments.

Sequential MAS. In the **sequential MAS**, we adopt a chain-of-agents design (Zhang et al., 2024; Zhao et al., 2025a) with four role-specialized agents: planner, critic, refiner, and solver. The planner first receives the input question, and each downstream agent processes the reasoning output of its immediate predecessor.

Hierarchical MAS. In the **hierarchical MAS**, we follow a domain-specialized design (Zhao et al., 2026; Zhuge et al., 2024), where code, math, and science agents independently reason from different domain perspectives, and a summarizer agent aggregates their intermediate responses to produce the final answer.

Decentralized MAS. In the **decentralized MAS**, agents have no predefined roles and communicate through a peer-to-peer mesh (Du et al., 2024; Liang et al., 2024). Each agent can condition on the input question and messages from the other agents, allowing us to evaluate uncertainty propagation under dense, role-free interactions.

C.3 Evaluation Metrics

Let $r_v \in [0,1]$ be the uncertainty score and $e_v \in \{0,1\}$ be the ground-truth error indicator ($e_v = 1$ for incorrect outputs) for node $v \in \mathcal{V}_{\text{eval}}$.

AUROC. AUROC quantifies the probability that an incorrect node is ranked higher than a correct one. Let $\mathcal{P} = \{v : e_v = 1\}$ and $\mathcal{N} = \{v : e_v = 0\}$:

$$\text{AUROC} = \frac{1}{|\mathcal{P}||\mathcal{N}|} \sum_{v\in\mathcal{P}} \sum_{w\in\mathcal{N}} \Big[\mathbf{1}(r_v > r_w) + \frac{1}{2}\mathbf{1}(r_v = r_w)\Big].$$

Prediction Rejection Ratio (PRR). PRR evaluates uncertainty-informed *selective prediction*. By sorting nodes in descending order of r_v, we compute the Area Under the Accuracy-Rejection Curve (AURC), denoted as AUC_{uq}. To ensure a metric independent of the base accuracy, PRR normalizes this area against random and oracle baselines:

$$\text{PRR} = \frac{\text{AUC}_{\text{uq}} - \text{AUC}_{\text{rand}}}{\text{AUC}_{\text{oracle}} - \text{AUC}_{\text{rand}}}.$$

Higher AUROC and PRR scores signify superior discriminative power and practical utility for error filtering.

D Additional Experiment Results

D.1 Final-output Results on Qwen3-14B.

Table 5 reports additional final-output UQ results using `Qwen3-14B`. The experiments follow the same datasets, MAS topologies, metrics, and local UQ baselines used in the main evaluation in Section 4.1. The results show that PROPUQ-MAS remains effective at a larger Qwen3 scale, providing further evidence that the proposed propagation rule is not limited to smaller base models.

D.2 Additional Intermediate-Step Results

Table 6 further reports the intermediate-step UQ results on `Qwen3-14B` under the decentralized MAS setting. Following the main intermediate-step evaluation in Section 4.2, we report results for intermediate agents whose outputs can be directly judged against the ground-truth answer. PROPUQ-MAS improves AUROC and PRR in most settings for both MSP and verbalized uncertainty baselines, indicating that the propagation rule also provides effective intermediate-step reliability signals at the larger Qwen3 scale.

Appendix Table 7 further summarizes the averaged intermediate-step results over Agents 2–4. The improvements are consistent across datasets, model families, and local UQ baselines: PROPUQ-MAS improves AUROC in all reported settings and improves PRR in nearly all settings, with especially large PRR gains for verbalized uncertainty. These averaged results confirm that propagation-aware UQ provides stable intermediate-step reliability gains, rather than improvements limited to a single agent or dataset.

D.3 Comparison with Sampling-Based Baselines

We compare PROPUQ-MAS with two concurrent sampling-based UQ baselines, MATU (Chen et al., 2026) and UProp (Duan et al., 2025), on MedQA using Qwen3-8B under sequential and hierarchical

Table 7: Mean uncertainty estimation under decentralized MAS. We report AUROC and PRR averaged over Agents 2–4; superscripts show relative changes for PROPUQ, with green/red indicating gains/drops.

Dataset	Metric	Qwen3-4B				Qwen3-8B				Gemma-3-12B			
		MSP	+PROPUQ	Verb.	+PROPUQ	MSP	+PROPUQ	Verb.	+PROPUQ	MSP	+PROPUQ	Verb.	+PROPUQ
GSM8K	AUROC	0.664	**0.697**$^{+5.0\%}$	0.560	**0.599**$^{+7.0\%}$	0.735	**0.766**$^{+4.1\%}$	0.529	**0.596**$^{+12.8\%}$	0.571	**0.575**$^{+0.6\%}$	0.634	**0.664**$^{+4.7\%}$
	PRR	0.351	**0.423**$^{+20.7\%}$	0.131	**0.210**$^{+60.7\%}$	0.453	**0.524**$^{+15.5\%}$	0.045	**0.209**$^{+361.0\%}$	0.146	**0.166**$^{+13.5\%}$	0.259	**0.285**$^{+10.1\%}$
MBPP+	AUROC	0.710	**0.735**$^{+3.5\%}$	0.595	**0.608**$^{+2.2\%}$	0.697	**0.718**$^{+3.0\%}$	0.522	**0.577**$^{+10.5\%}$	0.628	**0.651**$^{+3.7\%}$	0.666	**0.705**$^{+5.9\%}$
	PRR	0.388	**0.455**$^{+17.1\%}$	0.173	**0.210**$^{+21.6\%}$	0.335	**0.365**$^{+8.8\%}$	0.049	**0.135**$^{+177.4\%}$	0.231	**0.302**$^{+30.8\%}$	0.347	**0.394**$^{+13.5\%}$
MedQA	AUROC	0.767	**0.812**$^{+5.9\%}$	0.679	**0.728**$^{+7.2\%}$	0.795	**0.818**$^{+2.9\%}$	0.693	**0.831**$^{+19.9\%}$	0.612	**0.646**$^{+5.5\%}$	0.584	**0.615**$^{+5.4\%}$
	PRR	0.459	**0.543**$^{+18.5\%}$	0.334	**0.348**$^{+4.3\%}$	0.492	**0.539**$^{+9.6\%}$	0.386	**0.566**$^{+46.7\%}$	0.204	**0.239**$^{+17.1\%}$	0.209	**0.229**$^{+9.6\%}$

Table 8: Comparison with MATU and UProp on MedQA using Qwen3-8B. Time is in seconds per example, and UProp applies only to sequential MAS. Bold marks the best uncertainty metric within each topology.

Metric	Sampling-Based		MSP		Verb.	
	MATU	UProp	w/o	+PROPUQ	w/o	+PROPUQ
Sequential MAS						
AUROC	0.718	0.650	0.754	**0.842**	0.600	0.791
PRR	0.373	0.330	0.496	**0.624**	0.217	0.482
Time/sample (s)	41.410	11.310	1.129	1.328	4.059	4.258
Hierarchical MAS						
AUROC	0.626	–	0.658	0.711	0.604	**0.722**
PRR	0.263	–	0.277	0.349	0.223	**0.362**
Time/sample (s)	56.040	–	1.536	1.735	5.522	5.721

MAS. UProp is designed for single-agent trajectories and can therefore be adapted only to the sequential setting. At the time of our experiments, official implementations were unavailable; we therefore implemented both baselines following their original papers.

Table 8 shows that PROPUQ-MAS achieves the strongest uncertainty estimation in both topologies: it outperforms MATU and UProp on sequential MAS, and both propagated variants outperform MATU on hierarchical MAS. Moreover, PROPUQ-MAS is substantially more efficient than sampling-based baselines, as it introduces only a lightweight propagation layer during standard MAS execution. These results confirm that PROPUQ-MAS effectively provides robust, online reliability signals with minimal computational overhead.

D.4 PRR results for scale generalization

Figure 6 reports the PRR results for the scale generalization analysis in Section 4.3. We vary the chain length in sequential MAS and the expert-layer width in hierarchical MAS. Across GSM8K and MBPP-Plus, PROPUQ-MAS consistently improves over the verbalized uncertainty baseline, showing that the proposed propagation rule remains useful when the MAS trajectory becomes longer or when an aggregation node receives more parent inputs. These results complement the AUROC results in the main text and further support the scalability of PROPUQ-MAS under both deeper and wider MAS structures.

D.5 Sequential MAS Case Study

Figure 7 presents a GSM8K case under the sequential MAS topology. While the hierarchical case in Section 4.6 illustrates selective aggregation over multiple upstream branches, this example focuses on a different failure mode: an error introduced at one intermediate step can be successively adopted and carried forward along a reasoning chain.

In this example, the planner first produces a correct plan. The critic then introduces an alternative but incorrect interpretation of the problem, which is strongly adopted by the refiner. The final solver follows the refined plan and outputs an incorrect answer despite having low local uncertainty. This shows that a downstream agent may appear reliable from its own local perspective while still inheriting an earlier reasoning error.

PROPUQ-MAS identifies this failure by propagating uncertainty along the accepted dependencies in the sequential chain. As the incorrect interpretation is repeatedly accepted by later agents, the

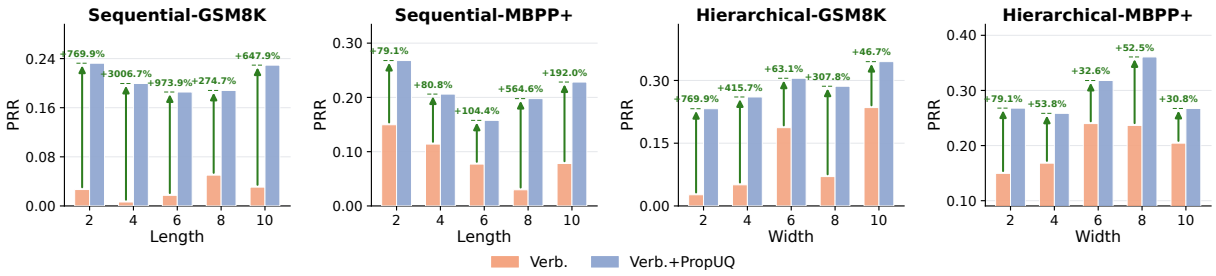

Figure 6: Structural scale generalization results measured by PRR. We evaluate the robustness of uncertainty propagation across varying sequential chain lengths and hierarchical expert-layer widths.

final output receives high propagation-aware uncertainty. This complementary case demonstrates that PROPUQ-MAS can capture not only selective risk aggregation in hierarchical MAS, but also chain-wise risk transmission in sequential MAS.

E Prompt Templates

This appendix provides the prompt templates used in our experiments.

E.1 MAS Agent Response Prompts

Figure 8 and Figure 9 show the role-specific prompts used for sequential and hierarchical MAS execution.

E.2 Acceptance Probability Elicitation Prompt

Figure 10 shows the prompt template used to elicit the self-reported acceptance score $\hat{\alpha}_{pv}$. This score serves as the plug-in estimate of the edge-level acceptance probability α_{pv} in online inference.

E.3 Prompt for Scale Generalization Analysis

In the scale generalization analysis, we vary the number of agents while keeping their behavior comparable across scales. Each agent follows the same critic-solver instruction: it verifies incoming answers, provides feedback, and then produces its own answer. This controlled design allows us to change the MAS depth or width without introducing additional role-specific effects. Figure 11 shows the prompt used in this experiment.

E.4 LLM-as-a-Judge Prompt for Intermediate-Step Evaluation

For the intermediate-step evaluation in the sequential MAS, GPT-5.5 judges whether each Critic or Refiner output contains a substantive medical or reasoning error. Figure 12 shows the prompt template.

Additional Case Study from GSM8K under Sequential MAS

Question. There are 36 penguins sunbathing. One-third swim away, and another one-third go inside a cave. How many penguins are still sunbathing?

Gold Answer: 12 **Final Prediction:** 16 (incorrect)

Planner. *Plan:* start with 36 penguins; compute one-third of 36 as 12 swimmers; compute another one-third of 36 as 12 cave-going penguins; subtract both groups from 36, giving $36 - 12 - 12 = 12$.

Local uncertainty: 0.15
Propagation-aware uncertainty: 0.15

↓ Planner to Critic: acceptance = 0.60

Critic. *I think the plan may be incomplete:* "another one-third" could refer to one-third of the remaining 24 penguins rather than one-third of the original 36. Under this interpretation, $24/3 = 8$, leading to $24 - 8 = 16$.

Local uncertainty: 0.75
Propagation-aware uncertainty: 0.773

↓ Critic to Refiner: acceptance = 0.95

Refiner. *Refined plan:* start with 36; subtract 12 swimmers to get 24 remaining; then take one-third of the 24 as 8 cave-going penguins; therefore $24 - 8 = 16$.

Local uncertainty: 0.20
Propagation-aware uncertainty: 0.787

↓ Refiner to Final Solver: acceptance = 0.90

Final Solver. *I follow the Refiner's plan:* after 12 penguins swim away, 24 remain; taking one-third of 24 gives 8, so the final answer is $24 - 8 = 16$. The steps seem internally consistent, so my local uncertainty is low.

Local uncertainty: 0.10
Propagation-aware uncertainty: 0.738

Figure 7: Additional sequential MAS case study from GSM8K. The planner initially gives a correct plan, but the critic introduces an incorrect interpretation that is strongly adopted by later agents. Although the final solver has low local uncertainty, PROPUQ-MAS assigns high propagation-aware uncertainty by tracking the accepted upstream risk.

Sequential MAS Role Prompts

Shared input:

```
## Input Question: {QUESTION}
```

Planner Agent:

```
You are a Planner Agent. Given the input question, design a concise step-by-step plan for solving
it. Do not produce the final answer.
## Output Format:
Planner Agent's Output: [Your plan here]
```

Critic Agent:

```
You are a Critic Agent. You are given the input question and the Planner Agent's plan. Evaluate
whether the plan is correct and complete, and provide helpful feedback.
## Plan from Planner Agent: {PLANNER_OUTPUT}
## Output Format:
Critic Agent's Output:
Original Plan: [Copy the provided plan here]
Feedback: [Your feedback here]
```

Refiner Agent:

```
You are a Refiner Agent. You are given the input question, the Planner Agent's plan, and the
Critic Agent's feedback. Produce an improved step-by-step plan.
## Original Plan and Critic Feedback: {PLANNER_AND_CRITIC_OUTPUT}
## Output Format:
Refiner Agent's Output: [Your refined plan here]
```

Solver Agent:

```
You are the final Solver Agent in a sequential MAS: Planner → Critic → Refiner → Solver. You are
given the input question and the Refiner Agent's plan. The plan may contain irrelevant or incorrect
content; ignore it if it is not useful.
## Refined Plan: {REFINER_OUTPUT}
Reason step by step and output the final answer inside \boxed{YOUR_FINAL_ANSWER}.
```

Figure 8: Role-specific prompt template used for the sequential MAS.

Hierarchical MAS Role Prompts

Shared input:

```
## Input Question: {QUESTION}
```

Math Agent:

```
You are a math agent. Solve the input question from a mathematical reasoning perspective. Output
the final answer inside \boxed{YOUR_FINAL_ANSWER}.
```

Science Agent:

```
You are a science agent. Solve the input question from a scientific reasoning perspective. Output
the final answer inside \boxed{YOUR_FINAL_ANSWER}.
```

Code Agent:

```
You are a code agent. Solve the input question using programming or algorithmic reasoning when
useful. Output the final answer inside \boxed{YOUR_FINAL_ANSWER}.
```

Task Summarizer:

```
You are a task summarizer. You are given the input question and the responses from the Math,
Science, and Code agents. Synthesize their outputs and produce the final answer.
## Previous Agent Outputs: {MATH_SCIENCE_CODE_OUTPUTS}
Output the final answer inside \boxed{YOUR_FINAL_ANSWER}.
```

Figure 9: Role-specific prompt template used for the hierarchical MAS.

Self-Reporting Prompt for $\hat{\alpha}_{pv}$ Elicitation

```
For each incoming agent, report message_adoption_weight as a number in [0,1].

  • A higher value means your response largely accepts or reuses that agent's message.
  • A lower value means your response critiques, revises, rejects, or only weakly relies on that
    message.

## Output one line for each directly connected incoming agent:
<message_adoption agent="{INCOMING_AGENT_NAME}" score="FLOAT_0_TO_1"/>
```

Figure 10: Prompt template used to elicit the self-reported acceptance score $\hat{\alpha}_{pv}$ after an agent response is generated.

Critic-Solver Prompt for Scale Generalization

```
Given the input question and answers from directly connected previous agents, independently verify
each incoming answer, decide whether you agree, and identify what should be improved.  If you
disagree, provide an alternative solution. Then provide feedback and your own answer.

{ANSWER_FORMAT_INSTRUCTION}

 ## Output Format
 ## Feedback
- Agent X: [State whether you agree, what is correct, and what should be improved]
```

Figure 11: Prompt template used in the scale generalization analysis, where each agent acts as both critic and solver.

LLM-as-a-Judge Prompt for Intermediate-Step Evaluation

System Message

```
You are a strict but fair expert judge for medical question answering.
Return exactly one valid JSON object and no additional text. /no_think
```

User Message

```
Evaluate the specified intermediate MAS node using the task, reference answer,
and preceding node outputs below. Assess the target node itself rather than
inferring its quality solely from the final MAS prediction.

### Task
Question: {TASK_QUESTION}
Gold answer option: {GOLD_ANSWER_OPTION}
Final MAS prediction: {FINAL_MAS_PREDICTION}
### Previous Intermediate Context
{PREVIOUS_INTERMEDIATE_CONTEXT}
### Target Node
Role: {NODE_ROLE}
Output: {NODE_OUTPUT}
### Role-Specific Criterion
{ROLE_SPECIFIC_JUDGING_INSTRUCTION}

### Label Definitions
```

- "helpful": medically valid and useful, without a substantive error.
- "mixed": useful but incomplete, ambiguous, or affected by a limited issue.
- "harmful": contains a substantive medical or reasoning error likely to mislead downstream agents.

```
Set "node_error" to true if and only if the label is "harmful"; otherwise, set it to false.
Return exactly one JSON object with only "label", "node_error", and "rationale".
Do not include Markdown fences or any other text. /no_think
```

Role-Specific Criteria

```
Critic: Determine whether the Critic gives a medically valid critique of the preceding
plan. It should identify genuine issues or appropriately confirm correct reasoning
without introducing a substantive error.
Refiner: Determine whether the Refiner produces a medically valid revised plan. It
should preserve or improve the preceding reasoning without introducing a substantive error.
```

Placeholder Definitions

- {TASK_QUESTION}: complete multiple-choice medical question, including answer options.
- {GOLD_ANSWER_OPTION}: reference answer option; {FINAL_MAS_PREDICTION}: final MAS answer.
- {PREVIOUS_INTERMEDIATE_CONTEXT}: preceding outputs in execution order; use [none] when unavailable.
- {NODE_ROLE}: target role (Critic or Refiner); {NODE_OUTPUT}: target output.
- {ROLE_SPECIFIC_JUDGING_INSTRUCTION}: criterion for the target role.

Figure 12: LLM-as-a-judge prompt for labeling Critic and Refiner outputs in the sequential MedQA evaluation.