UNDERSTANDING AND ESTIMATING UNCERTAINTY FLOW IN MULTI-AGENT SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-agent systems solve tasks by exchanging messages, and each message changes what remains uncertain about the final outcome, indicating uncertainty as the floor on how well failure can be predicted at that point. Predicting failure is therefore an uncertainty estimation problem over the whole communication record. Though recent studies show that errors and hallucinations propagated in the communication could cause the whole system to fail, they mainly focus on the final outputs, instead of capturing the inherent consensus that happens in a multi-agent system. To address these limitations, we first develop a theoretical framework of how uncertainty about the outcome evolves through communication. By applying standard martingale identities to the filtration generated by the messages, we assign each message a resolution without assuming any dependence structure among agents, and show that the same identities hold for the features observed by an estimator. Motivated by our theoretical findings, we propose CUE, a supervised probe on backbone hidden states that estimates correctness and its uncertainty after every message and attributes the resolved uncertainty to rounds. We evaluate CUE on whether a multi-agent system's final answer is correct, across 4 benchmarks, 3 multi-agent orchestrations, and 2 LLM families, against 7 baselines. CUE achieves the highest AUROC in most of the configurations on both Llama and Qwen.

1 INTRODUCTION

LLM-based multi-agent systems solve tasks by coordinating through communication, tool use, and environmental feedback (Minaee et al., 2024; Wang et al., 2024; Chang et al., 2023; Raiaan et al., 2024). Communication is what makes them work, and it is also how they fail: errors and overconfident beliefs propagate from agent to agent and can be amplified until the whole system fails (Pitre et al., 2025; Wynn et al., 2025). Figure 1 shows why this is hard to detect from the output alone. In a debate, the uncertainty of individual messages falls and rises as the agents exchange answers, and by the final message a wrong consensus can look as certain as a right one, although the difference between them emerged message by message. Uncertainty about the final outcome is therefore not a fixed property of the task but a quantity that every message revises, and predicting whether a multi-agent execution will fail is an uncertainty estimation problem over its whole communication record.

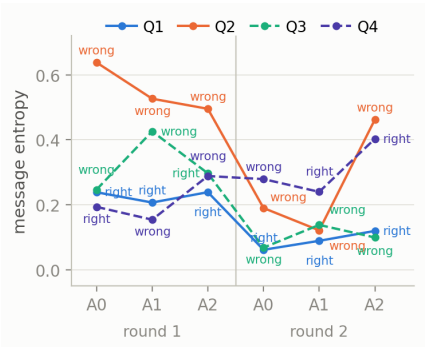

Figure 1: Per-message entropy along a two-round, three-agent MATH500 debate for four queries. Entropy falls and rises within a debate regardless of correctness, and the entropy of the final message does not separate the correct runs (Q1, Q4) from the failed ones (Q2, Q3).

Uncertainty estimators for single LLMs score the final generation, through token-level probabilities (Gal & Ghahramani, 2016; Gal et al., 2017; Malinin & Gales, 2021), agreement among semantically equivalent samples (Kuhn et al., 2023; Duan et al., 2024), self-assessment (Kadavath et al., 2022; Yin et al., 2023), or supervised probes on the model's outputs and representations (Liu et al., 2024; Yaldiz et al., 2025). They assume that the generation being scored is the whole computation. In a multi-agent execution, the delivered answer comes at the end of a chain of exchanges, and by then the agents

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

have usually converged: the estimator reads the consensus, not the disagreement that preceded it. Multi-agent methods do use intermediate outputs. Han et al. (2024) and Zhao et al. (2025) aggregate per-agent uncertainty over an execution, and Liu et al. (2026) propagate it over the communication graph. Each reduces an agent to one scalar and combines the scalars through an assumed dependence structure, so what the messages say, and how much each message resolved about the outcome, is lost. Process reward models (Xi et al., 2026) and runtime monitors (Custers & Aslansefat, 2027) score partial traces, but they target single-agent reasoning chains, need a label for every step, or summarize token likelihoods without representing what the messages say. The limitation of focusing only on the terminal state and lack of attention to the communication message motivate two research questions: **RQ1**: *What is the underlying theoretical mechanism governing how uncertainty about the final outcome evolves as agents communicate, and how is uncertainty resolved through communication?* **RQ2**: *How can we develop a computational framework to estimate correctness and its uncertainty throughout a multi-agent execution as communication unfolds?*

To address these two research questions, we propose CUE (Communication-aware Uncertainty Estimation), an algorithm motivated by the theoretical analysis. Particularly, we develop *Communication Variance Flow* (Section 3) to offer a principled understanding of how communication messages influence correctness and uncertainty to address **RQ1**. Our theoretical analysis reveals how uncertainty evolves through multi-agent communication by tracking the conditional variance of terminal correctness along the message-induced filtration. Applying standard martingale identities at the message level yields three key findings without assuming any dependence structure among agents. First, communication resolves uncertainty in expectation, with each message contributing an expected variance resolution. Second, uncertainty need not decrease along an individual trajectory: a message can instead increase uncertainty, a phenomenon we characterize as realized uncertainty amplification. Third, the total resolved uncertainty admits an exact message-level decomposition that aggregates across rounds, senders, and channels, enabling information-resolution attribution. This quantifies what each message reveals to an observer rather than its causal effect on the outcome.

Motivated by these theoretical findings, we propose CUE, a computational framework for estimating uncertainty throughout multi-agent communication. CUE continuously estimates correctness from the accumulated communication history after each message, using a single supervised probe trained only on final-answer correctness. From these estimates, it derives residual uncertainty, message-level resolution, and realized uncertainty amplification within a unified framework, without requiring separate estimators or intermediate supervision. Unlike existing supervised probes that score only the final output (Liu et al., 2024; Yaldiz et al., 2025), CUE tracks how uncertainty evolves as communication unfolds, enabling early detection of executions heading toward failure.

We conduct extensive experiments to demonstrate the effectiveness of CUE across four coding and question-answering benchmarks, three multi-agent orchestrations (SelfPlanner, GPT-Swarm, and multi-agent debate), and two backbone families (Llama-3.1-70B and Qwen at 30–32B). CUE outperforms seven competitive baselines in most configurations, improving the average AUROC from 70.5 to 81.8 on Llama and from 71.4 to 82.8 on Qwen. Moreover, CUE enables effective early failure detection: on Llama debate, its uncertainty estimate after only the first communication round already outperforms the strongest baselines evaluated on the complete debate in most configurations.

2 PRELIMINARIES

In this section, we provide the basic notations and setups in our paper. The capital letters denote random variables. The lowercase letters denote their realized values.

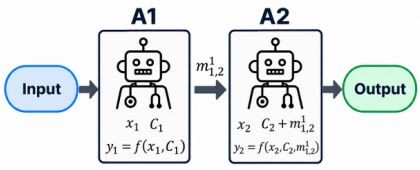

Figure 2: A two-agent exchange. Messages are generated in a fixed order, and it is that order, not the round grouping, that defines the sequence of prefixes conditioned on in Section 3.

Multi-agent systems. Let $\mathcal{X}$ and $\mathcal{Y}$ denote the input and output spaces, and let $\mathcal{M}$ denote the space of messages the agents may exchange, with $\mathcal{Y} \subseteq \mathcal{M}$ so that a delivered answer is itself a message. We consider a system of K agents $\mathcal{V} = \{A_1, \ldots, A_K\}$ communicating over a potentially time-varying directed graph $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t)$, $t = 1, \ldots, T$. All agents share a pretrained LLM with fixed parameters θ, which induces a message distribution $P_\theta(\cdot \mid C)$ over $\mathcal{M}$ for a context C.

Queries are drawn from a population, $X_\star \sim \mathcal{P}_\mathcal{X}$, with $x_\star$ a realization, and the query determines the target response $y_\star \in \mathcal{Y}$. The target is never observed at inference time and serves only to define the terminal outcome $Z^\star$ that Section 3 analyzes, which Equation (10) instantiates as a correctness margin read off a fixed benchmark evaluator. Population quantities are taken over $\mathcal{P}_\mathcal{X}$ and the system's own decoding randomness, so they depend on the temperature at which the system runs.

Execution proceeds over rounds $t = 1, \ldots, T$. Round t activates the edge set $\mathcal{E}_t$, which we write as

$$\mathcal{E}_t = (e_{t,1}, \ldots, e_{t,L_t}), e_{t,r} = (j_{t,r}, i_{t,r}), \tag{1}$$

where $L_t = |\mathcal{E}_t|$ is the number of directed communications in round t and $e_{t,r}$ is the r-th of them, sent by $A_{j_{t,r}}$ to $A_{i_{t,r}}$. The content of the message is denoted as $m^{(t)}_{e_{t,r}}$.

Messages are generated sequentially. At step (t, r) the sender samples from P_θ conditioned on its own context; the receiver appends the message to its context through $\mathrm{agg}(\cdot)$, and the contexts of agents not receiving it are unchanged. A receiver therefore updates before any later message that depends on it is generated (Figure 2), and rounds only group these exchanges for indexing. We use a receiver-only memory, in which a sender does not append its own message unless it is also a recipient; the convention is part of the fixed configuration used to collect the corpus, and the identities below hold under either convention for the execution law it induces. Parallel debate fits the same framework once an observation order is declared for its simultaneous messages.

Communication therefore forms an ordered trajectory:

$$\mathcal{T} = \left(e_{1,1}, m^{(1)}_{e_{1,1}}, \ldots, e_{1,L_1}, m^{(1)}_{e_{1,L_1}}, \ldots, e_{T,L_T}, m^{(T)}_{e_{T,L_T}}\right), \tag{2}$$

in which a message can affect later generations through context updates. The record contains the delivered answer $a^\star \in \mathcal{Y}$, the system's final response to $x_\star$, exactly once, as its last entry. If the answer is not delivered by an inter-agent message, we append it as a labeled output event with the same (t, r) indexing and a designated output recipient, and an external aggregator gets its own sender label. Output events extend the record but do not change the context dynamics above.

Problem 2.1 (Failure Prediction from a Communication Record)**.**

Given: (i) a shared query $x_\star$; (ii) a set of agents $\mathcal{V}$ communicating through $\{\mathcal{G}_t\}_{t=1}^T$; and (iii) a fixed backbone LLM P_θ.

Find: an estimate of the uncertainty in the multi-agent system that accounts for the uncertainty along the communication trajectory and its compounding during communication.

3 Communication Variance Flow

In this section, we address **RQ1** by investigating the underlying theoretical mechanism governing how uncertainty about the final outcome evolves and is resolved through multi-agent communication. We first characterize how each round of communication changes uncertainty about the final outcome and decompose the resolved uncertainty across individual messages without assuming any dependence structure among agents. We then extend the analysis to the full communication trajectory, revealing that communication resolves uncertainty in expectation while uncertainty can still increase along individual executions. Together, these results establish a theoretical characterization of uncertainty flow through communication and its resolution across messages, rounds, senders, and channels. Complete proofs are provided in Appendix A.

Consider the sequential communication trajectory defined in Section 2. Round t executes the L_t directed communications $e_{t,1}, \ldots, e_{t,L_t}$ in order, producing messages $M^{(t)}_{e_{t,1}}, \ldots, M^{(t)}_{e_{t,L_t}}$. Let

$$\mathcal{H}_{t,r} = \sigma\left(X_\star, \{M^{(u)}_{e_{u,q}}\}_{u<t,\, q\le L_u}, M^{(t)}_{e_{t,1}}, \ldots, M^{(t)}_{e_{t,r}}\right), \qquad r = 0, \ldots, L_t, \tag{3}$$

denote the query and messages observed through step (t, r). We write $\mathcal{H}_t := \mathcal{H}_{t,L_t}$, so that $\mathcal{H}_{t,0} = \mathcal{H}_{t-1}$ and $\mathcal{H}_0 = \sigma(X_\star)$, and communication induces the nested filtration $\mathcal{H}_0 \subseteq \mathcal{H}_{1,1} \subseteq \cdots \subseteq \mathcal{H}_1 \subseteq \cdots \subseteq \mathcal{H}_T$.

Let $Z^\star \in \mathbb{R}$ describe the final system outcome, with $\mathbb{E}[(Z^\star)^2] < \infty$. We define $\mu_t = \mathbb{E}[Z^\star \mid \mathcal{H}_t], V_t = \mathrm{Var}(Z^\star \mid \mathcal{H}_t)$, and analogously $\mu_{t,r}$ and $V_{t,r}$ at message resolution. Under squared-error

loss, we can have $V_t = \inf_a \mathbb{E}[(Z^\star - a)^2 \mid \mathcal{H}_t]$, the infimum is over $\mathcal{H}_t$-measurable decisions and attained at $a = \mu_t$, so V_t is the smallest error any predictor of the outcome can attain after the communication observed up to round t.

3.1 Uncertainty Resolution through Communication

This subsection characterizes how communication changes the uncertainty about the final outcome, as seen by an observer of the messages. Using μ_t and V_t, we first give the uncertainty that one round of communication resolves.

Theorem 3.1 (Communication Variance Flow)**.** *For any $Z^\star$ with finite second moment,*

$$V_{t-1} = \mathbb{E}[V_t \mid \mathcal{H}_{t-1}] + \underbrace{\mathbb{E}[(\mu_t - \mu_{t-1})^2 \mid \mathcal{H}_{t-1}]}_{\Gamma_t}. \tag{4}$$

We call Γ_t the *variance resolution* contributed by round-t communication; equivalently $\Gamma_t = V_{t-1} - \mathbb{E}[V_t \mid \mathcal{H}_{t-1}] \geq 0$. Theorem 3.1 separates the expected value of information from the realized change in uncertainty: more communication cannot increase the optimal Bayes risk in expectation, $\mathbb{E}[V_t \mid \mathcal{H}_{t-1}] \leq V_{t-1}$, but this does *not* mean $V_t \leq V_{t-1}$ for every realized message. How much a round resolves also depends on the trajectory so far. In particular, if the messages sent in round t carry no new task-relevant information, $Z^\star \perp \left(M^{(t)}_{e_{t,1}}, \ldots, M^{(t)}_{e_{t,L_t}}\right) \mid \mathcal{H}_{t-1}$, then $\Gamma_t = 0$. Repeated or redundant messages resolve nothing once the earlier trajectory already contains the information.

The round-level quantity splits further across the directed communications of the round, using the message-level filtration $\mathcal{H}_{t,r}$ of Equation (3) and the corresponding posterior means $\mu_{t,r}$. We can get:

Proposition 3.2 (Communication Decomposition)**.** *For every round t,*

$$\Gamma_t = \sum_{r=1}^{L_t} \mathbb{E}\left[(\mu_{t,r} - \mu_{t,r-1})^2 \mid \mathcal{H}_{t-1}\right]. \tag{5}$$

Each directed message thus contributes the belief resolution it adds given the communication already observed. The decomposition needs no independence assumption between agents or messages, because conditioning on the earlier trajectory discounts overlapping information automatically. It is stated for the fixed message order of Equation (1), so it attributes communication value sequentially under that order; it is not a unique causal allocation to individual messages. Write

$$\Gamma_{t,r} = \mathbb{E}\big[(\mu_{t,r} - \mu_{t,r-1})^2 \mid \mathcal{H}_{t,r-1}\big] \tag{6}$$

for the resolution contributed by the single communication (t, r), so that Proposition 3.2 reads $\Gamma_t = \sum_{r=1}^{L_t} \mathbb{E}[\Gamma_{t,r} \mid \mathcal{H}_{t-1}]$. For concurrent messages, we fix one serialization order and report it. Grouping these per-message resolutions by the labels the trajectory already carries gives exhaustive additive shares of the resolved total, which Appendix A.7 states together with the caveats it carries. The same filtration governs the whole conditional law of $Z^\star$ which Corollary A.1 records.

3.2 Uncertainty Flow over Communication Trajectories

We now extend the analysis from one round to the whole trajectory, relating the uncertainty before any message is sent to the uncertainty left at the end. Applying Theorem 3.1 gives the following:

Corollary 3.3 (Trajectory-Level Variance Flow)**.** *After T rounds,*

$$V_0 = \mathbb{E}[V_T \mid \mathcal{H}_0] + \sum_{t=1}^{T} \mathbb{E}[\Gamma_t \mid \mathcal{H}_0], \tag{7}$$

and, pathwise, with $\xi_t := V_t - \mathbb{E}[V_t \mid \mathcal{H}_{t-1}]$,

$$V_T = V_0 - \sum_{t=1}^{T} \Gamma_t + \sum_{t=1}^{T} \xi_t. \tag{8}$$

Equation (7) splits the initial task uncertainty into the uncertainty left after communication and the uncertainty resolved over the sequence of exchanges. With Proposition 3.2, this decomposes the flow from messages to rounds to the full trajectory.

The decomposition holds in expectation; it is not a pathwise conservation law. Each ξ_t in Equation (8) satisfies $\mathbb{E}[\xi_t \mid \mathcal{H}_{t-1}] = 0$, and the same identities hold at the message level. Expected contraction does not make uncertainty monotone along an observed trajectory. For the realized change $\Delta_t^{\text{real}} = V_t - V_{t-1}$,

$$\mathbb{E}[\Delta_t^{\text{real}} \mid \mathcal{H}_{t-1}] = -\Gamma_t \leq 0, \tag{9}$$

while individual rounds can have $\Delta_t^{\text{real}} > 0$. We call these events *realized uncertainty amplification* and write their cumulative size over a trajectory as $\mathcal{A}_{1:T} = \sum_{t=1}^{T}(V_t - V_{t-1})_+, (x)_+ = \max(x, 0)$. Communication can therefore help in expectation and still produce episodes, specific to a trajectory, in which uncertainty rises and is carried into later interactions. Appendix A.6 records how the same count is read at message granularity. Together, Equation (5) and the attribution identity of Appendix A.7 show every message carries a resolution that is defined without a model of inter-agent dependence, and those resolutions sum exactly over any grouping of the trajectory.

4 The CUE Framework

This section presents CUE. We first introduce a communication-aware prefix representation that captures the accumulated communication history (Section 4.1). We then establish feature-relative uncertainty flow, extending the theoretical quantities in Section 3 to the representations observed by CUE while preserving the underlying identities exactly (Section 4.2). Finally, we develop an outcome-supervised learning framework that estimates correctness and derives uncertainty and communication resolution throughout the execution using only final-answer correctness labels (Section 4.3).

4.1 Communication-Aware Prefix Representation

We run the multi-agent system unchanged and record its directed messages. Let $\mathcal{I}$ be the set of directed communications (t, r), ordered as in Equation (2); enumerate it as $\iota_1 \prec \cdots \prec \iota_M$ with $M = |\mathcal{I}|$, and write $\mathcal{H}_\ell := \mathcal{H}_{t_\ell, r_\ell}$, so ℓ runs over messages in the order of Section 3. The recorded prefix Π_ℓ after message ι_ℓ serializes the query and then every history message $\iota_1, \ldots, \iota_\ell$ in order, each tagged with its round, operation, sender, and receiver, and Π_0 is the query alone. The prefix therefore includes the whole communication history up to ι_ℓ, not only the latest message. It is the record of an observer, not the context of any agent: an agent generating a message conditions only on the query and the messages it receives (in debate, agents of the same round do not see one another's messages), whereas Π_M holds every message of every round together with the delivered answer. Serialization keeps the full message text and the labels of the fixed public schema, so $\sigma(\Pi_\ell) = \mathcal{H}_\ell$. The terminal outcome is the correctness margin:

$$Z^\star = 2C^\star - 1 \in \{-1, 1\}, C^\star = \mathbb{1}\{\text{final output is correct}\}, \mathrm{Var}(Z^\star \mid \mathcal{A}) = 1 - \mathbb{E}[Z^\star \mid \mathcal{A}]^2 \tag{10}$$

for any information set $\mathcal{A}$, so for binary correctness the conditional variance is a function of the conditional mean. Each prefix is mapped to a fixed-dimensional input:

$$F_\ell = \phi_{\Omega,\varrho}(\Pi_\ell), \tag{11}$$

Here $\phi_{\Omega,\varrho}$ is the hidden state of the frozen backbone P_θ, pooled over the tokens of Π_ℓ with rule $\varrho \in \{\text{mean}, \text{last}, \text{max}\}$, ℓ_2-normalized, and averaged over a fixed set Ω of layers; prefixes longer than the token budget are truncated from the left (Appendix C.2). Computing F_ℓ takes one forward pass of the backbone over the recorded Π_ℓ, with all of its history messages, and never changes the execution.

4.2 Feature-Relative Uncertainty Flow

With the representation F_ℓ in hand, we now derive the constraint, motivated by Section 3, that the estimator should satisfy along the trajectory. Let $\mathcal{H}_\ell^F$ be the information carried by the features observed up to message ℓ, and define the feature-relative counterparts of the quantities in Section 3:

$$\mathcal{H}_\ell^F = \sigma(F_0, \ldots, F_\ell), m_\ell = \mathbb{E}[Z^\star \mid \mathcal{H}_\ell^F], v_\ell = \mathrm{Var}(Z^\star \mid \mathcal{H}_\ell^F), \gamma_\ell = \mathbb{E}[(m_\ell - m_{\ell-1})^2 \mid \mathcal{H}_{\ell-1}^F]. \tag{12}$$

These are the analogues of $\mu_{t,r}$, $V_{t,r}$, and $\Gamma_{t,r}$ (Equation (6)) at message ι_ℓ. By construction $\mathcal{H}_0^F \subseteq \cdots \subseteq \mathcal{H}_M^F$, and $\mathcal{H}_\ell^F \subseteq \mathcal{H}_\ell$ because each F_j is a function of Π_j.

Because $\mathcal{H}^F$ is a filtration, $\{m_\ell\}$ is a martingale with respect to it, and for every $\ell \geq 1$, $v_{\ell-1} = \mathbb{E}[v_\ell \mid \mathcal{H}_{\ell-1}^F] + \gamma_\ell, v_0 = \mathbb{E}[v_M \mid \mathcal{H}_0^F] + \sum_{\ell=1}^M \mathbb{E}[\gamma_\ell \mid \mathcal{H}_0^F]$. This is Theorem 3.1 and Corollary 3.3 written for the feature history. The results of Section 3 require only that the conditioning sets form a filtration and that $Z^\star$ is square integrable, so Proposition 3.2, the pathwise identity Equation (8), and Proposition A.2 also hold with $(m_\ell, v_\ell, \gamma_\ell, \mathcal{H}_\ell^F)$ in place of $(\mu_{t,r}, V_{t,r}, \Gamma_{t,r}, \mathcal{H}_{t,r})$ (Appendix B.1). This demonstrates that every quantity CUE fits has a well-defined population target, corresponding to the identities of Section 3. CUE estimates m_ℓ and v_ℓ and reads off the realized resolution $\widehat{J}_\ell$ (Section 4.3), whose expectation matches γ_ℓ up to the error bounded in Proposition B.1; these describe what the observed representation has resolved about the outcome.

4.3 Learning Objective

Section 4.1 gives the representation F_ℓ of each prefix, and Section 4.2 gives the constraint that the theory places on the uncertainty it carries: the conditional variance v_ℓ cannot rise in expectation as messages arrive. We now combine the two into the learning objective that trains the estimator. CUE fits a single feedforward network on F_ℓ by penalized least squares, with the loss pooled over all positions so that one network scores every prefix; the penalty is the constraint of Section 4.2. It is thus a supervised probe on backbone hidden states, in the same family as probes of a single generation (Azaria & Mitchell, 2023; Kossen et al., 2024), with two differences: it reads the communication record at every prefix rather than only the final output, and the theory fixes three of its components, namely the regression target m_ℓ, the regularizer $\mathcal{R}_\mathrm{c}$, and the readouts $\widehat{J}$ and $\widehat{\mathcal{A}}$ defined below.

Section 3 does not only name m_ℓ but also characterizes it variationally. Under squared error $v_\ell = \inf_a \mathbb{E}[(Z^\star - a)^2 \mid \mathcal{H}_\ell^F]$, with the infimum attained at $a = m_\ell$, so the conditional mean *is* the squared-error minimizer and the residual risk is the value of that minimum. Equation (13) is the empirical form of minimizing pooled squared error over prefixes. Because the network reads F_ℓ, it targets m_ℓ up to the representation error ϵ_ℓ of Appendix B.2.

The training target regresses every prefix on the terminal margin, giving $\widehat{\mu}_\ell = \mathrm{clip}_{[-1,1]}\, f_{\psi_\mu}(F_\ell)$ with

$$\psi_\mu = \arg\min_\psi \; \frac{1}{N(M+1)} \sum_{i=1}^{N} \sum_{\ell=0}^{M} \big(f_\psi(F_\ell^{(i)}) - Z_i^\star\big)^2 \; + \; \lambda_\mathrm{c} \mathcal{R}_\mathrm{c}(\psi), \tag{13}$$

$$\mathcal{R}_\mathrm{c} = \frac{1}{M} \sum_{\ell=1}^{M} \underbrace{\big[\overline{\widehat{V}_\ell} - \overline{\widehat{V}_{\ell-1}}\big]_+^2}_{\text{average rise in uncertainty}} \quad . \tag{14}$$

History enters through the input: F_ℓ encodes the recorded prefix Π_ℓ, the observer's record of the query and every message up to ι_ℓ, not the final agent's context (Section 4.1). Since $f_\psi(F_\ell)$ is a function of $F_0, \ldots, F_\ell$, Appendix B.2 applies, and whatever F_ℓ loses of the earlier prefixes, e.g. through truncation or pooling, enters its error ϵ_ℓ. In Equation (14), $\widehat{V}_\ell = 1 - \widehat{\mu}_\ell^2$ is the residual uncertainty and $\bar{\cdot}$ averages over the training executions in a batch. $\mathcal{R}_\mathrm{c}$ penalizes a rise in this average, i.e. the observer growing less certain by reading more, which Theorem 3.1 rules out in expectation. A rise on a single trajectory is not penalized: it is the realized amplification that Equation (9) allows and that Section 5.5 finds on 41–51% of executions. λ_c is selected on validation, and $\lambda_\mathrm{c} = 0$ recovers unconstrained least squares. From the fitted mean, CUE reads the realized resolution of message ι_ℓ, $\widehat{J}_\ell = (\widehat{\mu}_\ell - \widehat{\mu}_{\ell-1})^2$, a realized shift whose conditional expectation is γ_ℓ when the mean is exact, and the realized amplification of a trajectory, $\widehat{\mathcal{A}} = \sum_{\ell=1}^M (\widehat{V}_\ell - \widehat{V}_{\ell-1})_+$.

5 Experiment

We evaluate CUE on four benchmarks, three multi-agent orchestrations, and two backbone families: terminal failure prediction against seven baselines (Section 5.2), ablations of the communication history and the uncertainty constraint (Section 5.3), cost and scalability (Section 5.4), and case studies that test the dynamics predicted in Section 3 (Section 5.5).

Table 1: Llama-3.1-70B-Instruct AUROC↑ ($\times 100$), across 14 multi-agent configurations. SP=SelfPlanner, S3/S5=GPT-Swarm with 3/5 agents, k=number of debate agents. **Bold**=best, underline=second per column. Final row is task accuracy, shared by all UQ methods.

	HumanEval			MBPP			MATH500				MMLU				
Method	SP	S3	S5	SP	S3	S5	$k2$	$k3$	$k5$	$k7$	$k2$	$k3$	$k5$	$k7$	Avg.
PE	70.5 ±4.7	66.8 ±5.1	61.8 ±5.6	58.2 ±4.0	60.4 ±2.0	58.4 ±2.0	52.9 ±2.4	66.9 ±1.7	71.3 ±1.8	75.6 ±2.1	55.0 ±1.1	68.3 ±1.5	70.7 ±3.2	72.9 ±2.0	65.0
SE	53.2 ±3.8	53.1 ±2.6	51.8 ±4.8	56.4 ±3.1	52.5 ±2.2	50.0 ±2.0	79.2 ±1.9	83.6 ±1.2	83.2 ±1.8	84.0 ±1.7	66.0 ±1.1	71.6 ±2.7	63.2 ±2.5	73.7 ±4.0	65.8
SAR	62.7 ±3.8	54.4 ±6.9	50.4 ±1.7	51.7 ±2.5	60.5 ±1.8	60.0 ±2.2	58.2 ±3.6	72.2 ±1.4	75.5 ±1.8	78.0 ±1.3	59.9 ±2.1	66.5 ±2.8	67.1 ±2.7	66.2 ±1.8	63.1
SAUP	67.8 ±5.6	58.8 ±4.5	66.4 ±3.1	50.5 ±3.7	59.5 ±2.4	59.1 ±2.8	67.4 ±1.9	72.3 ±1.5	76.1 ±1.6	77.1 ±2.8	62.6 ±3.5	73.2 ±3.6	67.2 ±1.8	73.7 ±4.8	66.6
LARS	60.8 ±3.9	60.1 ±3.8	58.9 ±4.7	59.1 ±2.0	62.7 ±1.9	63.0 ±1.6	58.0 ±2.6	72.0 ±2.0	77.8 ±2.2	79.0 ±2.4	59.9 ±3.7	68.1 ±1.6	67.9 ±1.6	71.4 ±3.5	65.6
Sup	69.2 ±6.2	74.6 ±2.5	74.1 ±2.9	59.6 ±1.7	63.4 ±1.0	63.5 ±2.9	64.0 ±2.1	80.9 ±1.0	79.3 ±1.9	79.6 ±2.1	63.2 ±2.8	65.8 ±3.5	72.9 ±1.7	77.5 ±3.0	70.5
PropUQ	67.6 ±5.9	68.8 ±4.0	76.6 ±2.9	51.6 ±3.7	59.7 ±2.2	59.2 ±2.8	64.0 ±1.5	70.4 ±1.3	73.7 ±1.7	76.2 ±2.6	60.2 ±1.6	74.9 ±3.7	71.1 ±3.5	73.6 ±2.9	67.7
CUE	**71.8** ±3.7	**90.8** ±2.8	**87.7** ±10.7	**68.7** ±2.4	**75.0** ±3.8	**75.8** ±3.2	**88.9** ±2.5	**87.3** ±2.7	**91.4** ±0.5	**89.1** ±3.5	**78.0** ±3.3	**84.5** ±1.6	**78.4** ±3.2	**78.0** ±8.4	**81.8**
Task acc.	73.42	68.94	69.57	70.53	62.97	63.22	60.40	65.00	67.60	67.00	87.60	86.40	88.60	86.88	72.72

5.1 Experimental Settings

Benchmarks. We use two coding benchmarks, HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021), scored by unit-test pass@1, and two QA benchmarks, MATH500 (Hendrycks et al., 2021b) and MMLU (Hendrycks et al., 2021a), scored by exact match of the final answer.

Multi-agent systems. We use SelfPlanner (Jiang et al., 2024), GPT-Swarm (Zhuge et al., 2024), and multi-agent debate (Du et al., 2024) (Appendix C.1).

Backbones. Llama-3.1-70B-Instruct (Touvron et al., 2023; Team, 2024) runs every task; Qwen2.5-Coder-32B-Instruct (Yang et al., 2024) runs code and Qwen3-30B-A3B-Instruct-2507 (Team, 2025) runs QA. CUE reads its features from the same frozen backbone.

Baselines. We compare against seven baselines in four families: likelihood-based PE (Malinin & Gales, 2021); semantic SE (Kuhn et al., 2023) and SAR (Duan et al., 2024); supervised Sup (Liu et al., 2024) and LARS (Yaldiz et al., 2025), trained on the same split as CUE; and multi-agent SAUP (Zhao et al., 2025) and PropUQ (Liu et al., 2026).

Training and evaluation. Each dataset is split by query into 40%/10%/50% train/validation/test, over six random splits. The width and λ_c of CUE and the baseline hyperparameters are selected on validation AUROC (Appendices C and C.2). The metrics are the AUROC of the risk score for separating incorrect from correct final answers (Kuhn et al., 2023; Liu et al., 2024) and the prediction rejection ratio (PRR). Brier score, NLL, and ECE are reported in Appendix C.3.

5.2 Effectiveness Analysis

Tables 1 and 2 report the AUROC of predicting final-answer failure on the 14 configurations of each backbone. CUE is best in 27 of the 28, averaging 81.8 on Llama and 82.8 on Qwen against 70.5 and 71.4 for the strongest baseline, Sup (Wilcoxon signed-rank over configurations, $p = 1.2 \times 10^{-4}$ on each backbone). The one loss, Qwen MMLU with k=7 (75.2 vs. 75.7 for PropUQ), and five of the wins are within one standard deviation of the best baseline: HumanEval SP and MMLU k=7 on Llama, and HumanEval SP and MMLU k=2, 5 on Qwen.

The margin is largest where the final response says least on its own. On the HumanEval swarms, SE and SAR are near chance on Llama (50.4–54.4) and the best baseline reaches 74.6 on S3 (Sup) and 76.6 on S5 (PropUQ), against 90.8 and 87.7 for CUE. It is smallest on SelfPlanner, whose trajectory holds only a plan and a program (+1.3 on Llama HumanEval), and on MMLU debate with k=7. On MATH500 debate, where agreement among the debaters is informative and SE is the strongest baseline in 7 of 8 configurations, CUE still gains 3.7–9.7 AUROC on Llama and 5.5–10.7 on Qwen. Qwen2.5-Coder-32B solves about 97% of HumanEval in both swarms, leaving about three failures per test split, so its 99.7–99.9 there rests on very few negatives. CUE also has the best PRR in all 28 configurations (Tables 3 and 4).

5.3 Ablation Study

Table 5 varies two things on Llama debate: what the network reads at each message, and which agents' messages it is trained on. Full reads the query and all messages so far, *w/o intermediate msgs.* the query and the current message, and *query only* the query; *Train on* is whether all agents' messages

Table 2: AUROC↑ (×100), across 14 multi-agent configurations. SP=SelfPlanner, S3/S5=GPT-Swarm with 3/5 agents, k=number of debate agents. **Bold**=best, underline=second per column. Final row is task accuracy, shared by all UQ methods.

	Qwen2.5-Coder-32B-Instruct						Qwen3-30B-A3B-Instruct-2507								
	HumanEval			MBPP			MATH500				MMLU				
Method	SP	S3	S5	SP	S3	S5	$k2$	$k3$	$k5$	$k7$	$k2$	$k3$	$k5$	$k7$	Avg.
PE	57.9 ±4.8	63.2 ±5.2	60.2 ±5.3	61.8 ±1.6	60.1 ±2.6	63.2 ±2.4	51.8 ±1.7	51.8 ±3.6	53.9 ±2.5	50.9 ±4.6	61.8 ±3.0	66.9 ±2.8	71.9 ±3.8	71.9 ±2.8	60.5
SE	52.5 ±6.5	54.9 ±1.2	54.8 ±0.7	57.9 ±2.2	52.4 ±1.7	50.2 ±0.8	83.0 ±2.1	87.6 ±2.0	87.3 ±2.0	84.2 ±2.1	60.0 ±1.8	64.7 ±3.0	64.5 ±3.6	64.7 ±2.9	65.6
SAR	51.8 ±6.8	70.8 ±2.4	67.8 ±9.9	50.0 ±2.5	54.4 ±3.9	59.6 ±2.4	61.5 ±2.6	61.1 ±1.8	62.7 ±1.6	63.4 ±4.5	53.8 ±2.2	52.4 ±1.8	57.7 ±4.3	59.9 ±2.7	59.1
SAUP	54.3 ±7.5	52.8 ±16.5	56.2 ±16.1	60.1 ±1.0	62.6 ±2.2	63.8 ±1.6	65.9 ±1.9	66.5 ±3.9	66.3 ±2.1	68.1 ±4.6	66.7 ±2.1	67.9 ±2.5	68.4 ±1.8	63.8 ±2.9	63.1
LARS	52.6 ±3.9	60.8 ±13.7	59.7 ±13.4	56.6 ±3.7	56.9 ±3.3	58.1 ±3.2	55.7 ±3.8	56.6 ±1.9	59.8 ±2.1	66.6 ±3.3	66.2 ±1.7	68.4 ±2.8	71.5 ±3.3	73.3 ±2.9	61.6
Sup	58.1 ±5.9	54.3 ±6.2	77.7 ±15.7	61.6 ±3.1	62.5 ±1.4	63.4 ±2.6	83.3 ±1.9	87.3 ±1.5	85.7 ±1.4	83.9 ±2.5	65.3 ±2.2	69.8 ±3.9	76.9 ±3.3	70.4 ±4.4	71.4
PropUQ	54.8 ±7.8	54.3 ±15.0	62.5 ±13.0	57.1 ±1.9	62.4 ±2.2	63.5 ±1.7	67.1 ±1.4	67.4 ±4.3	68.2 ±2.0	69.1 ±4.4	69.6 ±3.0	73.2 ±2.1	77.1 ±3.3	**75.7** ±2.8	65.9
CUE	**63.1** ±5.1	**99.7** ±0.4	**99.9** ±0.2	**67.0** ±4.1	**75.2** ±4.4	**74.7** ±2.6	**94.0** ±3.3	**95.3** ±1.9	**92.8** ±2.4	**94.1** ±2.0	**72.7** ±7.2	**76.9** ±1.7	**78.0** ±4.5	75.2 ±3.7	**82.8**
Task acc.	79.75	96.95	96.95	81.61	71.03	70.78	71.40	73.00	74.60	75.60	86.00	87.40	88.60	87.40	81.51

Table 3: **Llama-3.1-70B-Instruct.** PRR↑ (×100), across 14 multi-agent configurations. SP=SelfPlanner, S3/S5=GPT-Swarm with 3/5 agents, k=number of debate agents. **Bold**=best, underline=second per column.

	HumanEval			MBPP			MATH500				MMLU				
Method	SP	S3	S5	SP	S3	S5	$k2$	$k3$	$k5$	$k7$	$k2$	$k3$	$k5$	$k7$	Avg.
PE	49.5 ±10.9	28.5 ±11.7	16.0 ±13.1	16.9 ±10.1	21.2 ±4.4	22.7 ±6.3	0.4 ±8.1	27.1 ±6.6	49.5 ±3.9	55.5 ±3.9	17.6 ±2.0	48.0 ±4.6	53.7 ±7.3	58.0 ±3.6	33.2
SE	4.3 ±13.4	−0.9 ±8.0	−5.9 ±22.5	12.0 ±12.7	2.2 ±9.5	−0.2 ±8.3	60.9 ±3.9	70.4 ±6.8	66.0 ±9.2	68.8 ±7.5	38.2 ±9.5	41.0 ±9.0	21.3 ±15.5	49.1 ±17.0	30.5
SAR	37.8 ±7.9	18.0 ±16.3	9.0 ±8.2	0.8 ±8.0	24.5 ±4.3	26.8 ±8.0	22.2 ±9.7	50.0 ±3.7	57.0 ±4.4	62.7 ±2.3	30.1 ±4.2	50.5 ±6.0	43.7 ±8.9	38.7 ±9.2	33.7
SAUP	36.7 ±13.1	21.6 ±11.1	37.3 ±6.7	−2.8 ±10.6	18.8 ±5.3	19.3 ±8.7	38.8 ±3.5	48.0 ±7.2	58.0 ±3.8	58.9 ±5.1	31.3 ±12.4	59.8 ±6.8	50.1 ±3.8	62.3 ±10.8	38.4
LARS	39.5 ±6.6	17.5 ±9.6	17.8 ±14.0	19.4 ±5.1	26.5 ±4.2	27.3 ±2.1	16.0 ±7.3	42.5 ±5.8	57.6 ±6.5	55.2 ±5.5	28.3 ±9.2	51.3 ±4.6	50.6 ±3.9	51.1 ±9.4	35.8
Sup	49.5 ±13.6	42.4 ±10.1	44.7 ±8.4	19.2 ±9.6	32.3 ±3.5	29.7 ±7.2	33.0 ±4.4	70.0 ±3.6	61.9 ±3.4	64.8 ±4.8	34.4 ±7.1	29.0 ±15.5	53.9 ±7.6	61.7 ±9.4	44.8
PropUQ	34.0 ±13.5	38.1 ±11.0	53.0 ±6.9	−2.8 ±10.2	18.4 ±4.7	18.6 ±8.8	29.2 ±4.2	38.9 ±5.9	53.3 ±3.5	54.8 ±5.6	26.5 ±8.1	63.1 ±7.8	57.9 ±7.6	60.4 ±6.2	38.8
CUE	**56.5** ±5.5	**85.0** ±5.3	**71.2** ±37.2	**40.0** ±6.8	**56.9** ±7.4	**56.0** ±6.0	**81.7** ±5.1	**79.9** ±4.2	**88.4** ±1.1	**84.7** ±7.0	**62.7** ±10.0	**75.4** ±5.4	**66.2** ±5.7	**62.7** ±18.6	**69.1**

or only the final agent's are used for training. Every row is scored at the final answer. *What is read.* The query alone (*query only*) gives 75.7 against 89.2 for Full. Reading the query and one message at a time (*w/o intermediate msgs.*) gives 79.2, so the intermediate messages account for the remaining 10.0 points. *Which agents train it.* Training on the final agent's message only changes little: 88.9 for Full and 78.2 without intermediate messages. The terminal score therefore comes from reading the full record; fitting every prefix is what gives CUE a score after each message. Removing the variance constraint ($\lambda_c = 0$) changes the average by 0.16, well below the per-split standard deviations (0.5–3.5).

Table 5: Ablation study on MATH500 debate, Llama-3.1-70B-Instruct. AUROC↑ (×100)

Variant	Train on	$k2$	$k3$	$k5$	$k7$	Avg
Full	all agents	**88.94** ±2.54	**87.32** ±2.68	**91.44** ±0.52	**89.05** ±3.46	**89.2**
Full, $\lambda_c = 0$	all agents	88.82 ±2.53	87.22 ±2.80	91.39 ±0.49	88.73 ±3.54	89.0
Full	final agent	88.84 ±1.33	87.17 ±2.76	91.20 ±1.40	88.51 ±3.10	88.9
w/o intermediate msgs.	all agents	76.91 ±2.51	79.64 ±2.93	81.19 ±1.79	78.97 ±1.92	79.2
w/o intermediate msgs.	final agent	76.53 ±1.90	79.47 ±2.89	79.50 ±2.09	77.44 ±1.65	78.2
query only	all agents	73.62 ±3.08	75.40 ±2.49	76.39 ±2.64	77.36 ±1.42	75.7

5.4 Computation Cost and Scalability

The cost of CUE is dominated by feature extraction: one frozen-backbone forward pass per prefix, about 2.2×10^3 s per Llama-3.1-70B debate configuration, which makes CUE the most expensive method end to end. Extraction is done once and cached; fitting and scoring then take 1.3–22.5 s, grow by at most 35% from 20% to 100% of the training data, and grow roughly linearly with the number of prefixes, from 3.1 s at k=2 to 10.6 s at k=7 (Figure 4). Given cached features, CUE lies on the AUROC–scoring-time Pareto frontier in all eight debate configurations (Appendix C.4, Figure 5).

5.5 Case Studies

We test whether the dynamics estimated by CUE show the behaviors Section 3 predicts, on two-round Llama-3.1-70B-Instruct debates over MATH500 with $k \in \{2, 3, 5, 7\}$ agents. We group executions by whether the round-1 answers are unanimous or split, and report each claim as the number of the 24 seed-level comparisons (4 agent counts × 6 splits) in which it holds, with an exact Wilcoxon signed-rank p-value (Figure 3). **Uncertainty contracts on average, not on every trajectory.** $\hat{V}$ falls by 0.141 from the query to the delivered answer (24/24, $p = 1.2 \times 10^{-7}$), as Theorem 3.1 predicts, yet it rises at some point on 51% of executions with a split round 1 and 41% with a unanimous one. Equation (9) permits such amplification, and it is more frequent after disagreement (18/24, $p = 0.003$). **Resolution depends on the history.** The two groups diverge as communication proceeds: from the

Table 4: **Qwen models.** PRR↑ ($\times 100$), across 14 multi-agent configurations. HumanEval and MBPP use Qwen2.5-Coder-32B-Instruct; MATH500 and MMLU use Qwen3-30B-A3B-Instruct-2507. SP=SelfPlanner, S3/S5=GPT-Swarm with 3/5 agents, k=number of debate agents. **Bold**=best, underline=second per column.

	Qwen2.5-Coder-32B-Instruct						Qwen3-30B-A3B-Instruct-2507								
	HumanEval			MBPP			MATH500				MMLU				
Method	SP	S3	S5	SP	S3	S5	$k2$	$k3$	$k5$	$k7$	$k2$	$k3$	$k5$	$k7$	Avg.
PE	25.8 ±7.9	28.5 ±10.7	10.1 ±11.4	29.0 ±10.7	33.6 ±5.3	37.8 ±3.0	−16.7 ±4.7	−25.6 ±10.1	−18.6 ±8.8	−21.5 ±13.5	27.3 ±8.9	41.3 ±10.1	52.3 ±10.2	48.1 ±8.2	18.0
SE	2.4 ±24.9	28.4 ±20.6	15.5 ±28.7	8.8 ±9.5	3.4 ±4.7	−2.8 ±6.8	67.2 ±8.8	78.2 ±7.7	74.5 ±9.0	70.4 ±9.6	26.3 ±7.3	31.2 ±20.2	23.7 ±23.9	19.1 ±17.4	31.9
SAR	−11.9 ±16.3	64.4 ±3.0	53.7 ±22.6	2.8 ±6.5	15.5 ±11.3	20.6 ±7.6	22.5 ±10.0	29.5 ±4.4	28.9 ±5.7	32.0 ±10.6	15.1 ±4.3	−0.8 ±12.9	27.0 ±10.6	30.9 ±5.7	23.6
SAUP	11.3 ±26.9	5.1 ±35.4	21.9 ±31.1	21.2 ±5.3	37.3 ±4.1	39.5 ±3.0	22.4 ±6.1	20.9 ±11.9	33.1 ±5.8	32.4 ±9.3	42.1 ±6.2	35.8 ±10.3	43.1 ±8.2	40.2 ±7.4	29.0
LARS	5.1 ±8.5	−14.7 ±58.2	−16.1 ±58.1	10.4 ±11.2	20.6 ±7.0	23.6 ±5.8	2.3 ±10.7	10.2 ±7.7	5.9 ±6.7	30.5 ±7.0	39.8 ±6.4	41.6 ±5.0	52.2 ±7.2	58.4 ±6.5	19.3
Sup	24.1 ±14.1	−12.2 ±34.0	68.7 ±27.3	25.9 ±9.6	38.3 ±4.2	39.2 ±7.7	75.4 ±4.1	76.9 ±4.8	76.5 ±3.8	69.4 ±6.3	47.1 ±4.5	47.3 ±9.5	63.6 ±7.0	44.2 ±12.4	48.9
PropUQ	11.1 ±20.5	13.8 ±27.0	42.1 ±20.8	12.8 ±9.0	36.7 ±4.3	38.8 ±3.2	26.9 ±6.4	25.2 ±12.5	38.3 ±5.4	35.5 ±9.3	46.0 ±7.2	53.3 ±8.1	62.2 ±8.5	57.9 ±7.4	35.8
CUE	**32.3** ±21.0	**99.7** ±0.4	**99.9** ±0.2	**44.6** ±12.1	**52.8** ±9.4	**43.9** ±8.8	**89.2** ±6.2	**91.5** ±4.2	**87.4** ±3.8	**90.3** ±3.4	**53.5** ±15.0	**63.4** ±3.0	**64.3** ±7.4	**59.2** ±7.8	**69.4**

query to the delivered answer, $\hat{p}$ rises by 0.118 after a unanimous round 1 and falls by 0.097 after a split one, while $\hat{V}$ falls by 0.225 and 0.067. A round-2 message resolves more after disagreement ($\hat{J}$ larger in 23/24, median ratio 3.7×), and round 2 resolves less per message than round 1 (24/24, $p = 1.2 \times 10^{-7}$), as when later messages repeat earlier answers. This is the history dependence of Proposition 3.2: what a message resolves depends on what the earlier communication has revealed. MMLU shows the same patterns: amplification is more frequent after disagreement (17/24, $p = 0.009$), and round-2 resolution is larger after a split round 1 (22/24, median 2.9×). Appendix D gives round-level attribution, early detection, and representation diagnostics.

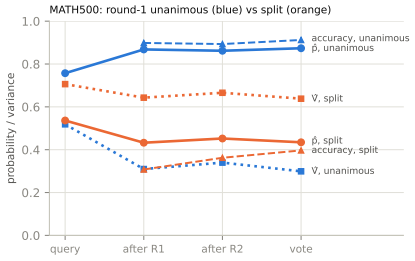

Figure 3: Evolution of estimated correctness $\hat{p}$ and uncertainty $\hat{V}$ along MATH500 debates, grouped by unanimous versus split round-1 answers.

6 Related Work

Multi-agent systems. LLM agents progressed from tool use (Nakano et al., 2021; Qiao et al., 2024) and interleaved reasoning and acting (Yao et al., 2023; Shinn et al., 2023; Renze & Guven, 2024) to communicating systems such as CAMEL (Li et al., 2023) and AutoGen (Wu et al., 2023). Communication can spread errors and overconfidence (Pitre et al., 2025; Wynn et al., 2025). Failure analyses explain failures after the execution, through taxonomies of annotated traces (Cemri et al., 2025) or by attributing them to an agent and step (Zhang et al., 2025). Multi-agent uncertainty estimators aggregate per-agent scores (Han et al., 2024; Zhao et al., 2025) or propagate them over the communication graph (Liu et al., 2026), and process reward models (Xi et al., 2026) and runtime monitors (Custers & Aslansefat, 2027) score partial traces.

Uncertainty estimation in LLMs. Existing methods rely on self-assessment (Kadavath et al., 2022; Lin et al., 2022; Tian et al., 2023), output likelihood (Manakul et al., 2023; Malinin & Gales, 2021), semantic aggregation (Kuhn et al., 2023; Duan et al., 2024), or supervised scoring functions trained on correctness labels (Liu et al., 2024; Yaldiz et al., 2025). Closest to CUE are hidden-state probes, which predict truthfulness with supervision (Azaria & Mitchell, 2023; Liu et al., 2024) or consistency constraints (Burns et al., 2023), or predict sampling-based uncertainty from one generation (Kossen et al., 2024); CUE applies such a probe to every prefix of a multi-agent execution.

7 Conclusion

We studied how uncertainty about a multi-agent system's outcome evolves through communication. Standard martingale identities on the message filtration show that communication resolves uncertainty in expectation, even though a single trajectory can amplify it. They split the resolved uncertainty across messages without assuming how agents depend on each other, and they hold for the features an estimator observes. Building on them, CUE estimates correctness and its uncertainty after every message and attributes the resolution to rounds, as information revealed to an observer. Across four benchmarks, three orchestrations, and two LLM families, it predicts failure better than seven baselines in most configurations. Uncertainty in multi-agent systems is thus best modeled as it evolves through communication, not from the final output alone.

Reproducibility statement

All theoretical results in Section 3 hold for any square-integrable outcome and nested information sets, with no further assumptions; their complete proofs are in Appendix A, and the proof that the identities also hold for the feature history the estimator observes is in Appendix B.1. CUE is specified in Section 4: the prefix representation, the training objective with its penalty, and the quantities derived from the fitted mean. Appendix C.2 lists the backbone layers, token budgets, network architecture, optimizer, training schedule, and the full hyperparameter grid with its selection rule. Section 5.1 and Appendix C.1 describe the benchmarks, the correctness criteria, the three orchestrations and their 14 configurations per backbone, the 40%/10%/50% split by query, and the six random splits over which every result is reported, and Section 5.1 also states the rule by which baseline scores are re-oriented. All backbones are publicly available open-weight models used with fixed parameters. Section 5.1 and Appendix C list the baselines and how they are tuned. Per-configuration AUROC and PRR are in Section 5.2, and Brier score, NLL, and ECE in Appendix C.3; computational costs are in Appendix C.4, and the setup of every case study is in Appendix D.

Broader Impact

This work aims to make LLM-based multi-agent systems more reliable by modeling how uncertainty propagates through inter-agent communication. These systems are increasingly deployed in real-world and high-stakes settings, where uncertainty that accumulates unnoticed can make the whole system overconfident and brittle. Quantifying system-level uncertainty, instead of relying only on the final output, makes multi-agent decisions easier to inspect and to act on with appropriate caution.

The method is diagnostic and adds no new task-solving or persuasive capabilities. Any uncertainty framework could be manipulated to hide risk, but this work makes uncertainty propagation explicit and measurable, which supports safer deployment, monitoring, and governance of collaborative AI systems.

AI use statement

Large language models were used only for language polishing and wording refinement. Specifically, LLM assistance was limited to improving grammar, clarity, conciseness, and presentation of author-written text. All scientific ideas, problem formulation, theoretical results, proofs, algorithms, experimental design, implementation, data processing, numerical results, analysis, and conclusions were produced and verified by the authors. LLMs were not used to generate experimental results, fabricate data, perform reviewer simulation for reported claims, or make autonomous scientific decisions. The authors take full responsibility for the final content of the paper.

References

Jacob Austin, Augustus Odena, Maxwell I. Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie J. Cai, Michael Terry, Quoc V. Le, and Charles Sutton. Program synthesis with large language models. *CoRR*, abs/2108.07732, 2021. URL https://arxiv.org/abs/2108.07732.

Amos Azaria and Tom M. Mitchell. The internal state of an LLM knows when it's lying. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, volume EMNLP 2023 of *Findings of ACL*, pp. 967–976. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-EMNLP.68. URL https://doi.org/10.18653/v1/2023.findings-emnlp.68.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=ETKGuby0hcs.

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. Why do multi-agent LLM systems fail? In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/b1041e52d3be19f0a9bc491657488e4a-Abstract-Datasets_and_Benchmarks_Track.html.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Kaijie Zhu, Hao Chen, Linyi Yang, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. A survey on evaluation of large language models. *CoRR*, abs/2307.03109, 2023. doi: 10.48550/ARXIV.2307.03109. URL https://doi.org/10.48550/arXiv.2307.03109.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *CoRR*, abs/2107.03374, 2021. URL https://arxiv.org/abs/2107.03374.

Bart Custers and Koorosh Aslansefat. Runtime uncertainty monitoring for llm-based multi-agent systems using bayesian networks. In Elena Troubitsyna, David de Andrés, Erwin Schoitsch, Joaquin Gracia-Moran, and Friedemann Bitsch (eds.), *Computer Safety, Reliability, and Security. SAFECOMP 2026 Workshops - CSRIS, DAIES, DECSoS, ORCAS, SafetyNXT, SASSUR, SENSEI, and WAISE, Valencia, Spain, September 22, 2026, Proceedings*, volume 16869 of *Lecture Notes in Computer Science*, pp. 736–748. Springer, 2027. doi: 10.1007/978-3-032-35506-5_53. URL https://doi.org/10.1007/978-3-032-35506-5_53.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11733–11763. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/du24e.html.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 5050–5063. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.276. URL https://doi.org/10.18653/v1/2024.acl-long.276.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria-Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 1050–1059. JMLR.org, 2016. URL http://proceedings.mlr.press/v48/gal16.html.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference*

on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, volume 70 of *Proceedings of Machine Learning Research*, pp. 1183–1192. PMLR, 2017. URL http://proceedings.mlr.press/v70/gal17a.html.

Jiuzhou Han, Wray L. Buntine, and Ehsan Shareghi. Towards uncertainty-aware language agent. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, volume ACL 2024 of *Findings of ACL*, pp. 6662–6685. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-ACL.398. URL https://doi.org/10.18653/v1/2024.findings-acl.398.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021a. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In Joaquin Vanschoren and Sai-Kit Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021b. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

Xue Jiang, Yihong Dong, Lecheng Wang, Zheng Fang, Qiwei Shang, Ge Li, Zhi Jin, and Wenpin Jiao. Self-planning code generation with large language models. *ACM Trans. Softw. Eng. Methodol.*, 33(7):182:1–182:30, 2024. doi: 10.1145/3672456. URL https://doi.org/10.1145/3672456.

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, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know. *CoRR*, abs/2207.05221, 2022. doi: 10.48550/ARXIV.2207.05221. URL https://doi.org/10.48550/arXiv.2207.05221.

Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth A. Malik, and Yarin Gal. Semantic entropy probes: Robust and cheap hallucination detection in llms. *CoRR*, abs/2406.15927, 2024. doi: 10.48550/ARXIV.2406.15927. URL https://doi.org/10.48550/arXiv.2406.15927.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 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*. OpenReview.net, 2023. URL https://openreview.net/forum?id=VD-AYtP0dve.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. CAMEL: communicative agents for "mind" exploration of large language model society. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/a3621ee907def47c1b952ade25c67698-Abstract-Conference.html.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *Trans. Mach. Learn. Res.*, 2022, 2022. URL https://openreview.net/forum?id=8s8K2UZGTZ.

Linyu Liu, Yu Pan, Xiaocheng Li, and Guanting Chen. Uncertainty estimation and quantification for llms: A simple supervised approach. *CoRR*, abs/2404.15993, 2024. doi: 10.48550/ARXIV.2404.15993. URL https://doi.org/10.48550/arXiv.2404.15993.

Yaokun Liu, Yifan Liu, Daniel Yue Zhang, Ruichen Yao, Zelin Li, and Dong Wang. Propuqmas: Propagation-aware uncertainty quantification for LLM multi-agent systems. *CoRR*, abs/2608.22130, 2026. doi: 10.48550/ARXIV.2608.22130. URL https://doi.org/10.48550/arXiv.2608.22130.

Andrey Malinin and Mark J. F. Gales. Uncertainty estimation in autoregressive structured prediction. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=jN5y-zb5Q7m.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 9004–9017. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.557. URL https://doi.org/10.18653/v1/2023.emnlp-main.557.

Shervin Minaee, Tomás Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey. *CoRR*, abs/2402.06196, 2024. doi: 10.48550/ARXIV.2402.06196. URL https://doi.org/10.48550/arXiv.2402.06196.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. Webgpt: Browser-assisted question-answering with human feedback. *CoRR*, abs/2112.09332, 2021. URL https://arxiv.org/abs/2112.09332.

Priya Pitre, Naren Ramakrishnan, and Xuan Wang. CONSENSAGENT: towards efficient and effective consensus in multi-agent LLM interactions through sycophancy mitigation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, volume ACL 2025 of *Findings of ACL*, pp. 22112–22133. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.FINDINGS-ACL.1141. URL https://doi.org/10.18653/v1/2025.findings-acl.1141.

Shuofei Qiao, Honghao Gui, Chengfei Lv, Qianghuai Jia, Huajun Chen, and Ningyu Zhang. Making language models better tool learners with execution feedback. In Kevin Duh, Helena Gómez-Adorno, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pp. 3550–3568. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.NAACL-LONG.195. URL https://doi.org/10.18653/v1/2024.naacl-long.195.

Mohaimenul Azam Khan Raiaan, Md. Saddam Hossain Mukta, Kaniz Fatema, Nur Mohammad Fahad, Sadman Sakib, Most Marufatul Jannat Mim, Jubaer Ahmad, Mohammed Eunus Ali, and Sami Azam. A review on large language models: Architectures, applications, taxonomies, open issues and challenges. *IEEE Access*, 12:26839–26874, 2024. doi: 10.1109/ACCESS.2024.3365742. URL https://doi.org/10.1109/ACCESS.2024.3365742.

Matthew Renze and Erhan Guven. Self-reflection in LLM agents: Effects on problem-solving performance. *CoRR*, abs/2405.06682, 2024. doi: 10.48550/ARXIV.2405.06682. URL https://doi.org/10.48550/arXiv.2405.06682.

Noah Shinn, Beck Labash, and Ashwin Gopinath. Reflexion: an autonomous agent with dynamic memory and self-reflection. *CoRR*, abs/2303.11366, 2023. doi: 10.48550/ARXIV.2303.11366. URL https://doi.org/10.48550/arXiv.2303.11366.

Llama Team. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL https://doi.org/10.48550/arXiv.2407.21783.

Qwen Team. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025. doi: 10.48550/ARXIV.2505.09388. URL https://doi.org/10.48550/arXiv.2505.09388.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 5433–5442. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.330. URL https://doi.org/10.18653/v1/2023.emnlp-main.330.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023. doi: 10.48550/ARXIV.2307.09288. URL https://doi.org/10.48550/arXiv.2307.09288.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Jirong Wen. A survey on large language model based autonomous agents. *Frontiers Comput. Sci.*, 18(6):186345, 2024. doi: 10.1007/S11704-024-40231-1. URL https://doi.org/10.1007/s11704-024-40231-1.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. Autogen: Enabling next-gen LLM applications via multi-agent conversation framework. *CoRR*, abs/2308.08155, 2023. doi: 10.48550/ARXIV.2308.08155. URL https://doi.org/10.48550/arXiv.2308.08155.

Andrea Wynn, Harsh Satija, and Gillian Hadfield. Talk isn't always cheap: Understanding failure modes in multi-agent debate. *CoRR*, abs/2509.05396, 2025. doi: 10.48550/ARXIV.2509.05396. URL https://doi.org/10.48550/arXiv.2509.05396.

Zhiheng Xi, Chenyang Liao, Guanyu Li, Zhihao Zhang, Wenxiang Chen, Binghai Wang, Senjie Jin, Yuhao Zhou, Jian Guan, Wei Wu, Tao Ji, Tao Gui, Qi Zhang, and Xuanjing Huang. Agentprm: Process reward models for LLM agents via step-wise promise and progress. In Hakim Hacid, Yoelle Maarek, Francesco Bonchi, Ido Guy, and Emine Yilmaz (eds.), *Proceedings of the ACM Web Conference 2026, WWW 2026, Dubai, United Arab Emirates, originally scheduled for April 13-17, 2026, rescheduled for June 29 - July 3, 2026*, pp. 4184–4195. ACM, 2026. doi: 10.1145/3774904.3792551. URL https://doi.org/10.1145/3774904.3792551.

Duygu Nur Yaldiz, Yavuz Faruk Bakman, Baturalp Buyukates, Chenyang Tao, Anil Ramakrishna, Dimitrios Dimitriadis, Jieyu Zhao, and Salman Avestimehr. Do not design, learn: A trainable scoring function for uncertainty estimation in generative llms. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, volume NAACL 2025 of *Findings of ACL*, pp. 691–713. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.FINDINGS-NAACL.41. URL https://doi.org/10.18653/v1/2025.findings-naacl.41.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin

Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report. *CoRR*, abs/2407.10671, 2024. doi: 10.48550/ARXIV.2407.10671. URL https://doi.org/10.48550/arXiv.2407.10671.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. Do large language models know what they don't know? In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, volume ACL 2023 of *Findings of ACL*, pp. 8653–8665. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-ACL.551. URL https://doi.org/10.18653/v1/2023.findings-acl.551.

Shaokun Zhang, Ming Yin, Jieyu Zhang, Jiale Liu, Zhiguang Han, Jingyang Zhang, Beibin Li, Chi Wang, Huazheng Wang, Yiran Chen, and Qingyun Wu. Which agent causes task failures and when? on automated failure attribution of LLM multi-agent systems. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/zhang25cq.html.

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. Uncertainty propagation on LLM agent. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 6064–6073. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.ACL-LONG.302. URL https://doi.org/10.18653/v1/2025.acl-long.302.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. Gptswarm: Language agents as optimizable graphs. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62743–62767. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/zhuge24a.html.

A Proofs and Additional Details for Theoretical Analysis

This appendix gives the proofs and further technical details for the distribution-free analysis of uncertainty flow in Section 3. All random variables live on a common probability space $(\Omega, \mathcal{F}, \mathbb{P})$, and $Z^\star \in L^2$, i.e.,

$$\mathbb{E}[(Z^\star)^2] < \infty.$$

This finite second moment is enough for every conditional expectation and conditional variance below to be well defined. We assume no Gaussianity, no parametric posterior family, and no independence between agents or messages.

A.1 Preliminaries

Recall that the observed multi-agent execution induces an increasing filtration

$$\mathcal{H}_0 \subseteq \mathcal{H}_1 \subseteq \cdots \subseteq \mathcal{H}_T,$$

where $\mathcal{H}_t$ contains the task context and all communication observed up to round t. Define

$$\mu_t = \mathbb{E}[Z^\star \mid \mathcal{H}_t], \qquad V_t = \mathrm{Var}(Z^\star \mid \mathcal{H}_t). \tag{15}$$

We first record two standard facts used repeatedly below.

Conditional variance as posterior Bayes risk. For any $\mathcal{H}_t$-measurable decision a,

$$\begin{aligned}\mathbb{E}[(Z^\star - a)^2 \mid \mathcal{H}_t] &= \mathbb{E}[(Z^\star - \mu_t + \mu_t - a)^2 \mid \mathcal{H}_t] \\ &= \mathbb{E}[(Z^\star - \mu_t)^2 \mid \mathcal{H}_t] + (\mu_t - a)^2 \\ &\quad + 2(\mu_t - a)\mathbb{E}[Z^\star - \mu_t \mid \mathcal{H}_t]. \end{aligned} \tag{16}$$

By the definition of conditional expectation,

$$\mathbb{E}[Z^\star - \mu_t \mid \mathcal{H}_t] = 0.$$

Hence

$$\mathbb{E}[(Z^\star - a)^2 \mid \mathcal{H}_t] = V_t + (\mu_t - a)^2. \tag{17}$$

The right-hand side is minimized by $a = \mu_t$, yielding

$$\inf_a \mathbb{E}[(Z^\star - a)^2 \mid \mathcal{H}_t] = V_t. \tag{18}$$

Martingale property. Since $\mathcal{H}_{t-1} \subseteq \mathcal{H}_t$, the tower property gives

$$\begin{aligned}\mathbb{E}[\mu_t \mid \mathcal{H}_{t-1}] &= \mathbb{E}\left[\mathbb{E}[Z^\star \mid \mathcal{H}_t] | \mathcal{H}_{t-1}\right] \\ &= \mathbb{E}[Z^\star \mid \mathcal{H}_{t-1}] = \mu_{t-1}. \end{aligned} \tag{19}$$

Therefore $\{\mu_t\}_{t=0}^T$ is a square-integrable Doob martingale with respect to $\{\mathcal{H}_t\}$.

A.2 Proof of Theorem 3.1

Proof. By definition,

$$V_{t-1} = \mathbb{E}\left[(Z^\star - \mu_{t-1})^2 \mid \mathcal{H}_{t-1}\right]. \tag{20}$$

Insert and subtract μ_t:

$$Z^\star - \mu_{t-1} = (Z^\star - \mu_t) + (\mu_t - \mu_{t-1}). \tag{21}$$

Expanding the square gives

$$\begin{aligned}V_{t-1} = &\mathbb{E}\left[(Z^\star - \mu_t)^2 \mid \mathcal{H}_{t-1}\right] \\ &+ \mathbb{E}\left[(\mu_t - \mu_{t-1})^2 \mid \mathcal{H}_{t-1}\right] \\ &+ 2\mathbb{E}\left[(Z^\star - \mu_t)(\mu_t - \mu_{t-1}) \mid \mathcal{H}_{t-1}\right]. \end{aligned} \tag{22}$$

For the first term, the tower property yields

$$\begin{aligned}\mathbb{E}\left[(Z^\star - \mu_t)^2 \mid \mathcal{H}_{t-1}\right] &= \mathbb{E}\left[\mathbb{E}[(Z^\star - \mu_t)^2 \mid \mathcal{H}_t] \middle| \mathcal{H}_{t-1}\right] \\ &= \mathbb{E}[V_t \mid \mathcal{H}_{t-1}]. \end{aligned} \tag{23}$$

For the cross term, $\mu_t - \mu_{t-1}$ is $\mathcal{H}_t$-measurable. Thus

$$\begin{aligned}&\mathbb{E}\left[(Z^\star - \mu_t)(\mu_t - \mu_{t-1}) \mid \mathcal{H}_{t-1}\right] \\ &= \mathbb{E}\left[(\mu_t - \mu_{t-1})\mathbb{E}[Z^\star - \mu_t \mid \mathcal{H}_t] | \mathcal{H}_{t-1}\right] = 0. \end{aligned} \tag{24}$$

Substituting Equations (23) and (24) into Equation (22) gives

$$V_{t-1} = \mathbb{E}[V_t \mid \mathcal{H}_{t-1}] + \mathbb{E}\left[(\mu_t - \mu_{t-1})^2 \mid \mathcal{H}_{t-1}\right], \tag{25}$$

which proves the result. □

Equivalently, the theorem is the conditional law of total variance applied to the nested information sets $\mathcal{H}_{t-1} \subseteq \mathcal{H}_t$:

$$\begin{aligned}\mathrm{Var}(Z^\star \mid \mathcal{H}_{t-1}) = &\mathbb{E}\left[\mathrm{Var}(Z^\star \mid \mathcal{H}_t) | \mathcal{H}_{t-1}\right] \\ &+ \mathrm{Var}\left(\mathbb{E}[Z^\star \mid \mathcal{H}_t] | \mathcal{H}_{t-1}\right). \end{aligned} \tag{26}$$

Using Equation (19), the final term is precisely

$$\mathbb{E}[(\mu_t - \mu_{t-1})^2 \mid \mathcal{H}_{t-1}].$$

A.3 Uninformative and Redundant Communication

Suppose the messages newly observed at round t satisfy

$$Z^\star \perp \left(M^{(t)}_{e_{t,1}}, \ldots, M^{(t)}_{e_{t,L_t}}\right) \mid \mathcal{H}_{t-1}. \tag{27}$$

Since

$$\mathcal{H}_t = \sigma(\mathcal{H}_{t-1}, \left(M^{(t)}_{e_{t,1}}, \ldots, M^{(t)}_{e_{t,L_t}}\right)),$$

conditional independence implies that the conditional distribution of $Z^\star$ is unchanged after observing $\left(M^{(t)}_{e_{t,1}}, \ldots, M^{(t)}_{e_{t,L_t}}\right)$. In particular,

$$\mathbb{E}[Z^\star \mid \mathcal{H}_t] = \mathbb{E}[Z^\star \mid \mathcal{H}_{t-1}] \qquad \text{a.s.}, \tag{28}$$

and therefore

$$\mu_t = \mu_{t-1} \qquad \text{a.s.} \tag{29}$$

It follows immediately that

$$\Gamma_t = \mathbb{E}\left[(\mu_t - \mu_{t-1})^2 \mid \mathcal{H}_{t-1}\right] = 0. \tag{30}$$

In fact, conditional independence yields the stronger identity

$$V_t = V_{t-1} \qquad \text{a.s.}, \tag{31}$$

because the full conditional distribution of $Z^\star$ does not change. Redundant communication therefore cannot resolve any further task-relevant uncertainty once the earlier trajectory contains its information.

A.4 Proof of Proposition 3.2

We now write out the message-level filtration that the main text leaves implicit. Suppose round t contains L_t directed messages, indexed under a fixed ordering as

$$M^{(t)}_{e_{t,1}}, \ldots, M^{(t)}_{e_{t,L_t}}.$$

Define

$$\mathcal{H}_{t,r} = \sigma\left(\mathcal{H}_{t-1}, M^{(t)}_{e_{t,1}}, \ldots, M^{(t)}_{e_{t,r}}\right), \qquad r = 0, \ldots, L_t, \tag{32}$$

where

$$\mathcal{H}_{t,0} = \mathcal{H}_{t-1}, \qquad \mathcal{H}_{t,L_t} = \mathcal{H}_t.$$

Define

$$\mu_{t,r} = \mathbb{E}[Z^\star \mid \mathcal{H}_{t,r}]. \tag{33}$$

The sequence $\{\mu_{t,r}\}_{r=0}^{L_t}$ is itself a martingale with respect to $\{\mathcal{H}_{t,r}\}$. Let

$$D_{t,r} = \mu_{t,r} - \mu_{t,r-1} \tag{34}$$

denote the corresponding martingale differences. Then

$$\mathbb{E}[D_{t,r} \mid \mathcal{H}_{t,r-1}] = 0. \tag{35}$$

Moreover,

$$\mu_t - \mu_{t-1} = \sum_{r=1}^{L_t} D_{t,r}. \tag{36}$$

Proof. Starting from the definition of Γ_t,

$$\begin{aligned} \Gamma_t &= \mathbb{E}\left[(\mu_t - \mu_{t-1})^2 \mid \mathcal{H}_{t-1}\right] \\ &= \mathbb{E}\left[\left(\sum_{r=1}^{L_t} D_{t,r}\right)^2 \middle| \mathcal{H}_{t-1}\right]. \end{aligned} \tag{37}$$

Expanding,

$$\Gamma_t = \sum_{r=1}^{L_t} \mathbb{E}[D_{t,r}^2 \mid \mathcal{H}_{t-1}] + 2 \sum_{1 \le r < s \le L_t} \mathbb{E}[D_{t,r} D_{t,s} \mid \mathcal{H}_{t-1}]. \tag{38}$$

We show that every cross term vanishes. For $r < s$, $D_{t,r}$ is $\mathcal{H}_{t,s-1}$-measurable. Therefore

$$\begin{aligned} &\mathbb{E}[D_{t,r} D_{t,s} \mid \mathcal{H}_{t-1}] \\ &= \mathbb{E}\left[\mathbb{E}[D_{t,r} D_{t,s} \mid \mathcal{H}_{t,s-1}] | \mathcal{H}_{t-1}\right] \\ &= \mathbb{E}\left[D_{t,r} \mathbb{E}[D_{t,s} \mid \mathcal{H}_{t,s-1}] | \mathcal{H}_{t-1}\right] = 0, \end{aligned} \tag{39}$$

where the last equality follows from Equation (35).

Hence Equation (38) reduces to

$$\Gamma_t = \sum_{r=1}^{L_t} \mathbb{E}\left[(\mu_{t,r} - \mu_{t,r-1})^2 \mid \mathcal{H}_{t-1}\right], \tag{40}$$

which proves Proposition 3.2. □

Order dependence of individual contributions. The total Γ_t depends only on $\mathcal{H}_{t-1}$ and the final information set $\mathcal{H}_t$, so it does not depend on the order in which the messages of a round are revealed. The individual increments

$$\mathbb{E}[D_{t,r}^2 \mid \mathcal{H}_{t-1}]$$

can depend on the chosen ordering.

This happens naturally when messages carry overlapping information. If two messages encode the same task-relevant evidence, whichever is revealed first may receive most of the incremental resolution, and the second becomes largely redundant once we condition on the first. Their sum still equals the order-invariant round-level resolution Γ_t.

Proposition 3.2 should therefore be read as a sequential attribution of communication value, not as a unique causal allocation to individual messages.

A.5 Proof of Corollary 3.3

From Theorem 3.1,

$$V_{t-1} - \mathbb{E}[V_t \mid \mathcal{H}_{t-1}] = \Gamma_t. \tag{41}$$

Condition both sides on $\mathcal{H}_0$. Because $\mathcal{H}_0 \subseteq \mathcal{H}_{t-1}$, the tower property gives

$$\mathbb{E}[V_{t-1} \mid \mathcal{H}_0] - \mathbb{E}[V_t \mid \mathcal{H}_0] = \mathbb{E}[\Gamma_t \mid \mathcal{H}_0]. \tag{42}$$

Summing over $t = 1, \ldots, T$ gives

$$\begin{aligned} &\sum_{t=1}^{T} \left(\mathbb{E}[V_{t-1} \mid \mathcal{H}_0] - \mathbb{E}[V_t \mid \mathcal{H}_0]\right) \\ &= \sum_{t=1}^{T} \mathbb{E}[\Gamma_t \mid \mathcal{H}_0]. \end{aligned} \tag{43}$$

The left-hand side telescopes:

$$V_0 - \mathbb{E}[V_T \mid \mathcal{H}_0] = \sum_{t=1}^{T} \mathbb{E}[\Gamma_t \mid \mathcal{H}_0], \tag{44}$$

where V_0 is $\mathcal{H}_0$-measurable. Rearranging proves

$$V_0 = \mathbb{E}[V_T \mid \mathcal{H}_0] + \sum_{t=1}^{T} \mathbb{E}[\Gamma_t \mid \mathcal{H}_0]. \tag{45}$$

Combining this identity with Proposition 3.2 yields the finer message-level decomposition

$$\boxed{V_0 - \mathbb{E}[V_T \mid \mathcal{H}_0] = \sum_{t=1}^{T} \sum_{r=1}^{L_t} \mathbb{E}\left[(\mu_{t,r} - \mu_{t,r-1})^2 \mid \mathcal{H}_0\right].} \tag{46}$$

The expected reduction in uncertainty thus decomposes hierarchically, from individual directed messages to rounds to the complete trajectory.

A.6 Deferred from Section 3

Prefix law recursion. The communication filtration governs how the conditional *law* of $Z^\star$ evolves along the trajectory, beyond its first two moments. Write $q_{t,r}(\cdot \mid h)$ for the conditional law of $Z^\star$ given $\mathcal{H}_{t,r} = h$.

Corollary A.1 (Prefix Law Recursion)**.** *For every step* (t, r)*,*

$$q_{t,r-1}(\cdot \mid h) = \mathbb{E}\left[q_{t,r}(\cdot \mid \mathcal{H}_{t,r}) \mid \mathcal{H}_{t,r-1} = h\right]. \tag{47}$$

This is the tower property applied to the conditional law instead of its mean, and every quantity in it is a moment of $q_{t,r}$; taking the first two moments of Equation (47) gives back Theorem 3.1. The estimator of Section 4 fits only the conditional mean; for binary correctness the variance and the realized resolution are functions of it ($\widehat{V}_\ell = 1 - \widehat{\mu}_\ell^2$), and the recursion is not enforced beyond the penalty $\mathcal{R}_{\mathrm{c}}$. For a general real-valued margin, two laws can have the same mean and variance but put different mass below zero, so an event probability needs more distributional information or a direct regression on the event. Binary correctness is the special case in which the mean alone determines the event probability.

Amplification at message granularity. Realized amplification can be read by labels in the same way as resolution (Equation (49)) once the realized changes $(V_{t,r} - V_{t,r-1})_+$ are recorded per message. This finer count is not equal to the round-level count $\mathcal{A}_{1:T}$ defined after Equation (9), because increases within a round can cancel at round granularity.

A.7 Structural Attribution

The communication graph records which agent sent each message and over which channel. Grouping the message-level resolution by these labels assigns the same resolved total to different parts of the system.

Write $\mathcal{I} = \{(t, r) : 1 \le t \le T,\ 1 \le r \le L_t\}$ for the directed communications of the trajectory, with $\Gamma_{t,r}$ the per-message resolution of Equation (6). For any group $P \subseteq \mathcal{I}$ let $\Gamma[P] = \sum_{(t,r)\in P} \mathbb{E}[\Gamma_{t,r} \mid \mathcal{H}_0]$ denote the resolution that group carries.

Proposition A.2 (Structural Attribution)**.** *For every partition* $\mathcal{P}$ *of* $\mathcal{I}$*,*

$$\boxed{\sum_{P\in\mathcal{P}} \Gamma[P] = V_0 - \mathbb{E}[V_T \mid \mathcal{H}_0].} \tag{48}$$

The identity follows from summing Equation (6) over $\mathcal{I}$ and applying Proposition 3.2 and Corollary 3.3. Because it holds for *every* partition, the labels that the trajectory already carries give exhaustive, additive shares. Three follow directly:

$$\begin{aligned} P_t &= \{(t', r) \in \mathcal{I} : t' = t\}, && \text{round } t, \\ P_{A_k} &= \{(t, r) \in \mathcal{I} : j_{t,r} = k\}, && \text{sender } A_k, \\ P_{j\to i} &= \{(t, r) \in \mathcal{I} : (j_{t,r}, i_{t,r}) = (j, i)\}, && \text{channel } (j, i). \end{aligned} \tag{49}$$

The round partition recovers Corollary 3.3, since $\Gamma[P_t] = \mathbb{E}[\Gamma_t \mid \mathcal{H}_0]$. The agent and channel shares measure how much uncertainty the observer resolves through each sender or link.

The agent shares do not require per-agent belief states. A per-agent theory would give each A_k its own filtration and posterior; Equation (49) instead reads the share of A_k off the single filtration Equation (3), so it describes what an observer of the recorded history Equation (3) learned from the messages of A_k, not the posterior of A_k. When an external aggregation rule, rather than an agent, produces the delivered response, its entry should be reported as a separate group instead of being folded into some P_{A_k}, so that no agent receives credit for the aggregation.

Attribution is sequential and depends on the order: a later message is credited only for what it reveals beyond the messages before it. If one message reveals $Z^\star$ completely and the next repeats it, the first resolves all the remaining variance and the second resolves none. For concurrent messages we fix one serialization order and report it. The shares measure information revealed to the observer, not causal improvements in system accuracy.

A.8 Realized Uncertainty Amplification

Recall the realized uncertainty change

$$\Delta_t^{\text{real}} = V_t - V_{t-1}. \tag{50}$$

Since V_{t-1} is $\mathcal{H}_{t-1}$-measurable,

$$\begin{aligned}\mathbb{E}[\Delta_t^{\text{real}} \mid \mathcal{H}_{t-1}] &= \mathbb{E}[V_t \mid \mathcal{H}_{t-1}] - V_{t-1} \\ &= -\Gamma_t \le 0,\end{aligned} \tag{51}$$

where the second equality follows from Theorem 3.1.

Equation (51) holds in expectation and does not require monotonicity pointwise. In general,

$$V_t(\omega) > V_{t-1}(\omega) \tag{52}$$

may hold for some realized trajectories $\omega \in \Omega$. Such trajectories correspond to the realized uncertainty-amplification events defined in the main text.

The cumulative positive variation

$$\mathcal{A}_{1:T} = \sum_{t=1}^{T} (V_t - V_{t-1})_+ \tag{53}$$

records the total magnitude of these realized increases.

It is useful to contrast $\mathcal{A}_{1:T}$ with net variance change:

$$V_T - V_0 = \sum_{t=1}^{T} (V_t - V_{t-1}). \tag{54}$$

A trajectory can therefore show substantial amplification along the way even when it ends with less uncertainty than it started with. Terminal uncertainty and trajectory-level amplification measure different aspects of the communication dynamics.

Why realized variance can increase. Conditioning reduces variance in expectation, but that does not make every realized conditional variance smaller. The identity

$$V_{t-1} = \mathbb{E}[V_t \mid \mathcal{H}_{t-1}] + \Gamma_t \tag{55}$$

constrains only the conditional average over the possible communications of the next round. A particular realized message can move the posterior toward regions where $Z^\star$ is more uncertain.

Thus there is no contradiction between

$$\mathbb{E}[V_t \mid \mathcal{H}_{t-1}] \le V_{t-1}$$

and

$$V_t > V_{t-1}$$

on an observed trajectory.

B Feature-History Uncertainty Flow

This section proves that the feature-history identities bound the error of the derived readouts $\widehat{V}_\ell = 1 - \widehat{\mu}_\ell^2$ and $\widehat{J}_\ell = (\widehat{\mu}_\ell - \widehat{\mu}_{\ell-1})^2$ by the error of the single fitted mean. Throughout, $\mathcal{H}_\ell^F = \sigma(F_0, \dots, F_\ell)$, $m_\ell = \mathbb{E}[Z^\star \mid \mathcal{H}_\ell^F]$, $v_\ell = \mathrm{Var}(Z^\star \mid \mathcal{H}_\ell^F)$, and $\gamma_\ell = \mathbb{E}[(m_\ell - m_{\ell-1})^2 \mid \mathcal{H}_{\ell-1}^F]$, as in Equation (12); all identities hold almost surely. CUE fits exactly one regression, Equation (13); no quantity in this section is a separately trained model.

B.1 Proof of the Feature-History Identities

The feature-relative and full-history spreads are related by the law of total variance,

$$v_\ell = \mathbb{E}[V_\ell \mid \mathcal{H}_\ell^F] + \mathrm{Var}(\mu_\ell \mid \mathcal{H}_\ell^F), \tag{56}$$

so $v_\ell \geq \mathbb{E}[V_\ell \mid \mathcal{H}_\ell^F]$: the feature-relative spread adds the part of the full-history mean that the features do not pin down.

Proof. Since $\mathbb{E}[(Z^\star)^2] < \infty$ and $\mathcal{H}_{\ell-1}^F \subseteq \mathcal{H}_\ell^F$, the tower property gives $\mathbb{E}[m_\ell \mid \mathcal{H}_{\ell-1}^F] = m_{\ell-1}$, so $\{m_\ell\}$ is a square-integrable martingale. For $\ell \geq 1$, write $Z^\star - m_{\ell-1} = (Z^\star - m_\ell) + (m_\ell - m_{\ell-1})$. The increment $m_\ell - m_{\ell-1}$ is $\mathcal{H}_\ell^F$-measurable and $\mathbb{E}[Z^\star - m_\ell \mid \mathcal{H}_\ell^F] = 0$, so the cross term vanishes after conditioning on $\mathcal{H}_\ell^F$ (both factors are square integrable, so the product is integrable by Cauchy–Schwarz). Hence

$$v_{\ell-1} = \mathbb{E}\big[\mathbb{E}[(Z^\star - m_\ell)^2 \mid \mathcal{H}_\ell^F] \,\big|\, \mathcal{H}_{\ell-1}^F\big] + \mathbb{E}[(m_\ell - m_{\ell-1})^2 \mid \mathcal{H}_{\ell-1}^F] = \mathbb{E}[v_\ell \mid \mathcal{H}_{\ell-1}^F] + \gamma_\ell.$$

Conditioning on $\mathcal{H}_0^F$ and summing over ℓ gives the trajectory identity. The proofs of Proposition 3.2, Corollary 3.3, Equation (8), and Proposition A.2 in Appendix A use only that the conditioning σ-fields are nested, that $Z^\star$ is square integrable, and the one-step identity just proved. They therefore hold with $(m_\ell, v_\ell, \gamma_\ell, \mathcal{H}_\ell^F)$ in place of $(\mu_\ell, V_\ell, \Gamma_\ell, \mathcal{H}_\ell)$.

For Equation (56), $\mathcal{H}_\ell^F \subseteq \mathcal{H}_\ell$ and the law of total variance give $\mathrm{Var}(Z^\star \mid \mathcal{H}_\ell^F) = \mathbb{E}[\mathrm{Var}(Z^\star \mid \mathcal{H}_\ell) \mid \mathcal{H}_\ell^F] + \mathrm{Var}(\mathbb{E}[Z^\star \mid \mathcal{H}_\ell] \mid \mathcal{H}_\ell^F)$, which is Equation (56). If μ_j is $\mathcal{H}_j^F$-measurable for $j \in \{\ell - 1, \ell\}$, then $m_j = \mathbb{E}[\mu_j \mid \mathcal{H}_j^F] = \mu_j$, the second term vanishes, and $\gamma_\ell = \mathbb{E}\big[\mathbb{E}[(\mu_\ell - \mu_{\ell-1})^2 \mid \mathcal{H}_{\ell-1}] \,\big|\, \mathcal{H}_{\ell-1}^F\big] = \mathbb{E}[\Gamma_\ell \mid \mathcal{H}_{\ell-1}^F]$ by the tower property. □

B.2 Plug-in Error of the Derived Readouts

Condition on the training data and let g_ℓ be a mean predictor that is a function of $F_0, \dots, F_\ell$ and was trained without the evaluated query. The frozen $\widehat{\mu}$ of Equation (13), read on a held-out execution, is such a predictor. Because every other reported quantity is a deterministic function of $\widehat{\mu}$, no second model is trained on targets computed from $\widehat{\mu}$, and the train/test split alone keeps g_ℓ independent of the evaluated query; no out-of-fold construction is needed. At message ℓ, the input F_ℓ is computed from the prefix Π_ℓ, which includes the query and every history message $\iota_1, \dots, \iota_\ell$; any information lost in encoding the prefix, for example through left truncation at the token budget or pooling, enters the error ϵ_ℓ below. Let $\eta_\ell = g_\ell - m_\ell$, $\epsilon_\ell = \|\eta_\ell\|_{L^2}$, $\Delta m_\ell = m_\ell - m_{\ell-1}$ and $\Delta\eta_\ell = \eta_\ell - \eta_{\ell-1}$.

The realized resolution $\widehat{J}_\ell = (g_\ell - g_{\ell-1})^2$ expands as $(\Delta m_\ell + \Delta\eta_\ell)^2$, giving

$$\mathbb{E}[\widehat{J}_\ell \mid \mathcal{H}_{\ell-1}^F] = \gamma_\ell + 2\,\mathbb{E}[\Delta m_\ell\, \Delta\eta_\ell \mid \mathcal{H}_{\ell-1}^F] + \mathbb{E}[(\Delta\eta_\ell)^2 \mid \mathcal{H}_{\ell-1}^F], \tag{57}$$

so the resolution read off the fitted mean can be biased in either direction through the cross term. The residual uncertainty is not a separate regression but the derived $\widehat{V}_\ell = 1 - g_\ell^2$. By Equation (10), $v_\ell = 1 - m_\ell^2$ exactly, so

$$\widehat{V}_\ell - v_\ell = -\big(2m_\ell\eta_\ell + \eta_\ell^2\big), \qquad \big\|\widehat{V}_\ell - v_\ell\big\|_{L^1} \leq 2\,\|m_\ell\|_{L^2}\epsilon_\ell + \epsilon_\ell^2 \leq 2\epsilon_\ell + \epsilon_\ell^2. \tag{58}$$

In particular $\widehat{V}_\ell$ and $1 - \widehat{\mu}_\ell^2$ are the same quantity, with correlation one by construction; the spread carries no regression error beyond that of the mean. Because $\widehat{J}_\ell$ is computed from the observed trajectory rather than regressed on $F_{\ell-1}$, no representation term arises.

B.3 Error Transfer from Mean Estimation to Attribution

Proposition B.1 (Error Transfer)**.** *Condition on the training data. Let g_ℓ be as in Appendix B.2, with $s_\ell = \epsilon_\ell + \epsilon_{\ell-1}$, and let $\widehat{J}_\ell = (g_\ell - g_{\ell-1})^2$ and $\widehat{V}_\ell = 1 - g_\ell^2$ be the derived readouts. Then*

$$\left|\mathbb{E}[\widehat{J}_\ell] - \mathbb{E}[\gamma_\ell]\right| \le 2\sqrt{\mathbb{E}[\gamma_\ell]}\, s_\ell + s_\ell^2, \qquad \left\|\widehat{V}_\ell - v_\ell\right\|_{L^1} \le 2\,\|m_\ell\|_{L^2}\epsilon_\ell + \epsilon_\ell^2. \tag{59}$$

Consequently, for every group $P \subseteq \mathcal{I}$ with marginal share $\gamma[P] = \sum_{\iota_\ell \in P} \mathbb{E}[\gamma_\ell]$,

$$\left|\sum_{\iota_\ell \in P} \mathbb{E}[\widehat{J}_\ell] - \gamma[P]\right| \le \sum_{\iota_\ell \in P} \left(2\sqrt{\mathbb{E}[\gamma_\ell]}\, s_\ell + s_\ell^2\right).$$

Proof. Since $\mathbb{E}[\gamma_\ell] = \mathbb{E}[(\Delta m_\ell)^2]$, $\mathbb{E}[\widehat{J}_\ell] - \mathbb{E}[\gamma_\ell] = \mathbb{E}[(\Delta m_\ell + \Delta\eta_\ell)^2 - (\Delta m_\ell)^2] = \mathbb{E}[\Delta\eta_\ell(2\Delta m_\ell + \Delta\eta_\ell)]$, whose absolute value is at most $2\|\Delta\eta_\ell\|_2\|\Delta m_\ell\|_2 + \|\Delta\eta_\ell\|_2^2$ by Cauchy–Schwarz. Minkowski's inequality gives $\|\Delta\eta_\ell\|_2 \le s_\ell$, and $\|\Delta m_\ell\|_2^2 = \mathbb{E}[\gamma_\ell]$. The variance bound is Equation (58), and the group bound follows by summation. □

The bound has two sources of error, both from the single fitted mean: its interaction with the resolution scale, which enters linearly, and its own size, which enters quadratically. There is no regression error of a second model, because none is fitted. It is an error-transfer statement; rates in the sample size would also need learning assumptions on the network class. The corresponding full-history statement replaces $(m_\ell, \gamma_\ell, \mathcal{H}_\ell^F)$ by $(\mu_\ell, \Gamma_\ell, \mathcal{H}_\ell)$ and adds the representation term of Equation (56), because no estimator observes $\mathcal{H}_\ell$ itself.

C Additional Experimental Results

This section gives further details on the experiments. We test the baselines with the parameters from their original papers and run a grid search when those defaults do not give the best performance; we tune our own method by grid search as well. All code is written in Python 3.10.13 and PyTorch 2.8.0, and all experiments run on a Linux server with 384 AMD EPYC 7313 CPUs and 4 NVIDIA H200 NVL GPUs with 141 GB of memory each.

C.1 Multi-Agent System Introduction

We use the three orchestrations of Section 5.1, 14 configurations per backbone. On the coding benchmarks (HumanEval, MBPP) we run SelfPlanner (Jiang et al., 2024) (SP), a two-agent pipeline in which a planner writes a plan and a coder writes the program from it, and GPT-Swarm (Zhuge et al., 2024) with a fully connected graph of three (S3) or five (S5) agents. On the QA benchmarks (MATH500, MMLU) we run multi-agent debate (Du et al., 2024) with $k \in \{2, 3, 5, 7\}$ agents and two rounds, in which every agent answers in round 1, reads the other agents' answers and revises in round 2, and the delivered answer is the majority vote over round-2 answers. This gives 2×3 coding and 2×4 debate configurations.

C.2 Implementation Details

Features. Hidden states are read at four evenly spaced layers of the backbone, $\{20, 40, 60, 80\}$ for Llama-3.1-70B, $\{12, 24, 36, 48\}$ for Qwen3-30B, and $\{16, 32, 48, 64\}$ for Qwen2.5-Coder-32B. Each is pooled over the tokens of Π_ℓ and ℓ_2-normalized, and ϕ in Equation (11) is their average, ℓ_2-normalized again, so no layer is selected. Prefixes are truncated from the left at 8192 tokens for debate and 4096 tokens for the coding orchestrations.

Network and training. CUE trains a single feedforward network f_ψ (Equation (13)) with two hidden layers of width w, ReLU activations, and dropout 0.1. It is trained in full batch with Adam (learning rate 10^{-3}) for 300 epochs on the squared loss pooled over positions, plus the penalty $\lambda_c \mathcal{R}_c$.

Selection. For each split, the grid crosses the width $w \in \{32, 64, 128, 256, 512\}$ with $\lambda_c \in \{10^{-3}, 10^{-2}, 10^{-1}, 1\}$. The pooling rule $\varrho \in \{\text{mean}, \text{last}, \max\}$ and the weight decay $\in \{10^{-2}, 10^{-1}\}$ are selected once per split on the validation set and then held fixed. The reported model is the cell with the best validation AUROC of $\widehat{\mu}_M$, with ties broken toward the smaller λ_c. The *w/o*

constraint ablation (Table 5) repeats the width search with $\lambda_c = 0$. The splits (50% test, then 80/20 train/validation, six seeds) are those of the baselines, and test queries never enter fitting or selection.

C.3 Additional Metrics

We report the Brier score (Table 6 and Table 7) and NLL (Table 8 and Table 9), two standard metrics for probabilistic predictions. The Brier score is the mean squared difference between the predicted probability and the observed outcome, and NLL is the negative log-likelihood of the observed outcomes under the predicted probabilities. ECE measures only the agreement between stated confidence and empirical accuracy within a bin. It is blind to whether the confidences separate correct from incorrect executions at all.

Table 6: **Llama-3.1-70B-Instruct.** Brier score↓ ($\times 100$), across 14 multi-agent configurations. SP=SelfPlanner, S3/S5=GPT-Swarm with 3/5 agents, k=number of debate agents. **Bold**=best (lowest), underline=second per column.

	HumanEval			MBPP			MATH500				MMLU				
Method	SP	S3	S5	SP	S3	S5	$k2$	$k3$	$k5$	$k7$	$k2$	$k3$	$k5$	$k7$	Avg.
PE	19.53 ±0.01	21.31 ±0.03	21.33 ±0.02	20.85 ±0.01	23.24 ±0.05	23.27 ±0.02	23.99 ±0.09	21.66 ±0.26	20.54 ±0.24	20.24 ±0.33	10.85 ±0.01	11.63 ±0.05	10.14 ±0.13	11.31 ±0.04	18.56
SE	20.13 ±0.33	21.53 ±0.23	21.32 ±0.24	**20.65** ±0.27	23.47 ±0.21	23.42 ±0.08	15.80 ±1.12	**14.58** ±0.49	14.24 ±0.85	14.29 ±0.76	10.04 ±0.17	9.85 ±0.42	9.82 ±0.10	**9.97** ±0.56	16.37
SAR	19.43 ±0.10	21.30 ±0.09	21.42 ±0.05	20.89 ±0.04	23.04 ±0.13	23.05 ±0.15	23.50 ±0.24	20.15 ±0.37	18.94 ±0.44	18.71 ±0.40	10.82 ±0.03	11.60 ±0.09	10.13 ±0.13	11.34 ±0.02	18.17
SAUP	19.47 ±0.04	21.33 ±0.01	21.31 ±0.01	20.86 ±0.01	23.29 ±0.04	23.17 ±0.05	22.10 ±0.46	19.96 ±0.52	18.49 ±0.53	18.31 ±0.78	10.64 ±0.17	11.02 ±0.22	10.04 ±0.14	10.97 ±0.21	17.93
LARS	19.49 ±0.06	21.31 ±0.05	21.31 ±0.08	20.70 ±0.06	22.92 ±0.20	22.80 ±0.17	23.69 ±0.21	20.16 ±0.43	18.02 ±0.56	17.97 ±0.77	10.83 ±0.03	11.54 ±0.03	10.08 ±0.11	11.23 ±0.10	18.00
Sup	19.22 ±0.38	18.21 ±0.81	17.82 ±0.55	20.74 ±0.11	22.98 ±0.13	22.87 ±0.07	23.38 ±0.23	17.48 ±0.26	17.02 ±0.68	16.38 ±0.64	10.75 ±0.04	11.46 ±0.11	9.62 ±0.09	10.40 ±0.26	17.02
PropUQ	19.51 ±0.02	21.26 ±0.04	20.94 ±0.09	20.86 ±0.01	23.24 ±0.06	23.06 ±0.16	23.30 ±0.16	21.58 ±0.17	20.63 ±0.20	20.60 ±0.33	10.84 ±0.01	11.61 ±0.04	10.16 ±0.12	11.38 ±0.02	18.50
CUE	**18.49** ±0.63	**9.02** ±0.88	**11.50** ±4.54	20.95 ±0.85	**22.89** ±1.13	**20.70** ±3.50	**13.22** ±1.38	15.01 ±3.68	**11.24** ±0.16	**13.60** ±1.64	**9.35** ±0.80	**8.93** ±0.45	**9.33** ±0.59	10.15 ±1.32	**13.88**

Table 7: **Qwen models.** Brier score↓ ($\times 100$), across 14 multi-agent configurations. HumanEval and MBPP use Qwen2.5-Coder-32B-Instruct; MATH500 and MMLU use Qwen3-30B-A3B-Instruct-2507. SP=SelfPlanner, S3/S5=GPT-Swarm with 3/5 agents, k=number of debate agents. **Bold**=best (lowest), underline=second per column.

	HumanEval			MBPP			MATH500				MMLU				
Method	SP	S3	S5	SP	S3	S5	$k2$	$k3$	$k5$	$k7$	$k2$	$k3$	$k5$	$k7$	Avg.
PE	16.17 ±0.03	3.33 ±0.43	3.33 ±0.43	15.10 ±0.03	20.65 ±0.02	20.63 ±0.02	20.47 ±0.08	19.77 ±0.07	19.03 ±0.08	18.43 ±0.04	12.00 ±0.03	11.03 ±0.10	10.12 ±0.12	11.00 ±0.13	14.36
SE	16.26 ±0.18	3.33 ±0.43	3.33 ±0.43	15.34 ±0.11	20.72 ±0.11	20.80 ±0.09	11.62 ±1.09	8.95 ±0.89	8.45 ±0.99	9.89 ±0.92	11.39 ±0.23	10.21 ±0.39	9.55 ±0.35	**10.15** ±0.43	11.43
SAR	16.24 ±0.05	3.33 ±0.43	3.33 ±0.43	15.15 ±0.00	20.63 ±0.11	20.48 ±0.12	20.18 ±0.31	19.39 ±0.12	18.60 ±0.15	18.06 ±0.26	12.05 ±0.01	11.14 ±0.11	10.18 ±0.13	11.07 ±0.13	14.27
SAUP	16.17 ±0.02	3.33 ±0.43	3.33 ±0.43	15.13 ±0.05	20.63 ±0.03	20.62 ±0.03	19.74 ±0.22	18.96 ±0.31	18.44 ±0.16	17.76 ±0.25	11.77 ±0.08	10.90 ±0.13	10.00 ±0.09	10.99 ±0.15	14.13
LARS	16.19 ±0.02	3.33 ±0.43	3.33 ±0.43	15.14 ±0.02	20.62 ±0.05	20.62 ±0.04	20.35 ±0.25	19.61 ±0.13	18.67 ±0.20	17.68 ±0.29	11.85 ±0.05	10.91 ±0.11	9.96 ±0.11	10.77 ±0.14	14.22
Sup	16.18 ±0.05	3.34 ±0.43	3.32 ±0.42	15.08 ±0.04	20.23 ±0.05	20.17 ±0.17	15.31 ±0.69	13.00 ±0.73	13.07 ±0.54	12.89 ±0.55	11.77 ±0.10	10.62 ±0.13	**9.31** ±0.32	10.40 ±0.22	12.48
PropUQ	16.18 ±0.01	3.33 ±0.43	3.33 ±0.43	15.14 ±0.01	20.61 ±0.05	20.48 ±0.11	20.00 ±0.15	19.33 ±0.20	18.69 ±0.11	18.09 ±0.14	11.90 ±0.05	10.99 ±0.11	10.09 ±0.12	10.99 ±0.13	14.23
CUE	**15.67** ±0.54	**1.73** ±0.24	**1.70** ±0.25	**14.95** ±0.17	**16.42** ±1.75	**16.03** ±0.76	**7.09** ±2.30	**6.14** ±2.12	**7.20** ±2.16	**7.24** ±1.69	**10.74** ±0.69	**9.88** ±0.59	9.57 ±0.80	10.17 ±0.57	**9.61**

Table 8: **Llama-3.1-70B-Instruct.** NLL↓ ($\times 100$), across 14 multi-agent configurations. SP=SelfPlanner, S3/S5=GPT-Swarm with 3/5 agents, k=number of debate agents. **Bold**=best (lowest), underline=second per column.

	HumanEval			MBPP			MATH500				MMLU				
Method	SP	S3	S5	SP	S3	S5	$k2$	$k3$	$k5$	$k7$	$k2$	$k3$	$k5$	$k7$	Avg.
PE	57.94 ±0.03	61.73 ±0.08	61.78 ±0.04	60.78 ±0.03	65.75 ±0.10	65.81 ±0.04	67.28 ±0.19	62.41 ±0.54	59.94 ±0.53	59.25 ±0.73	37.44 ±0.07	39.25 ±0.19	35.42 ±0.38	38.39 ±0.16	55.23
SE	59.51 ±0.89	62.30 ±0.59	61.78 ±0.57	**60.28** ±0.65	66.26 ±0.46	66.15 ±0.18	**49.44** ±2.63	**46.04** ±1.11	45.24 ±2.19	**45.48** ±1.70	**34.53** ±0.57	**33.90** ±1.29	34.23 ±0.46	**33.81** ±1.83	49.93
SAR	57.68 ±0.26	61.71 ±0.22	61.99 ±0.11	60.87 ±0.10	65.31 ±0.27	65.34 ±0.32	66.26 ±0.51	59.06 ±0.81	56.25 ±1.01	55.65 ±0.89	37.27 ±0.13	39.13 ±0.36	35.38 ±0.38	38.51 ±0.11	54.32
SAUP	57.79 ±0.10	61.79 ±0.03	61.73 ±0.03	60.80 ±0.02	65.84 ±0.09	65.61 ±0.12	63.33 ±0.98	58.73 ±1.32	55.23 ±1.20	54.88 ±1.79	36.60 ±0.66	36.89 ±0.83	34.89 ±0.45	36.88 ±0.86	53.64
LARS	57.84 ±0.16	61.72 ±0.12	61.74 ±0.20	60.42 ±0.13	**65.07** ±0.42	**64.81** ±0.35	66.67 ±0.43	59.18 ±0.94	54.33 ±1.28	54.17 ±1.74	37.35 ±0.14	38.87 ±0.14	35.13 ±0.30	38.05 ±0.41	53.95
Sup	57.14 ±0.96	54.92 ±1.82	54.10 ±1.24	60.50 ±0.27	65.20 ±0.27	64.95 ±0.15	66.01 ±0.47	52.78 ±0.57	51.81 ±1.63	50.09 ±1.55	36.98 ±0.16	38.63 ±0.44	**32.94** ±0.47	34.84 ±1.01	51.49
PropUQ	57.88 ±0.06	61.62 ±0.10	60.87 ±0.20	60.79 ±0.01	65.75 ±0.12	65.36 ±0.35	65.85 ±0.34	62.21 ±0.38	60.12 ±0.45	60.04 ±0.73	37.40 ±0.05	39.19 ±0.17	35.56 ±0.34	38.70 ±0.10	55.10
CUE	**57.13** ±5.93	**31.00** ±3.88	**36.37** ±12.38	87.19 ±25.70	83.15 ±12.11	76.11 ±25.06	52.72 ±5.54	49.24 ±9.04	**36.42** ±1.73	46.01 ±6.51	35.86 ±6.35	36.54 ±10.81	34.05 ±3.95	35.88 ±6.78	**49.83**

Table 10: **Llama-3.1-70B-Instruct.** ECE↓ ($\times 100$), across 14 multi-agent configurations. SP=SelfPlanner, S3/S5=GPT-Swarm with 3/5 agents, k=number of debate agents. **Bold**=best, underline=second per column. Baseline probabilities are cross-fitted Platt calibrations of their scores; CUE reports the raw $\hat{p} = (1 + \hat{\mu}_M)/2$ with no calibration step. ECE is a calibration-only criterion and does not reward discrimination: a score that collapses to the base rate is perfectly calibrated while ranking at chance, which is why the lowest-ECE columns here often belong to the weakest-AUROC baselines; see *Why ECE is reported last*, below.

	HumanEval			MBPP			MATH500				MMLU				
Method	SP	S3	S5	SP	S3	S5	$k2$	$k3$	$k5$	$k7$	$k2$	$k3$	$k5$	$k7$	Avg.
PE	**0.00** ±0.00	1.10 ±0.80	0.10 ±0.30	1.00 ±1.10	0.30 ±0.30	0.40 ±0.00	3.30 ±2.50	6.70 ±1.90	6.40 ±1.00	10.00 ±1.70	**0.00** ±0.00	0.10 ±0.20	0.10 ±0.10	**0.00** ±0.00	**2.11**
SE	5.10 ±1.70	2.60 ±0.80	4.60 ±2.60	3.30 ±1.60	2.00 ±0.80	1.30 ±1.00	6.10 ±2.10	6.90 ±1.40	7.60 ±1.10	8.80 ±1.30	4.20 ±0.60	3.30 ±0.70	1.80 ±1.00	4.40 ±0.90	4.43
SAR	0.60 ±0.90	1.50 ±1.40	1.50 ±2.00	2.10 ±1.20	3.90 ±2.10	0.80 ±0.50	**3.10** ±1.30	6.80 ±1.10	8.10 ±1.20	11.30 ±0.80	0.20 ±0.20	0.20 ±0.20	0.50 ±0.40	0.10 ±0.10	2.91
SAUP	0.00 ±0.00	**0.00** ±0.00	**0.00** ±0.00	1.30 ±0.90	**0.10** ±0.20	**0.00** ±0.00	6.10 ±1.60	7.90 ±0.90	7.50 ±1.20	8.60 ±1.40	1.30 ±0.60	2.30 ±0.90	3.00 ±0.70	3.00 ±1.10	2.94
LARS	0.00 ±0.00	0.80 ±0.50	1.30 ±0.90	6.40 ±1.50	3.60 ±1.40	3.30 ±1.30	5.50 ±1.90	8.30 ±0.90	9.70 ±1.10	11.00 ±0.50	0.30 ±0.20	0.20 ±0.10	1.30 ±0.60	0.30 ±0.40	3.71
Sup	3.10 ±2.40	14.10 ±1.70	13.80 ±2.20	3.40 ±1.10	1.70 ±1.60	1.70 ±1.20	5.10 ±1.50	7.20 ±1.50	6.20 ±1.20	9.10 ±1.40	0.30 ±0.30	1.20 ±1.20	4.00 ±1.10	4.30 ±1.40	5.37
PropUQ	0.00 ±0.00	0.60 ±0.50	7.10 ±2.30	**0.90** ±0.50	0.40 ±0.30	1.70 ±1.60	8.40 ±1.40	8.60 ±1.20	7.40 ±1.10	**8.10** ±1.60	0.00 ±0.00	**0.00** ±0.00	**0.00** ±0.00	0.00 ±0.00	3.09
CUE	8.70 ±4.00	6.60 ±1.90	7.80 ±3.30	12.70 ±5.50	14.30 ±6.80	15.80 ±5.90	7.00 ±1.40	**6.20** ±2.30	**6.10** ±1.20	9.30 ±2.20	5.30 ±2.40	5.20 ±1.70	4.50 ±0.80	5.60 ±1.50	8.22

Table 11: **Qwen2.5-Coder-32B-Instruct (code) and Qwen3-30B-A3B-Instruct-2507 (QA).** ECE↓ ($\times 100$), across 14 multi-agent configurations. SP=SelfPlanner, S3/S5=GPT-Swarm with 3/5 agents, k=number of debate agents. **Bold**=best, underline=second per column. Baseline probabilities are cross-fitted Platt calibrations of their scores; CUE reports the raw $\hat{p} = (1+\hat{\mu}_M)/2$ with no calibration step. ECE is a calibration-only criterion and does not reward discrimination: a score that collapses to the base rate is perfectly calibrated while ranking at chance, which is why the lowest-ECE columns here often belong to the weakest-AUROC baselines; see *Why ECE is reported last*, below.

	HumanEval			MBPP			MATH500				MMLU				
Method	SP	S3	S5	SP	S3	S5	$k2$	$k3$	$k5$	$k7$	$k2$	$k3$	$k5$	$k7$	Avg.
PE	2.30 ±1.40	**0.00** ±0.00	**0.00** ±0.00	0.20 ±0.20	**0.30** ±0.70	**0.60** ±0.70	**0.90** ±0.50	**0.10** ±0.10	**0.00** ±0.00	**0.00** ±0.00	**0.00** ±0.00	**0.00** ±0.00	0.40 ±0.40	**0.00** ±0.00	**0.34**
SE	4.30 ±2.90	0.00 ±0.00	0.00 ±0.00	3.20 ±1.90	3.00 ±1.50	2.40 ±1.80	5.80 ±0.30	5.60 ±0.50	5.60 ±1.70	4.50 ±0.70	2.90 ±0.60	4.10 ±0.80	3.10 ±1.20	3.00 ±1.20	3.39
SAR	4.50 ±2.40	0.00 ±0.00	0.00 ±0.00	**0.00** ±0.00	1.70 ±1.30	3.40 ±1.20	5.60 ±1.10	3.70 ±1.40	3.00 ±1.10	1.30 ±1.70	0.00 ±0.00	0.00 ±0.00	**0.30** ±0.20	0.10 ±0.10	1.69
SAUP	2.20 ±1.10	0.00 ±0.00	0.00 ±0.00	2.20 ±1.30	0.50 ±0.50	1.30 ±0.90	8.80 ±1.40	6.70 ±1.60	3.30 ±1.00	3.30 ±1.70	1.10 ±0.60	1.20 ±0.50	2.10 ±1.00	0.60 ±0.50	2.38
LARS	2.70 ±0.60	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	1.40 ±1.90	2.20 ±1.70	2.50 ±2.20	2.20 ±1.90	3.60 ±2.20	4.80 ±2.40	0.90 ±0.50	0.70 ±0.30	2.10 ±1.10	1.60 ±0.60	1.76
Sup	**2.10** ±1.00	0.00 ±0.00	0.00 ±0.00	1.20 ±0.80	5.50 ±1.30	5.30 ±2.40	7.40 ±1.40	9.10 ±1.90	9.80 ±1.80	7.80 ±1.60	1.80 ±0.60	2.90 ±1.40	4.10 ±0.70	3.30 ±1.00	4.31
PropUQ	2.40 ±0.00	0.00 ±0.00	0.00 ±0.00	0.40 ±0.20	2.20 ±2.20	3.70 ±1.90	9.40 ±1.40	4.90 ±1.30	1.00 ±0.90	0.50 ±0.60	0.00 ±0.00	0.10 ±0.00	0.60 ±0.40	0.00 ±0.00	1.80
CUE	7.60 ±4.40	1.70 ±0.40	1.50 ±0.30	3.30 ±1.80	11.10 ±3.60	11.50 ±1.60	6.10 ±1.20	5.60 ±2.00	5.90 ±1.00	6.70 ±1.40	5.70 ±2.50	5.70 ±1.50	3.70 ±1.80	5.90 ±2.40	5.86

Table 9: **Qwen models.** NLL↓ ($\times 100$), across 14 multi-agent configurations. HumanEval and MBPP use Qwen2.5-Coder-32B-Instruct; MATH500 and MMLU use Qwen3-30B-A3B-Instruct-2507. SP=SelfPlanner, S3/S5=GPT-Swarm with 3/5 agents, k=number of debate agents. **Bold**=best (lowest), underline=second per column.

	HumanEval			MBPP			MATH500				MMLU				
Method	SP	S3	S5	SP	S3	S5	$k2$	$k3$	$k5$	$k7$	$k2$	$k3$	$k5$	$k7$	Avg.
PE	**50.44** ±0.11	14.99 ±1.58	14.99 ±1.58	47.91 ±0.10	60.34 ±0.04	60.30 ±0.04	59.96 ±0.18	58.45 ±0.15	56.85 ±0.17	55.52 ±0.09	40.32 ±0.13	37.77 ±0.27	35.34 ±0.35	37.64 ±0.39	45.06
SE	50.74 ±0.60	14.97 ±1.57	14.97 ±1.57	48.74 ±0.39	60.54 ±0.25	60.71 ±0.23	39.45 ±2.64	31.85 ±2.58	30.71 ±2.93	34.53 ±2.51	**38.43** ±0.68	**35.05** ±1.15	33.32 ±1.15	**35.03** ±1.23	**37.79**
SAR	50.65 ±0.16	14.98 ±1.57	14.99 ±1.58	48.09 ±0.01	60.30 ±0.27	59.93 ±0.29	59.26 ±0.74	57.52 ±0.27	55.77 ±0.37	54.55 ±0.70	40.55 ±0.04	38.27 ±0.30	35.65 ±0.38	37.93 ±0.37	44.89
SAUP	50.46 ±0.07	14.99 ±1.58	14.99 ±1.58	47.97 ±0.17	60.30 ±0.07	60.26 ±0.06	58.27 ±0.50	56.54 ±0.73	55.39 ±0.38	53.81 ±0.64	39.40 ±0.32	37.21 ±0.44	34.79 ±0.26	37.54 ±0.49	44.42
LARS	50.53 ±0.05	14.99 ±1.58	15.00 ±1.58	48.04 ±0.06	60.28 ±0.13	60.26 ±0.10	59.69 ±0.58	58.08 ±0.31	55.99 ±0.46	53.63 ±0.75	39.70 ±0.22	37.25 ±0.35	34.59 ±0.36	36.63 ±0.49	44.62
Sup	50.48 ±0.15	15.15 ±1.57	14.77 ±1.57	47.84 ±0.12	**59.30** ±0.13	**59.15** ±0.42	47.79 ±1.70	41.98 ±1.93	42.12 ±1.50	41.33 ±1.50	39.28 ±0.45	36.19 ±0.49	**31.93** ±1.26	35.43 ±0.78	40.20
PropUQ	50.48 ±0.03	14.99 ±1.58	14.99 ±1.57	48.04 ±0.05	60.24 ±0.13	59.95 ±0.26	58.85 ±0.34	57.39 ±0.48	55.99 ±0.28	54.63 ±0.37	39.92 ±0.19	37.60 ±0.31	35.21 ±0.33	37.57 ±0.37	44.70
CUE	57.94 ±19.82	**6.48** ±0.93	**6.42** ±0.55	**47.39** ±0.57	93.99 ±58.74	67.31 ±10.31	**35.58** ±7.24	**26.37** ±8.39	**30.18** ±8.04	**26.16** ±4.53	39.10 ±6.57	35.92 ±4.56	33.92 ±2.26	35.42 ±2.46	38.73

Why ECE is reported last. ECE measures only the agreement between stated confidence and empirical accuracy within a bin; it is blind to whether the confidences separate correct from incorrect executions at all. A predictor that ignores its input and returns the base rate attains ECE $\to 0$ while its AUROC is exactly 0.5. Averaged over the 28 configurations this is the regime several baselines occupy: PE attains the best mean ECE (1.2×10^{-2}) with the second-lowest mean AUROC (0.627; SAR is lowest at 0.611), and 36 of the 196 baseline cells pair ECE $< 2 \times 10^{-2}$ with AUROC < 0.60. CUE sits at the other end – mean ECE 7.0×10^{-2} with mean AUROC 0.823 – because $\hat{p} = (1 + \hat{\mu}_M)/2$ is trained as a ranking score under squared loss and is never rescaled. The ordering in Tables 10 and 11 should therefore be read as a statement about calibration alone, and alongside Tables 3 and 4.

C.4 Efficiency Analysis

We measure wall-clock time for CUE and all seven baselines on the Llama-3.1-70B debate configurations (500 executions each). For CUE we report two numbers separately. Feature extraction, one forward pass of the frozen backbone per prefix, takes about 2.2×10^3 s per configuration and is done once and cached. Fitting and scoring the network on the cached features then takes 1.3–22.5 s. The baselines run on stored generations: PE, SAUP, PropUQ, and Sup take 0.01–0.24 s, SE 18–63 s, LARS 22–93 s, and SAR 14–391 s. LARS is not a pure output-probability method: it runs a trained transformer scorer over the generated tokens and their probabilities, which is why it costs tens of seconds. With extraction included, CUE is the most expensive method in wall-clock time; without it, its scoring cost lies between the cheap scores (PE, SAUP, PropUQ, Sup) and the sampling- and model-based ones (SE, SAR, LARS). Table 12 lists the scoring time of every method, and Figure 5 plots AUROC against *scoring time*, which for CUE excludes the one-time feature extraction. On that axis CUE lies on the Pareto frontier in all eight debate configurations. If the 2.2×10^3 s extraction is charged to CUE, it would sit to the right of every baseline and would no longer be on the frontier, so the Pareto comparison should be read as a comparison of scoring cost given cached features. Extraction could reuse the forward passes the backbone already performs while the system runs, which we have not implemented.

Scalability. Figure 4 times fitting and test scoring of CUE, the same quantity as above, on the cached Llama-3.1-70B MATH500 debate features for $k \in \{2, 3, 5, 7\}$ (6 to 16 prefixes per execution). We train on 20% to 100% of the training split of one split (40 to 200 executions) with one fixed configuration (two hidden layers of width 128, $\lambda_c = 0.01$), and report the median of three runs on an otherwise idle GPU; feature loading is excluded. At the full split the time is 3.1, 4.4, 6.7, and 10.6 s for $k = 2, 3, 5, 7$, close to linear in the number of prefixes, and going from 20% to 100% of the training data raises it by 14–35%.

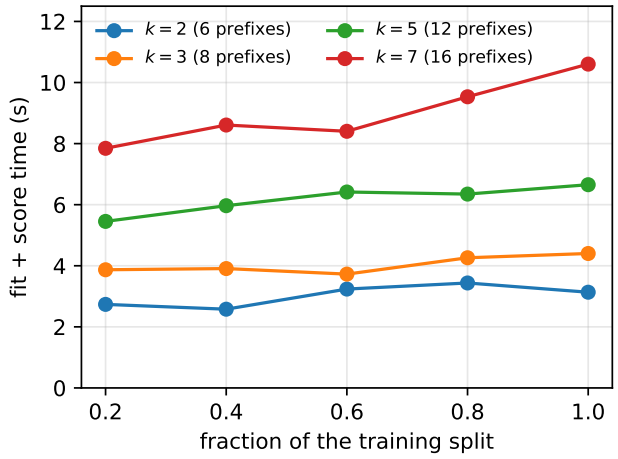

Figure 4: Fitting and scoring time of CUE (median of three runs) against the fraction of the training split on Llama-3.1-70B MATH500 debate with k agents, for one fixed network configuration on cached features.

D Additional Case Studies

All case studies share one setup. The backbone is Llama-3.1-70B-Instruct, and the orchestration is two-round multi-agent debate on MATH500 and MMLU with $k \in \{2, 3, 5, 7\}$ agents, which gives 8 configurations. For every configuration and each of the six seeds, we refit the CUE model reported in Table 1 (the cell selected on validation, refit on the same training split, which reproduces the reported test AUROC exactly) and evaluate it on that seed's test split. Baselines are scored on the same test executions. Per-seed quantities are reported as mean±SD over the six seeds, and every claim across configurations uses an exact two-sided Wilcoxon signed-rank test over the 8 configuration values.

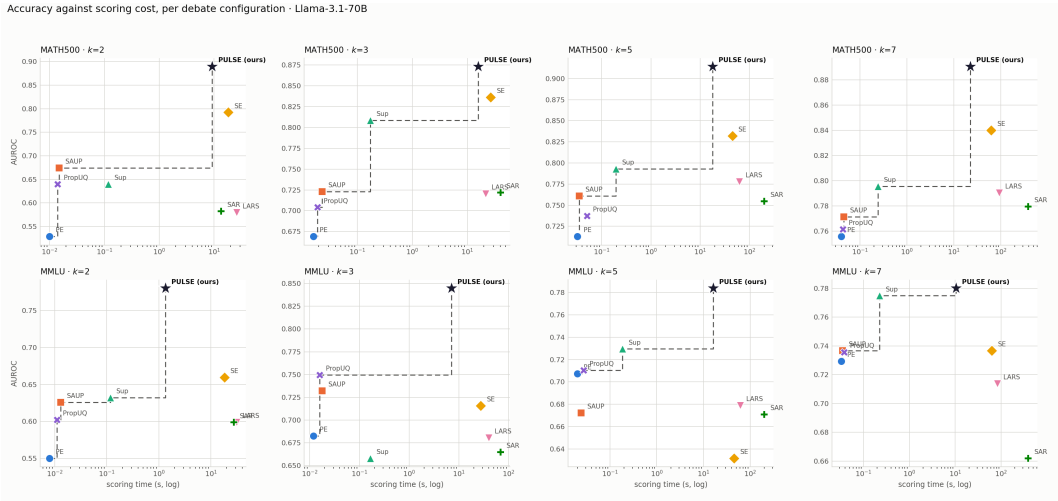

Figure 5: AUROC against scoring time (log scale) on the eight Llama-3.1-70B debate configurations (MATH500 and MMLU, $k \in \{2, 3, 5, 7\}$). The scoring time of CUE excludes the one-time backbone feature extraction (about 2.2×10^3 s per configuration); with extraction included, CUE is the most expensive method. The dashed line is the Pareto frontier.

Table 12: Scoring time (seconds) on the Llama-3.1-70B debate configurations (500 executions each). MATH500 and MMLU give the mean over $k \in \{2, 3, 5, 7\}$; Mean, Min, and Max are taken over all eight configurations. AUROC is the mean over the same configurations, and the last column is the mean time relative to CUE. The time of CUE is fitting and scoring on cached features and excludes the one-time backbone feature extraction (about 2.2×10^3 s per configuration).

Method	MATH500	MMLU	Mean	Min	Max	AUROC↑	×CUE
PE	0.024	0.018	0.021	0.008	0.040	66.7	0.002
SAUP	0.030	0.022	0.026	0.013	0.045	71.2	0.002
PropUQ	0.032	0.023	0.027	0.011	0.050	70.5	0.002
Sup	0.182	0.174	0.178	0.117	0.241	72.9	0.014
SE	38.38	38.28	38.33	18.20	63.11	75.5	3.05
LARS	50.82	53.78	52.30	21.64	92.74	69.3	4.16
SAR	160.94	172.07	166.50	13.57	391.22	67.9	13.25
CUE	16.38	8.74	12.56	1.34	22.48	**84.5**	1.00

Appendix D.1 completes the attribution study of Section 5.5, Appendix D.2 scores partial debates, Appendix D.3 tests realized amplification, Appendix D.4 collects diagnostics of the representation, and Appendix D.5 reports a causal intervention on the attributed messages.

D.1 Attribution: Resolution by Round and the Martingale Check

Where resolution is attributed. Table 13 gives the $\hat{J}$ share of round 1, round 2, and the vote. CUE fits one network, so there is no prospective $\hat{\Gamma}$; the shares are read off the realized $\hat{J}_\ell = (\hat{\mu}_\ell - \hat{\mu}_{\ell-1})^2$, whose conditional mean is γ_ℓ when the mean is exact. Round 1 carries more than its uniform share in 8 of 8 configurations ($p = 0.008$), and round 2 less in 8 of 8 ($p = 0.008$). The uniform share is only a reference point; the theory does not predict it.

Table 13: **Attributed resolution by round.** $\hat{J}$ share (mean±SD over 6 seeds) against the uniform share for that slot count.

		Round 1		Round 2		Vote	
Dataset	k	share	uniform	share	uniform	share	uniform
MATH500	2	0.615 ± 0.110	0.400	0.231 ± 0.043	0.400	0.129 ± 0.058	0.200
MATH500	3	0.636 ± 0.110	0.429	0.210 ± 0.113	0.429	0.145 ± 0.111	0.143
MATH500	5	0.570 ± 0.035	0.455	0.336 ± 0.034	0.455	0.088 ± 0.015	0.091
MATH500	7	0.659 ± 0.130	0.467	0.262 ± 0.102	0.467	0.046 ± 0.025	0.067
MMLU	2	0.619 ± 0.163	0.400	0.199 ± 0.076	0.400	0.178 ± 0.133	0.200
MMLU	3	0.560 ± 0.050	0.429	0.314 ± 0.033	0.429	0.125 ± 0.020	0.143
MMLU	5	0.620 ± 0.131	0.455	0.285 ± 0.128	0.455	0.093 ± 0.056	0.091
MMLU	7	0.583 ± 0.077	0.467	0.259 ± 0.177	0.467	0.157 ± 0.116	0.067

Round 1: the fitted mean follows message correctness. For each round-1 message we take the shift in the fitted mean at that slot. In every configuration it falls on messages graded wrong and rises on messages graded right, and its negative separates wrong from right messages with a mean AUROC of 0.599, above chance in 8 of 8 (Wilcoxon against 0.5, $p = 0.008$; Table 14).

Shifts average to zero. Under the feature-history identities of Section 4.2, the mean shift at every slot is zero in expectation, although individual shifts need not be. Averaged over all messages of a round, the fitted shift averages $+0.005$ in round 1, $+0.001$ in round 2 and $+0.012$ at the vote, against a typical per-message shift of 0.206 in round 1, and is not distinguishable from zero in round 1 ($p = 0.81$), round 2 ($p = 0.38$), or at the vote ($p = 0.64$). The fitted mean moves a lot on individual trajectories and not at all on average, as a posterior mean should.

Table 14: **Round-1 messages and the martingale check.** Left: mean shift in $\hat{\mu}$ at round-1 messages graded *wrong* and *right*, and AUROC↑ of the negative shift for separating them. Right: mean shift over all messages of each round, which the feature-history identities of Section 4.2 put at zero. Mean±SD over 6 seeds; message grades are external labels the estimator is never trained on.

		Round-1 messages			Mean shift over all messages		
Dataset	k	wrong	right	AUROC	round 1	round 2	vote
MATH500	2	-0.128	$+0.142$	0.647 ± 0.024	$+0.031$	$+0.002$	-0.012
MATH500	3	-0.104	$+0.054$	0.625 ± 0.015	-0.011	$+0.007$	-0.027
MATH500	5	-0.098	$+0.065$	0.610 ± 0.021	-0.000	$+0.001$	-0.010
MATH500	7	-0.065	$+0.046$	0.588 ± 0.026	$+0.001$	$+0.003$	$+0.040$
MMLU	2	-0.032	$+0.043$	0.578 ± 0.043	$+0.032$	-0.001	-0.017
MMLU	3	-0.145	$+0.013$	0.627 ± 0.022	-0.012	$+0.001$	$+0.077$
MMLU	5	-0.029	$+0.015$	0.563 ± 0.028	$+0.008$	-0.001	$+0.005$
MMLU	7	-0.067	$+0.001$	0.558 ± 0.017	-0.010	-0.002	$+0.038$

D.2 Early Detection from a Partial Debate

As configured in our experiments, every baseline scores the complete execution: PE, SE, SAR, LARS, and Sup read the delivered responses, and SAUP and PropUQ aggregate over all agents of the finished debate. We therefore compare them only on the complete debate. CUE fits one estimator across all prefixes (Equation (13)), so $\hat{\mu}_\ell$ exists at every step without refitting. This lets us ask how much of the final accuracy is available after the first round.

Table 15 reports the test AUROC of $\hat{\mu}_\ell$ read at four points of the trajectory: the query alone (Π_0), after round 1, after round 2, and at the delivered answer (the terminal score reported in Table 1). The last columns give the best of the seven baselines on that configuration, computed from the complete debate on the same test executions, and the number of seeds on which CUE after round 1 beats it. We pick the best baseline by its own test AUROC, which favors the baselines.

Table 15: **Early detection.** AUROC↑ (mean±SD over 6 seeds) of the CUE mean read at four points of the trajectory, against the best baseline's *full-debate* score on the same test executions. **Bold**: CUE after round 1 exceeds the best baseline's full-debate score. Last column: seeds (of 6) on which it does.

		CUE $\hat{\mu}_\ell$ read at				Best baseline (full debate)		
Dataset	k	Π_0	after R1	after R2	terminal	method	AUROC	R1 wins
MATH500	2	0.744 ± 0.027	$\mathbf{0.864} \pm 0.029$	0.876 ± 0.025	0.889 ± 0.025	SE	0.792	6/6
MATH500	3	0.765 ± 0.029	$\mathbf{0.849} \pm 0.021$	0.854 ± 0.018	0.873 ± 0.027	SE	0.836	5/6
MATH500	5	0.770 ± 0.016	$\mathbf{0.874} \pm 0.011$	0.887 ± 0.009	0.914 ± 0.005	SE	0.832	5/6
MATH500	7	0.771 ± 0.021	$\mathbf{0.848} \pm 0.029$	0.880 ± 0.029	0.891 ± 0.035	SE	0.840	4/6
MMLU	2	0.610 ± 0.043	$\mathbf{0.726} \pm 0.045$	0.719 ± 0.044	0.780 ± 0.033	SE	0.660	5/6
MMLU	3	0.694 ± 0.021	$\mathbf{0.813} \pm 0.039$	0.804 ± 0.018	0.845 ± 0.016	PropUQ	0.749	6/6
MMLU	5	0.555 ± 0.033	$\mathbf{0.751} \pm 0.059$	0.760 ± 0.033	0.784 ± 0.032	Sup	0.729	4/6
MMLU	7	0.600 ± 0.044	0.741 ± 0.066	0.765 ± 0.059	0.780 ± 0.084	Sup	0.775	2/6

In 7 of 8 configurations, the score after round 1 already exceeds what the best baseline achieves from the complete debate, and it does so on 37 of 48 seed-level comparisons (mean difference $+0.032$). With only 8 configurations the difference does not reach significance ($p = 0.055$). The largest margin is on MATH500 k=2, where CUE after round 1 scores 0.864 against 0.792 for SE. The one exception is MMLU k=7 (0.741 against 0.775 for Sup, -0.034), within one seed standard deviation.

CUE is supervised on terminal outcomes and most baselines are not, so part of its advantage is a learned prior over query difficulty: the query alone reaches 0.744 on MATH500 k=2, against 0.792 for SE. The advantage after round 1 goes beyond that prior, since reading the first round adds 0.120 AUROC on average over the query-only readout, in every configuration ($p = 0.008$).

The second round adds little. The score moves by only $+0.010$ between round 1 and round 2 ($p = 0.078$), and falls slightly on MMLU with $k \in \{2, 3\}$, consistent with the attribution study, in which most round-2 messages repeat round-1 answers and resolve nothing. Appendix D.4 discusses the rise from round 2 to the delivered answer ($+0.026$, $p = 0.008$).

D.3 Realized Amplification

Theorem 3.1 separates expected resolution, which is nonnegative, from realized change, which need not be: communication resolves uncertainty on average, yet the conditional variance can rise on individual trajectories Equation (9). We ask whether the realized increases that the estimator records, $\hat{\mathcal{A}} = \sum_\ell (\hat{V}_\ell - \hat{V}_{\ell-1})_+$, with $\hat{V}_\ell = 1 - \hat{\mu}_\ell^2$, occur where the theory says they should, on trajectories that change course. We group executions by whether the round-1 majority was right and whether the delivered answer is right: *stable right*, *recovered* (wrong→right), *corrupted* (right→wrong), and *stable wrong*. Executions with a tied round-1 vote are left out.

Corrupted trajectories accumulate more amplification than stable-right ones in 8 of 8 configurations ($p = 0.008$), and $\hat{V}$ rises during round 2 on a larger fraction of them in 8 of 8 ($p = 0.008$; Table 16). Recovered trajectories also carry more amplification than stable-right ones, in 8 of 8. This agrees with the theory: amplification marks a belief that moved toward uncertainty, whichever way the outcome then went, and says nothing about the direction of the outcome.

Table 16: **Realized amplification by trajectory group.** Mean $\hat{\mathcal{A}} = \sum_\ell (\hat{V}_\ell - \hat{V}_{\ell-1})_+$ per group, pooled over the six test splits because the changed groups are small (n: corrupted executions in total); fraction of corrupted and stable-right executions whose $\hat{V}$ rises during round 2; failure-detection AUROC (mean±SD) of the terminal score alone and combined with $\hat{\mathcal{A}}$ (5-fold cross-validated logistic combination on the test split).

		Mean $\hat{\mathcal{A}}$					$\hat{V}$ rises in R2		Failure AUROC	
Dataset	k	st. right	recovered	corrupted	st. wrong	n	corrupted	st. right	terminal	$+\hat{\mathcal{A}}$
MATH500	2	0.313	0.590	0.793	0.602	12	0.50	0.35	0.889 ± 0.025	0.881 ± 0.023
MATH500	3	0.418	0.601	0.630	0.665	36	0.62	0.56	0.873 ± 0.027	0.836 ± 0.095
MATH500	5	0.891	1.384	1.328	1.463	36	0.61	0.44	0.914 ± 0.005	0.912 ± 0.004
MATH500	7	0.765	1.610	1.581	1.640	30	0.62	0.33	0.891 ± 0.035	0.883 ± 0.035
MMLU	2	0.261	0.396	0.462	0.362	22	0.73	0.56	0.780 ± 0.033	0.739 ± 0.066
MMLU	3	0.575	0.800	0.969	0.765	57	0.65	0.32	0.845 ± 0.016	0.832 ± 0.016
MMLU	5	0.524	0.775	0.726	0.677	18	0.85	0.60	0.784 ± 0.032	0.720 ± 0.125
MMLU	7	0.798	1.130	1.129	1.135	25	0.75	0.60	0.780 ± 0.084	0.728 ± 0.141

Amplification therefore describes the trajectory; it is not an extra predictor of failure. On its own it detects failure with an AUROC of 0.627–0.763, but adding it to the terminal score raises the AUROC in 0 of 8 configurations. The corrupted group is also small (12–57 executions over six test splits), so we pool the group means in Table 16 instead of averaging them per seed.

D.4 Diagnostics of the Representation

The identities of Section 3 hold for the feature filtration, and the input of CUE at message ℓ is built from the prefix Π_ℓ, which includes every history message. Three observations show where encoding the prefix, which is truncated at the token budget and pooled over tokens, loses information that the history contains. First, for odd k the vote is a deterministic function of the round-2 answers, so the full-history theory gives it zero resolution. The fitted models give it 0.046–0.157 of $\hat{J}$ (Table 13), 69–234% of its uniform share, and for odd k the AUROC rises by $+0.011$ to $+0.041$ from round 2 to the vote (Table 15). The features at the vote state the answer explicitly, while the features after round 2 do not fully encode it. Second, the expected optimal Brier risk cannot increase as messages arrive. The fitted Brier risk falls from the query to round 1 in 7 of 8 configurations but rises in 9 of 16 later steps, each time by less than one seed standard deviation. Third, the vote adds information the earlier prefixes do not carry: on MMLU the fitted mean at the vote separates corrupted executions (mean $\hat{\mu}$ 0.030–0.478) from recovered ones (0.192–0.626) in 4 of 4 configurations, so a round-1 majority that later fails is not read as confident once the delivered answer is in the prefix.

D.5 Causal Intervention and Scope

The attribution is associational. On MATH500 with $k{=}2$, for each of 250 test executions we resampled the message with the largest attributed shift and, as a paired control, a randomly chosen other message, regenerating only the messages that depend on it. The graded outcome changed in 7.2% of executions for the attributed message and in 5.2% for the control (McNemar exact test on 31 discordant pairs, $p{=}0.47$), and the attributed magnitude of a resampled message is only weakly related to whether the outcome changed ($r{=}0.086$ over 500 resamplings, $p{=}0.055$). This is consistent with Section 3, which defines the shares as information revealed to an observer, so we report the attribution as a decomposition of the estimator's belief and make no causal claim about the debate. All case studies use one backbone (Llama-3.1-70B-Instruct) and one orchestration (two-round debate), and we do not claim that they transfer to the code and swarm configurations of Table 1.