EXPLAINING AND CALIBRATING DYNAMICALLY ORCHESTRATED LLM MAS THROUGH STEPWISE UNCERTAINTY DECOMPOSITION

Anonymous authors
Paper under double-blind review

ABSTRACT

Motivated by the need for trustworthiness and interpretability, reliable uncertainty decomposition and calibration techniques provide transparency into the black-box behavior of Large Language Models (LLMs). However, extending this explainability to LLM-based Multi-Agent Systems (MAS) remains challenging, as stochastic interactions among agents and system-level control introduce complex, nested uncertainty. To address this challenge, we propose Explanation and Calibration through Orchestration for Multi-Agent Systems (EcoMAS), a framework that reveals stepwise information contributions in dynamically orchestrated MAS through uncertainty decomposition and optimizes orchestration policies to improve calibration. We model system output uncertainty using second-order Tsallis entropy and derive an exact decomposition into stepwise information contributions from orchestration and agent execution. We develop a paired sampling algorithm that efficiently produces unbiased estimates of these contributions with theoretical finite-sample concentration bounds. We further derive an unbiased calibration gradient estimator from the semantic increments underlying information attribution, reusing the explanation samples for orchestration policy optimization. We evaluate EcoMAS on MMLU-Pro, MATH-500, and ChaosNLI to assess estimation accuracy, explanation faithfulness, and orchestration calibration. The results show that paired sampling accurately estimates information contributions with small sampling budgets. Case studies and intervention evaluations qualitatively illustrate how the attributions explain answer revision and reinforcement, while quantitative faithfulness evaluations demonstrate that EcoMAS reliably identifies steps that strongly influence system output uncertainty. Finally, comparisons of orchestration optimization methods show that EcoMAS reduces calibration error while improving task accuracy, highlighting the association between semantic calibration and MAS output quality.

1 INTRODUCTION

As LLM-based multi-agent systems (MAS) increasingly support complex reasoning and decision-making (Du et al., 2024; Chen et al., 2024; Wang et al., 2025b), understanding their collective behavior is essential for trustworthy deployment (Cemri et al., 2025; Zhang et al., 2025; Huang et al., 2025). Research on individual LLMs offers two complementary perspectives: uncertainty decomposition distinguishes sources of output variability (Hou et al., 2024; Ling et al., 2024; Ahdritz et al., 2024), while calibration assesses the agreement between predictive probabilities and observed outcomes or reference distributions (Detommaso et al., 2024; Band et al., 2024; Xiao et al., 2025). Semantic uncertainty further treats differently worded but equivalent outputs as the same answer, separating substantive disagreement from linguistic variation (Kuhn et al., 2023; Farquhar et al., 2024; Nikitin et al., 2024). Applying these perspectives to MAS could reveal how intermediate decisions contribute to final uncertainty and whether the resulting answer distribution is reliable.

However, stochastic interactions among agents and system-level control make this extension difficult (Chen et al., 2026; Kostka & Chudziak, 2026). Intermediate outputs change the context governing subsequent execution, so local response variability does not determine a step's contribution to final output uncertainty (Triantafyllou et al., 2025; Zhao et al., 2025; Zhang et al., 2025).

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.

Quantifying that contribution requires integrating over possible continuations, whose number grows **exponentially** with the remaining execution length under direct enumeration (Fang et al., 2026; Yao et al., 2023). Calibration poses a related difficulty: optimizing the frequency of correct answers does not determine how probability is allocated across semantic alternatives, particularly when reference answers are ambiguous or disputed (Min et al., 2020; Nie et al., 2020; Lee et al., 2023). Existing work studies uncertainty propagation across sequential decisions or aggregate variability across trajectories (Duan et al., 2025; Zhao et al., 2025; Chen et al., 2026), but exact attribution of final semantic uncertainty with tractable estimation and a principled connection to calibration optimization remains insufficiently established (Oh et al., 2026).

Our key observation is that agent selection and agent execution successively update the conditional semantic distribution of the final output. The expected squared magnitudes of these updates quantify their information contributions, while their alignment with the discrepancy from the true-answer distribution determines the contribution of orchestration decisions to the calibration gradient. We formalize this connection and propose *Explanation and Calibration through Orchestration for Multi-Agent Systems* (EcoMAS). Representing system output uncertainty by second-order Tsallis entropy (Tsallis, 1988), we show that it equals semantic feature variance under a same-cluster kernel and decomposes exactly into nonnegative stepwise contributions from orchestration and agent execution. We estimate these contributions through paired sampling, avoiding exhaustive continuation enumeration while retaining finite-sample unbiasedness and concentration guarantees. Finally, we reuse the explanation samples to estimate the calibration gradient for orchestration optimization.

We evaluate EcoMAS on MMLU-Pro (Wang et al., 2024), MATH-500 (Lightman et al., 2024), and ChaosNLI (Nie et al., 2020) across three LLM backbones. Comparisons with exact contributions on finite execution trees show accurate estimation with small sampling budgets. Case studies illustrate how EcoMAS explains answer revision and reinforcement, while quantitative evaluations demonstrate the faithfulness of its attributions. In orchestration optimization, EcoMAS achieves the lowest calibration error across all dataset and backbone combinations and the highest accuracy in eight of the nine settings. Our contributions are threefold: (1) We are the first to establish an exact, nonnegative decomposition of final semantic uncertainty into stepwise information contributions from orchestration and agent execution in LLM MAS. (2) We are the first to develop an efficient paired sampling estimator for these contributions and establish its finite-sample unbiasedness and concentration guarantees. (3) We are the first to derive an unbiased calibration gradient estimator from the semantic increments used for attribution, allowing the same samples to support both explanation and orchestration optimization.

2 Background

2.1 LLM Uncertainty Decomposition

For a fixed input, an LLM may generate different outputs across repeated executions. Semantic uncertainty groups differently worded outputs with equivalent meanings into the same cluster (Kuhn et al., 2023) and measures the dispersion of the resulting answer distribution. It has been used to assess the reliability of generated answers (Farquhar et al., 2024). Uncertainty decomposition asks how much uncertainty about the final answer remains after observing a random factor, such as a model configuration or an intermediate response (Depeweg et al., 2018; Duan et al., 2025). The difference between the original uncertainty and the expected remaining uncertainty measures the **information provided** by that factor. This connects uncertainty analysis to attribution by quantifying what an observation reveals about the final answer (Covert et al., 2020).

2.2 Uncertainty Calibration

The correctness of one generated answer provides limited evidence about how a model distributes probability across possible answers (Jiang et al., 2021; Nie et al., 2020). Calibration instead assesses the agreement between the model's output distribution and the true-answer distribution over the same semantic clusters (Vaicenavicius et al., 2019; Nie et al., 2020). Uncertainty decomposition identifies factors associated with variation in the final output (Depeweg et al., 2018; Covert et al., 2020). Comparing the conditional answer distributions induced by controllable factors with the true-answer distribution can therefore inform how those factors are adjusted to improve distributional agreement.

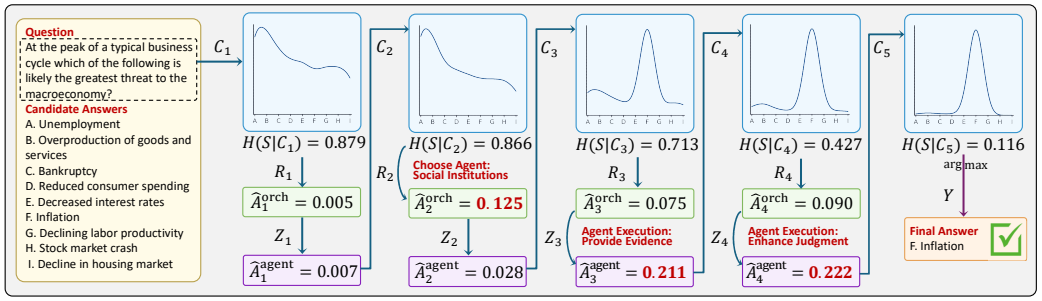


Figure 1: MMLU-Pro attribution: selecting relevant expertise and reinforcing the correct answer.

3 EcoMAS

EcoMAS explains uncertainty in dynamically orchestrated MAS (Dang et al., 2025; Wang et al., 2025c) and reuses the resulting information to calibrate orchestration. We first define stepwise contributions to system output uncertainty, then estimate them through paired sampling and use the same samples for orchestration optimization. Formal proofs and supporting analyses are organized in Appendices B and D. Our analysis builds on four general assumptions:

Assumption 1 (Sufficiency): Orchestration and agent execution depend on the current context only. *Assumption 2 (Stochasticity):* Randomness arises only from orchestration and agent execution; environmental updates are deterministic. *Assumption 3 (Consistency):* A fixed semantic clustering function consistently determines equivalence between system outputs. *Assumption 4 (Optimizability):* The parameterized orchestration policy is differentiable with respect to its parameters.

3.1 Explaining MAS through Uncertainty Decomposition

For a fixed input x, consider T execution steps with initial context $C_1 = x$. At step t, the policy π_θ selects one of N available agents $R_t \sim \pi_\theta(\cdot \mid C_t)$, which generates $Z_t \sim p_{R_t}(\cdot \mid C_t)$. The environment then updates the context as $C_{t+1} = F(C_t, R_t, Z_t)$. The policy and agent distributions induce a trajectory distribution $\mathbb{P}_\theta$; for a realized trajectory $(r_{1:T}, z_{1:T})$ and its corresponding contexts c_t,

$$\mathbb{P}_\theta(r_{1:T}, z_{1:T} \mid C_1 = x) = \prod_{t=1}^{T} \pi_\theta(r_t \mid c_t) p_{r_t}(z_t \mid c_t).$$

The final system output Y is determined once (C_T, R_T, Z_T) is known. Earlier selections and responses change the contexts governing later steps, so local response variability alone cannot quantify their information about the final output (Duan et al., 2025).

Under Assumption 3, γ assigns an output Y to its semantic cluster $S = \gamma(Y)$ in $\mathcal{S}$. We write $q_\theta(s) = \mathbb{P}_\theta(S = s)$, and we measure MAS output uncertainty by the second-order Tsallis entropy (Tsallis, 1988), which quantifies probability dispersion across semantic clusters:

$$H(S) = 1 - \sum_{s \in \mathcal{S}} q_\theta(s)^2.$$

A larger $H(S)$ indicates greater dispersion across semantic clusters. Its quadratic structure provides the mathematical basis for the theoretical analysis throughout this work, and it satisfies the basic properties of an uncertainty measure described in Appendix D.1.

Define the same-cluster kernel $k(y, y') = \mathbf{1}\{\gamma(y) = \gamma(y')\}$ and its one-hot feature map $\phi(y) = (\mathbf{1}\{\gamma(y) = s\})_{s \in \mathcal{S}}$. Thus $k(y, y') = \langle \phi(y), \phi(y') \rangle$; throughout, $\|\cdot\|$ is the Euclidean norm in this feature space. The kernel is positive semidefinite, as established in Appendix D.2. Because the mean of $\phi(Y)$ is the vector of semantic-cluster probabilities, the quadratic form of $H(S)$ admits the following variance and pairwise-disagreement interpretations.

Theorem 1. *Let Y and Y' be independent outputs generated from $C_1 = x$, and let $\mathbb{E}_\theta$ denote expectation under $\mathbb{P}_\theta$. Then*

$$H(S) = \mathbb{E}_\theta \left[\|\phi(Y) - \mathbb{E}_\theta[\phi(Y)]\|^2 \right] = \mathbb{E}_\theta[1 - k(Y, Y')] = \mathbb{P}_\theta\big(\gamma(Y) \neq \gamma(Y')\big). \tag{1}$$

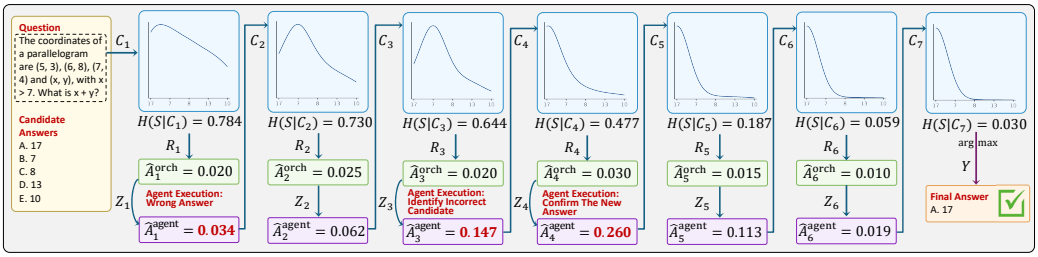


Figure 2: MATH-500 attribution: correcting an answer and consolidating the revised solution.

The proof is provided in Appendix B.1. Thus, system output uncertainty can be read as the probability that two independent executions produce semantically different answers. Its variance representation enables the stepwise decomposition below.

Observing the context, the selected agent, and its response successively refines the conditional semantic distribution of Y. We represent these three stages by

$$\mu_t(c) = \mathbb{E}_\theta[\phi(Y) \mid C_t = c], \quad \mu_t(c, r) = \mathbb{E}_\theta[\phi(Y) \mid C_t = c, R_t = r],$$
$$\mu_t(c, r, z) = \mathbb{E}_\theta[\phi(Y) \mid C_t = c, R_t = r, Z_t = z].$$

Each expectation integrates over the execution variables that remain unobserved, including subsequent steps. For brevity, we write $H(S \mid V) := \mathbb{E}_\theta[1 - \sum_{s \in \mathcal{S}} \mathbb{P}_\theta(S = s \mid V)^2]$ for any conditioning variable or tuple V, where the outer expectation averages over V.

The **orchestration contribution** is the expected reduction in final semantic uncertainty upon observing the selected agent:

$$A_t^{\text{orch}} := H(S \mid C_t) - H(S \mid C_t, R_t) = \mathbb{E}_\theta \left[\|\mu_t(C_t, R_t) - \mu_t(C_t)\|^2 \right]. \tag{2}$$

Observing the agent's response yields the additional **agent execution contribution**:

$$A_t^{\text{agent}} := H(S \mid C_t, R_t) - H(S \mid C_t, R_t, Z_t) = \mathbb{E}_\theta \left[\|\mu_t(C_t, R_t, Z_t) - \mu_t(C_t, R_t)\|^2 \right]. \tag{3}$$

These attribution values quantify how agent selection and outputs contribute to resolving uncertainty about the system's final answer, capturing the information provided at each stage of execution.

Theorem 2. *Under Assumptions 1–3, system output uncertainty decomposes exactly as*

$$H(S) = \sum_{t=1}^{T} \left(A_t^{\text{orch}} + A_t^{\text{agent}} \right).$$

The proof is provided in Appendix B.2. Every contribution is nonnegative, and it is zero precisely when observing the corresponding selection or response leaves the conditional distribution of the final semantic output unchanged almost surely. The decomposition therefore separates information provided by *which agent is selected* from information provided by *what that agent produces*; neither contribution by itself implies that the final answer is correct.

3.2 Estimating Information Contributions

Computing the conditional means in Equations 2 and 3 requires averaging over future orchestration decisions and agent outputs. Even with only two possible selection–output combinations per step, direct enumeration at step t requires 2^{T-t+1} continuations. EcoMAS instead uses paired sampling: it draws B independent main trajectories under $\mathbb{P}_\theta(\cdot \mid C_1 = x)$, with outputs $Y^{(b)}$, and constructs sparse continuation branches at each step. For $b = 1, \ldots, B$, write $C_t^{(b)}, R_t^{(b)}, Z_t^{(b)}$ for the main trajectory variables at step t.

For main trajectory b at step t, the branches produce

$$\widetilde{Y}_{t,C}^{(b)} \sim \mathbb{P}_\theta(Y \mid C_t = C_t^{(b)}),$$
$$\widetilde{Y}_{t,R}^{(b)} \sim \mathbb{P}_\theta(Y \mid C_t = C_t^{(b)}, R_t = R_t^{(b)}),$$
$$\widetilde{Y}_{t,Z}^{(b)} \sim \mathbb{P}_\theta(Y \mid C_t = C_t^{(b)}, R_t = R_t^{(b)}, Z_t = Z_t^{(b)}).$$

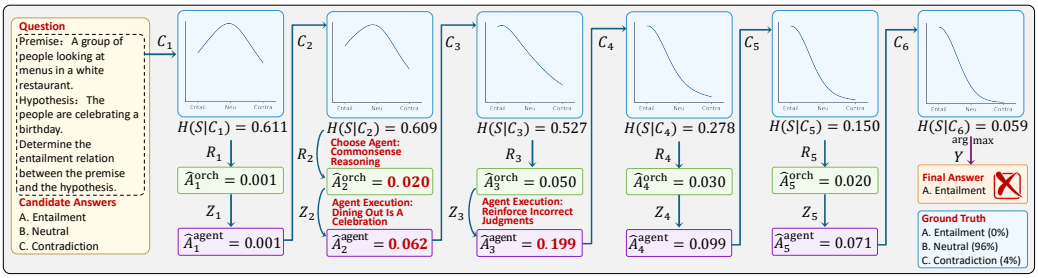


Figure 3: ChaosNLI attribution: an unsupported inference reinforced toward an incorrect answer.

They respectively retain the current context, the selected agent, and the complete current step. All unretained variables are resampled independently conditional on the retained variables; at $t = T$, $\widetilde{Y}_{T,Z}^{(b)} = Y^{(b)}$. Comparing each branch with the main output gives the single-trajectory estimates

$$\widehat{a}_{t,b}^{\text{orch}} = k(Y^{(b)}, \widetilde{Y}_{t,R}^{(b)}) - k(Y^{(b)}, \widetilde{Y}_{t,C}^{(b)}), \qquad \widehat{a}_{t,b}^{\text{agent}} = k(Y^{(b)}, \widetilde{Y}_{t,Z}^{(b)}) - k(Y^{(b)}, \widetilde{Y}_{t,R}^{(b)}). \tag{4}$$

For $\ell \in \{\text{orch}, \text{agent}\}$, we average over main trajectories:

$$\widehat{A}_t^{\ell} = \frac{1}{B} \sum_{b=1}^{B} \widehat{a}_{t,b}^{\ell}. \tag{5}$$

The full procedure is given in Algorithm 1 in Appendix D.3. The main outputs also yield a pairwise system uncertainty estimate $\widehat{H}(S)$, defined in Appendix D.4 and proved unbiased in Appendix B.3. Main trajectories cost $O(BT)$, while the three branches have total suffix length $O(B\sum_{t=1}^{T}(T - t + 1))$. Thus the overall complexity is $O(BT^2)$, replacing exponential enumeration with quadratic dependence on execution length.

Proposition 1. *Under Assumptions 1–3 and the paired sampling construction, $\forall t$ and $B \geq 1$,*

$$\mathbb{E}[\widehat{A}_t^{\text{orch}}] = A_t^{\text{orch}}, \qquad \mathbb{E}[\widehat{A}_t^{\text{agent}}] = A_t^{\text{agent}}.$$

The proof is provided in Appendix B.4. Each branch uses only one sampled continuation, yet averaging its same-cluster comparison over the sampling process recovers the corresponding conditional semantic change exactly. Unbiasedness therefore holds even for a single main trajectory; increasing B controls estimation error.

To quantify this error simultaneously across steps, the bounded single-trajectory estimates give the following distribution-free guarantee.

Theorem 3. *For any $\delta \in (0, 1)$, with probability at least $1 - \delta$, all $2T$ estimates satisfy*

$$\max_{\substack{1 \leq t \leq T \\ \ell \in \{\text{orch}, \text{agent}\}}} \left| \widehat{A}_t^{\ell} - A_t^{\ell} \right| < \sqrt{\frac{2\log(4T/\delta)}{B}}. \tag{6}$$

The proof and the corresponding tail probabilities are given in Appendices B.5 and D.4. The bound gives a sufficient budget $B \geq 2\varepsilon^{-2}\log(4T/\delta)$ for additive error below ε across all steps, and an $O(B^{-1/2})$ rate for fixed T and δ.

This worst-case bound does not use the discrete structure of the same-cluster kernel. A sharper characterization depends on whether the two kernel comparisons in Equation 4 disagree.

Theorem 4. *For $t \in \{1, \ldots, T\}$ and $\ell \in \{\text{orch}, \text{agent}\}$, define the semantic-boundary probability*

$$\beta_t^{\ell} := \mathbb{P}(\widehat{a}_{t,b}^{\ell} \neq 0). \tag{7}$$

Then $\text{Var}(\widehat{a}_{t,b}^{\ell}) = \beta_t^{\ell} - (A_t^{\ell})^2$, and for every $\varepsilon > 0$,

$$\mathbb{P}\left(\left| \widehat{A}_t^{\ell} - A_t^{\ell} \right| \geq \varepsilon \right) \leq 2\exp\left(-\frac{B\varepsilon^2}{2[\beta_t^{\ell} - (A_t^{\ell})^2] + 4\varepsilon/3} \right).$$

The proof is provided in Appendix B.6. Since each estimate is the difference of two binary same-cluster indicators, its variance depends on their probability of disagreement. The number of possible future trajectories does not enter this variance explicitly; execution length and agent choice affect it through β_t^ℓ and A_t^ℓ. The combined concentration bound and a separate bound for $\widehat{H}(S)$ are given in Appendices D.4 and D.5, respectively.

3.3 Calibrating Orchestration through Attribution Estimation

The preceding decomposition identifies how orchestration and agent execution change the conditional distribution of the final semantic output. We now use these changes to optimize agreement between the system output distribution and the true-answer distribution. For the fixed input $C_1 = x$, let $\mathbb{P}^\star$ denote the true-answer distribution and $Y^\star \sim \mathbb{P}^\star$. The semantic-cluster probability vectors are $q_\theta = \mathbb{E}_\theta[\phi(Y)]$ and $q^\star = \mathbb{E}_{Y^\star \sim \mathbb{P}^\star}[\phi(Y^\star)]$. We define the semantic calibration error as

$$\mathcal{L}_{\mathrm{cal}}(\theta) := \|q_\theta - q^\star\|^2 = \mathrm{MMD}_k^2\left(\mathbb{P}_\theta(Y \mid C_1 = x), \mathbb{P}^\star\right),$$

where MMD_k is the maximum mean discrepancy under the same-cluster kernel (Gretton et al., 2006). This objective measures how probability is allocated across semantic answers. It also equals the part of the expected Brier score (Brier, 1950) that depends on the system distribution, as stated in Appendix D.6 and proved in Appendix B.7. Under Assumption 4, we optimize the orchestration policy π_θ. Equation 2 shows that the expected squared norm of $\mu_t(C_t, R_t) - \mu_t(C_t)$ is the orchestration information contribution. The same conditional semantic change determines the calibration gradient.

Theorem 5. *Under Assumptions 1–4, suppose the trajectory distribution has locally parameter-independent support and differentiation can be interchanged with expectation. Then*

$$\nabla_\theta \mathcal{L}_{\mathrm{cal}}(\theta) = 2\sum_{t=1}^{T} \mathbb{E}_\theta\left[\langle q_\theta - q^\star,\, \mu_t(C_t, R_t) - \mu_t(C_t)\rangle\, \nabla_\theta \log \pi_\theta(R_t \mid C_t)\right]. \tag{8}$$

The proof is provided in Appendix B.8. Attribution measures the magnitude of a conditional semantic change, whereas its inner product with $q_\theta - q^\star$ determines the signed weight of that change in the calibration gradient. An informative orchestration decision therefore need not improve calibration; its direction relative to the true-answer distribution also matters.

Estimating q_θ and the conditional semantic change from the same trajectory would introduce a finite-sample self-product bias. For $B \geq 2$, we therefore exclude trajectory b when estimating q_θ, using $\widehat{q}_{-b} = (B-1)^{-1}\sum_{j \neq b} \phi(Y^{(j)})$. The branches $\widetilde{Y}_{t,R}^{(b)}$ and $\widetilde{Y}_{t,C}^{(b)}$ from Algorithm 1 estimate the two conditional means in Equation 8. Together they yield

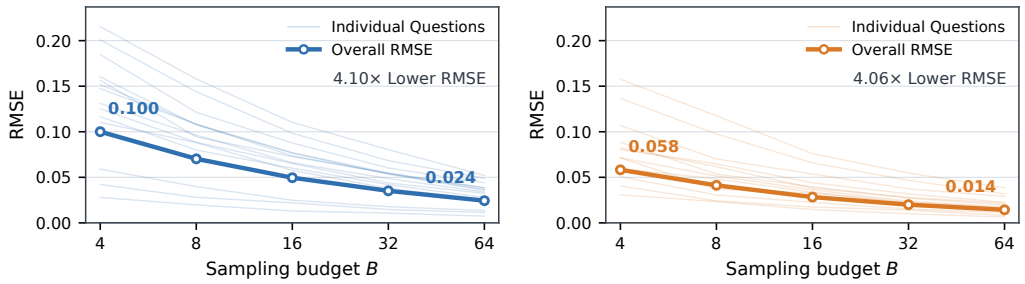
$$\widehat{\nabla_\theta \mathcal{L}_{\mathrm{cal}}} := \frac{2}{B}\sum_{b=1}^{B}\sum_{t=1}^{T} \left\langle \widehat{q}_{-b} - q^\star,\, \phi(\widetilde{Y}_{t,R}^{(b)}) - \phi(\widetilde{Y}_{t,C}^{(b)}) \right\rangle \nabla_\theta \log \pi_\theta(R_t^{(b)} \mid C_t^{(b)}). \tag{9}$$

The estimation procedure is given in Algorithm 2 (Appendix D.7).

Theorem 6. *Under Assumptions 1–4 and the conditions of Theorem 5, if $q^\star$ is known and $B \geq 2$, then $\mathbb{E}[\widehat{\nabla_\theta \mathcal{L}_{\mathrm{cal}}}] = \nabla_\theta \mathcal{L}_{\mathrm{cal}}(\theta)$.*

The proof is provided in Appendix B.9. Leaving out trajectory b makes $\widehat{q}_{-b}$ independent of its gradient term, removing the self-product bias. Because the estimator reuses the main outputs and two branches already generated for attribution, calibration optimization requires no additional MAS continuation calls within the same sampling batch. Appendix D.8 characterizes when varying the orchestration policy can attain zero calibration error.

EcoMAS connects stepwise information attribution and semantic calibration through the conditional semantic changes induced during execution. The exact decomposition expresses system output uncertainty as the sum of orchestration and agent-execution information contributions, while paired sampling estimates these contributions with explicit error guarantees. Reusing the sampled semantic increments yields an unbiased calibration gradient, allowing the same samples to support both explanation and orchestration optimization.



(a) Sampling estimation error of A_t^{orch}

(b) Sampling estimation error of A_t^{agent}

Figure 4: Sampling estimation error of stepwise information contributions: (a) orchestration and (b) agent execution. Light curves represent individual questions; dark curves represent overall RMSE.

4 Experiments

In this section, we conduct extensive experiments to validate and evaluate EcoMAS by addressing four research questions (RQs): (1) Can paired sampling accurately estimate stepwise information contributions from orchestration and agent execution? (2) What insights can EcoMAS's stepwise information attribution provide into dynamically orchestrated MAS? (3) How faithfully do EcoMAS explanations reflect the reduction in system output uncertainty throughout execution? (4) Can EcoMAS's orchestration optimization improve task accuracy by reducing system calibration error? We provide the code of EcoMAS in Anonymous Github. Appendix D.9 distinguishes benchmark comparisons from finite-support validation and implementation diagnostics.

4.1 Experiment Settings

Datasets. We evaluate MMLU-Pro (Wang et al., 2024), MATH-500 (Lightman et al., 2024), and ChaosNLI (Nie et al., 2020), covering multiple-choice, mathematical, and human-disagreement answer spaces. Semantic clusters distinguish answer options, canonical mathematical answers, and the three NLI labels. Appendix C specifies parsing and semantic labeling; run manifests record splits and seeds.

Baselines. Our baselines fall into two categories: (1) explanation baselines based on semantic uncertainty, including Semantic Entropy (Kuhn et al., 2023), sentSAR (Duan et al., 2024), and UProp (Duan et al., 2025); and (2) learned orchestration baselines, including Puppeteer (Dang et al., 2025), STRMAC (Wang et al., 2025a), MACA (Li et al., 2026), and BOT-Orch (Oliver et al., 2026), together with uniform random orchestration as a reference baseline. All baseline methods share the same experimental environment and common settings.

MAS and LLM backbones. Each benchmark uses five role-specialized agents (Appendix C, Table 3). At each step, a pretrained orchestrator selects an agent from the current context, allowing repeated selections; its response updates the context, and the final answer is parsed from the last output. We evaluate Qwen2.5-7B-Instruct (Qwen Team, 2025), Olmo-3-7B-Instruct (Team Olmo, 2025), and GPT-OSS-20B (OpenAI, 2025). Experiments use one NVIDIA A100 GPU and fixed random seeds. Task-specific parsers and semantic equivalence checks are described in Appendix C.

4.2 Approximation Error Analysis of Stepwise Contribution Estimation (RQ1)

We evaluate paired sampling on 28 diagnostic questions spanning all 14 disciplines of MMLU-Pro, using four execution steps per question. This is an executed exact finite-support validation, chosen to make the target quantity observable rather than to estimate the full MMLU-Pro distribution. For each question, we construct a complete finite-support execution tree and compute exact stepwise information contributions through dynamic programming. At each sampling budget, we run EcoMAS for 200 independent repetitions and compute the root mean squared error (RMSE) against the

Table 1: Explanation faithfulness across datasets and LLM backbones. H_{AUC} is the area under the normalized remaining uncertainty curve; $H@25\%$ and $H@75\%$ are the normalized remaining uncertainty after intervening on the top 25% and 75% of ranked decisions. Lower values are better.

Method	MMLU-Pro			MATH-500			ChaosNLI		
	H_{AUC}	$H@25\%$	$H@75\%$	H_{AUC}	$H@25\%$	$H@75\%$	H_{AUC}	$H@25\%$	$H@75\%$
Qwen2.5-7B-Instruct									
Random	0.7977	0.9542	0.8306	0.5458	0.7333	0.4833	0.5125	0.7457	0.4278
EcoMAS	**0.5635**	**0.6917**	**0.3625**	**0.3972**	**0.4833**	**0.2500**	**0.2250**	**0.3259**	**0.1028**
Semantic Entropy	0.6659	0.7847	0.5292	0.4250	0.5170	**0.2500**	0.4250	0.6250	0.3163
sentSAR	0.6581	0.7832	0.5091	0.4365	**0.4833**	0.2875	0.4875	0.6094	0.3347
UProp	0.7885	0.9261	0.6736	0.4694	0.5833	0.4021	0.2875	0.4065	0.1728
Olmo-3-7B-Instruct									
Random	0.7888	0.9607	0.8591	0.5601	0.7570	0.4653	0.5127	0.7492	0.4393
EcoMAS	**0.5443**	**0.7149**	**0.3844**	**0.3996**	**0.4935**	**0.2267**	**0.2405**	**0.3537**	0.1521
Semantic Entropy	0.6697	0.7750	0.5011	0.4102	0.5020	0.2409	0.4327	0.6347	0.3107
sentSAR	0.6316	0.7786	0.4853	0.4160	0.5093	0.2924	0.4983	0.5985	0.3414
UProp	0.7659	0.9208	0.6577	0.4575	0.5635	0.3839	0.2917	0.4146	**0.1490**
GPT-OSS-20B									
Random	0.7701	0.9366	0.8456	0.5309	0.7402	0.4439	0.4851	0.7373	0.4140
EcoMAS	**0.5297**	**0.6864**	**0.3646**	**0.3869**	**0.4699**	**0.2096**	**0.2143**	**0.3332**	**0.1167**
Semantic Entropy	0.6468	0.7587	0.4740	0.3920	0.4905	0.2161	0.4136	0.6058	0.2969
sentSAR	0.6059	0.7662	0.4637	0.3892	0.4944	0.2629	0.4806	0.5754	0.3306
UProp	0.7527	0.9013	0.6296	0.4418	0.5412	0.3727	0.2672	0.3972	0.1224

exact contributions across steps and repetitions, separately for orchestration and agent execution. We report both the RMSE for each question and the overall RMSE across all 28 questions.

Figure 4 shows modest overall errors even with a small sampling budget. Increasing the budget reduces overall RMSE by approximately fourfold for both contribution types, consistent with the theoretical $O(B^{-1/2})$ convergence rate. Errors also decline across individual questions. Thus the executed tree experiment directly validates the estimator against its exact target and supports its use at small B; it does not claim that 28 questions characterize every benchmark. Appendix D.10 further demonstrates the substantial speedup (over $30{,}000\times$) of EcoMAS over exact computation.

4.3 Interpreting Stepwise Information Contributions (RQ2)

We present case studies from MMLU-Pro, MATH-500, and ChaosNLI with stable contribution estimates and consistent answer distribution trends across repeated tests. The figures show averaged stepwise information contributions, remaining uncertainty, and evolving answer distributions, illustrating how EcoMAS attributes answer revision and reinforcement to agent selection and execution.

In MMLU-Pro (Figure 1), EcoMAS attributes the shift toward inflation to selecting the social institutions agent and its consolidation to subsequent economic reasoning. In MATH-500 (Figure 2), checking a geometric constraint corrects the answer, while subsequent verification contributes more information by consolidating it. In ChaosNLI (Figure 3), an unsupported birthday inference shifts the answer toward entailment and is subsequently reinforced, showing that positive information contributions can increase certainty in an incorrect conclusion. These cases show that information contributions explain answer revision and reinforcement through expected uncertainty reduction, without implying correctness. We further observe an interesting pattern: in both correct and erroneous convergence, answer revision contributes less information than subsequent reinforcement, highlighting the distinction between changing an answer and consolidating it.

4.4 Faithfulness of Information Attribution (RQ3)

We evaluate explanation faithfulness by ranking stepwise orchestration decisions and agent executions, then progressively making them deterministic: selecting the most probable agent or using greedy decoding, respectively. Rankings remain fixed, and system uncertainty is re-estimated after each cumulative intervention. We use 100 fixed questions per dataset, five evaluation repetitions,

Table 2: Task accuracy (Acc) and semantic calibration error (CErr) across orchestration methods, datasets, and LLM backbones. Higher Acc and lower CErr indicate better performance.

Method	Uniform		EcoMAS		Puppeteer		STRMAC		MACA		BOT-Orch	
	Acc(↑)	CErr(↓)	Acc(↑)	CErr(↓)	Acc(↑)	CErr(↓)	Acc(↑)	CErr(↓)	Acc(↑)	CErr(↓)	Acc(↑)	CErr(↓)
Qwen2.5-7B-Instruct												
MMLU-Pro	0.584	0.5596	**0.667**	**0.4264**	0.623	0.4979	0.615	0.5396	0.612	0.5086	0.608	0.5189
MATH-500	0.706	0.4171	**0.742**	**0.3679**	0.738	0.4021	0.724	0.4054	0.729	0.3946	0.703	0.3996
ChaosNLI	0.655	0.3595	**0.736**	**0.2295**	0.721	0.2935	0.658	0.3595	0.704	0.2817	0.717	0.3267
Olmo-3-7B-Instruct												
MMLU-Pro	0.598	0.5783	**0.647**	**0.4140**	0.636	0.5242	0.605	0.5347	0.620	0.5184	0.579	0.4974
MATH-500	0.684	0.4228	**0.764**	**0.3662**	0.722	0.3945	0.735	0.4196	0.692	0.3910	0.706	0.3828
ChaosNLI	0.661	0.3711	**0.746**	**0.2047**	0.720	0.3129	0.639	0.3492	0.727	0.3092	0.707	0.3236
GPT-OSS-20B												
MMLU-Pro	0.582	0.5601	0.626	**0.3893**	**0.628**	0.5106	0.582	0.5056	0.601	0.5019	0.566	0.4756
MATH-500	0.654	0.4085	**0.745**	**0.3376**	0.697	0.3751	0.719	0.4067	0.661	0.3675	0.691	0.3656
ChaosNLI	0.618	0.3484	**0.722**	**0.1886**	0.703	0.2855	0.647	0.3307	0.706	0.2975	0.684	0.2987

and sampling budgets of $B = 32$ for attribution and $B = 64$ for interventions (Appendix C). Metrics include H_{AUC}, the area under the remaining uncertainty curve normalized by initial uncertainty, and $H@25\%$ and $H@75\%$, the normalized remaining uncertainty after intervening on the top 25% and 75% of ranked components. **Lower** values indicate more faithful prioritization of decisions and executions contributing to system uncertainty. Questions are the independent evaluation units; stochastic executions within each question estimate its output distribution.

Table 1 shows that EcoMAS achieves the lowest H_{AUC} and $H@25\%$ across dataset–backbone combinations, indicating that its rankings prioritize orchestration decisions and agent executions whose determinization reduces system uncertainty most effectively. This advantage generally persists at $H@75\%$. Gains are larger on MMLU-Pro and ChaosNLI than on MATH-500, with H_{AUC} reductions of approximately 18%–22% over the strongest baseline on ChaosNLI. These results support the faithfulness of EcoMAS's stepwise information attribution across tasks and backbones.

4.5 Orchestration Calibration and Task Accuracy (RQ4)

We compare orchestration methods using the same agent pool, execution settings, and data splits for each backbone, training all learned policies on the same 300 examples per dataset. EcoMAS reuses attribution samples to estimate calibration gradients. Evaluation uses 100 fixed questions with 10 stochastic executions each. Accuracy (Acc) averages over executions; semantic calibration error (CErr) uses a same-cluster kernel U-statistic per question. Targets are the correct answers for MMLU-Pro and MATH-500 and human annotation distributions for ChaosNLI.

Table 2 shows that EcoMAS achieves the lowest calibration error in each tested setting and the highest accuracy in eight of nine settings. On MMLU-Pro with GPT-OSS-20B, its accuracy approaches Puppeteer's with substantially lower calibration error. On ChaosNLI, calibration error decreases by approximately 34% relative to the strongest baseline. Lower calibration error generally accompanies higher accuracy, supporting the effectiveness of reusing attribution samples for orchestration optimization toward both correct answers and human answer distributions within the evaluated settings.

5 Conclusion

EcoMAS explains and calibrates dynamically orchestrated LLM MAS through the conditional semantic changes induced by agent selection and execution. Second-order Tsallis entropy yields an exact decomposition of system output uncertainty; paired sampling estimates its stepwise contributions with finite-sample guarantees and supplies an unbiased orchestration calibration gradient. Exact-tree validation supports estimator accuracy, while intervention experiments and orchestration comparisons show faithful attribution, lower calibration error, and generally higher task accuracy across the tested settings. Together, these results connect understanding how collective answers emerge to improving their distributional reliability. Future work may extend this analysis to adaptive execution and joint optimization of agents and orchestration.

6 AI USE STATEMENT

LLMs were used to aid in the writing and polishing of the manuscript. Specifically, we used an LLM to assist in refining the language, improving readability, and ensuring clarity in various sections of the paper. The model helped with tasks such as sentence rephrasing, grammar checking, and enhancing the overall flow of the text.

The authors take full responsibility for the content of the manuscript, including any text generated or polished by the LLM. We have ensured that the LLM-generated text adheres to ethical guidelines and does not contribute to plagiarism or scientific misconduct.

7 ETHICS STATEMENT

This work does not involve the collection of new data from human participants or personally identifiable information. We use the corresponding benchmarks and model services subject to their applicable licenses and terms of use. As with other language-model-based agent systems, the behaviors studied in this work may inherit limitations or biases from the underlying models and environments.

8 REPRODUCIBILITY STATEMENT

Section 3 defines the MAS, estimators, and calibration objective; Section 4 specifies the benchmarks, baselines, backbones, and evaluation protocols. Appendix B proves the main results, Appendix C records agent roles, generation, parsing, and intervention settings, and Appendix D gives algorithms and additional analyses. Appendix D.9 identifies the scope of the finite-support validation, benchmark experiments, and diagnostic probe. Code and run configurations are available at the anonymous repository.

REFERENCES

Gustaf Ahdritz, Tian Qin, Nikhil Vyas, Boaz Barak, and Benjamin L. Edelman. Distinguishing the knowable from the unknowable with language models. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 503–549, 2024.

Neil Band, Xuechen Li, Tengyu Ma, and Tatsunori Hashimoto. Linguistic calibration of long-form generations. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 2732–2778, 2024.

Glenn W Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950.

Mert Cemri, Melissa Z Pan, Shuyi Yang, Lakshya A Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei A Zaharia, Joseph E Gonzalez, and Ion Stoica. Why do multi-agent llm systems fail? In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Justin Chen, Swarnadeep Saha, and Mohit Bansal. Reconcile: Round-table conference improves reasoning via consensus among diverse llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7066–7085, 2024.

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

Ian Covert, Scott M Lundberg, and Su-In Lee. Understanding global feature contributions with additive importance measures. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Yufan Dang, Chen Qian, Xueheng Luo, Jingru Fan, Zihao Xie, Ruijie Shi, Weize Chen, Cheng Yang, Xiaoyin Che, Ye Tian, Xuantang Xiong, Lei Han, Zhiyuan Liu, and Maosong Sun. Multi-agent collaboration via evolving orchestration. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Stefan Depeweg, José Miguel Hernández-Lobato, Finale Doshi-Velez, and Steffen Udluft. Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 1184–1193, 2018.

Gianluca Detommaso, Martin Andres Bertran, Riccardo Fogliato, and Aaron Roth. Multicalibration for confidence scoring in LLMs. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 10624–10641, 2024.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 11733–11763, 2024.

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 *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5050–5063, 2024.

Jinhao Duan, James Diffenderfer, Sandeep Madireddy, Tianlong Chen, Bhavya Kailkhura, and Kaidi Xu. UProp: Investigating the uncertainty propagation of LLMs in multi-step agentic decision-making, 2025.

Jinyuan Fang, Zaiqiao Meng, and Craig Macdonald. SPARKLE: A structured and plug-and-play agentic retrieval policy for adaptive RAG models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 38699–38719, 2026.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630:625–630, 2024.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alex J Smola. A kernel method for the two-sample-problem. In *Advances in Neural Information Processing Systems*, volume 19, pp. 513–520, 2006.

Bairu Hou, Yujian Liu, Kaizhi Qian, Jacob Andreas, Shiyu Chang, and Yang Zhang. Decomposing uncertainty for large language models through input clarification ensembling. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 19023–19042, 2024.

Jen-Tse Huang, Jiaxu Zhou, Tailin Jin, Xuhui Zhou, Zixi Chen, Wenxuan Wang, Youliang Yuan, Michael Lyu, and Maarten Sap. On the resilience of llm-based multi-agent collaboration with faulty agents. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 26202–26226, 2025.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977, 2021.

Adam Kostka and Jaroslaw A. Chudziak. Controlling uncertainty and hallucination risk in multi-agent fact verification. In *Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence*, pp. 3143–3161, 2026.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *International Conference on Learning Representations*, 2023.

Noah Lee, Na Min An, and James Thorne. Can large language models capture dissenting human voices? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4569–4585, 2023.

Haoran Li, Shulun Chen, Shaoyuan Sun, and Hanchen Wang. Multi-agent coordination adaptation via structure-guided orchestration, 2026.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024.

Chen Ling, Xujiang Zhao, Xuchao Zhang, Wei Cheng, Yanchi Liu, Yiyou Sun, Mika Oishi, Takao Osaki, Katsushi Matsuda, Jie Ji, Guangji Bai, Liang Zhao, and Haifeng Chen. Uncertainty quantification for in-context learning of large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3357–3370, 2024.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5783–5797, 2020.

Yixin Nie, Xiang Zhou, and Mohit Bansal. What can we learn from collective human opinions on natural language inference data? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9131–9143, 2020.

Alexander Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. Kernel language entropy: Fine-grained uncertainty quantification for LLMs from semantic similarities. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Changdae Oh, Seongheon Park, To Eun Kim, Jiatong Li, Wendi Li, Samuel Yeh, Sean Du, Hamed Hassani, Paul Bogdan, Dawn Song, and Sharon Li. Uncertainty quantification in LLM agents: Foundations, emerging challenges, and opportunities. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16219–16250, 2026.

Mary Chriselda Antony Oliver, Lan Jiang, Aaron Bundi Anampiu, Elaf Almahmoud, Francesco Quinzan, and Umang Bhatt. Learning to orchestrate agents under uncertainty, 2026.

OpenAI. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025.

Team Olmo. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

Stelios Triantafyllou, Aleksa Sukovic, Yasaman Zolfimoselo, and Goran Radanovic. Counterfactual effect decomposition in multi-agent sequential decision making. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 60072–60098, 2025.

Constantino Tsallis. Possible generalization of boltzmann-gibbs statistics. *Journal of Statistical Physics*, 52(1–2):479–487, 1988. doi: 10.1007/BF01016429.

Juozas Vaicenavicius, David Widmann, Carl Andersson, Fredrik Lindsten, Jacob Roll, and Thomas Schön. Evaluating model calibration in classification. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, pp. 3459–3467, 2019.

Jingbo Wang, Sendong Zhao, Haochun Wang, Yuzheng Fan, Lizhe Zhang, Yan Liu, and Ting Liu. Optimal-agent-selection: State-aware routing framework for efficient multi-agent collaboration, 2025a.

Shilong Wang, Guibin Zhang, Miao Yu, Guancheng Wan, Fanci Meng, Chongye Guo, Kun Wang, and Yang Wang. G-safeguard: A topology-guided security lens and treatment on llm-based multi-agent systems. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7261–7276, 2025b.

Song Wang, Zhen Tan, Zihan Chen, Shuang Zhou, Tianlong Chen, and Jundong Li. AnyMAC: Cascading flexible multi-agent collaboration via next-agent prediction. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 11555–11567, 2025c.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Jiancong Xiao, Bojian Hou, Zhanliang Wang, Ruochen Jin, Qi Long, Weijie J Su, and Li Shen. Restoring calibration for aligned large language models: A calibration-aware fine-tuning approach. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 68364–68390, 2025.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

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 *Proceedings of the 42nd International Conference on Machine Learning*, pp. 76583–76599, 2025.

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 *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6064–6073, 2025.

A Related Works

A.1 Uncertainty Quantification and Decomposition in LLMs

Understanding uncertainty in language generation requires distinguishing variation in meaning from variation in wording. Semantic Entropy addresses this issue by grouping mutually entailing generations and estimating entropy over their aggregated probabilities (Kuhn et al., 2023). The sentence-level variant of SAR, sentSAR, instead incorporates probability-weighted similarities between generated responses into uncertainty estimation (Duan et al., 2024). These approaches establish semantic consistency as a basis for assessing predictive uncertainty, but their response-level scores do not directly quantify how intermediate decisions contribute to uncertainty in a later system output.

Sequential agent execution introduces an additional distinction between uncertainty within the current decision and uncertainty inherited from earlier decisions. UProp formalizes this distinction through intrinsic and extrinsic uncertainty and estimates propagation using trajectory-dependent sampling and pointwise mutual information (Duan et al., 2025). This advances uncertainty analysis beyond isolated responses, while leaving the separate information contributions of agent selection and agent execution in dynamically orchestrated MAS unresolved. EcoMAS addresses this attribution problem by measuring their successive contributions to the final semantic distribution. The variance representation of second-order Tsallis entropy enables an exact, nonnegative decomposition, and paired conditional continuations provide unbiased contribution estimates with finite-sample concentration guarantees.

A.2 Learning and Optimizing MAS Orchestration

Learning orchestration policies enables agent selection to adapt to evolving execution contexts. Existing approaches obtain supervision either from task rewards or from successful execution paths. Puppeteer trains a centralized orchestrator through reinforcement learning, balancing answer quality against computational cost (Dang et al., 2025). STRMAC learns to match execution states with agent expertise through contrastive training, supported by an iterative data collection procedure (Wang et al., 2025a). MACA further introduces a task- and budget-conditioned structural prior to guide reinforcement learning of orchestration policies (Li et al., 2026). Together, these approaches improve the effectiveness and efficiency of coordination, while their training objectives do not directly specify agreement between the final semantic answer distribution and a reference distribution.

Distribution-aware orchestration extends this perspective beyond expected task reward. BOT-Orch formulates agent delegation as a bandit problem and regularizes rewards using optimal transport distances between agent output distributions and task-specific reference distributions (Oliver et al., 2026). This explicitly incorporates distributional alignment into agent selection. EcoMAS focuses on the final answer distribution induced by an entire collaborative execution and connects its calibration objective to stepwise information attribution. Specifically, the conditional semantic increments underlying orchestration contributions also determine the calibration gradient. Reusing attribution samples through leave-one-out estimation yields an unbiased gradient estimator without additional continuation calls within the same batch, connecting the explanation of system uncertainty to orchestration optimization.

B Proofs of Main-Text Results

This section proves the results stated in the main text, in their order of appearance.

B.1 Proof of Theorem 1

Proof. Since $\phi(Y)$ is a one-hot vector, $\|\phi(Y)\|^2 = 1$, and its expectation is the semantic cluster probability vector $(q_\theta(s))_{s\in\mathcal{S}}$. Therefore,

$$\mathbb{E}_\theta\left[\|\phi(Y) - \mathbb{E}_\theta[\phi(Y)]\|^2\right] = \mathbb{E}_\theta[\|\phi(Y)\|^2] - \|\mathbb{E}_\theta[\phi(Y)]\|^2 = 1 - \sum_{s\in\mathcal{S}} q_\theta(s)^2 = H(S).$$

The independence of Y and Y' further gives

$$\mathbb{E}_\theta[k(Y, Y')] = \sum_{s \in \mathcal{S}} \mathbb{P}_\theta(S = s)^2 = \sum_{s \in \mathcal{S}} q_\theta(s)^2.$$

Because $1 - k(Y, Y')$ indicates that the two outputs belong to different semantic clusters, the remaining equalities follow. □

B.2 Proof of Theorem 2

Proof. Adding the entropy reductions in Equations 2 and 3 gives

$$A_t^{\text{orch}} + A_t^{\text{agent}} = H(S \mid C_t) - H(S \mid C_t, R_t, Z_t).$$

By context sufficiency and the deterministic update $C_{t+1} = F(C_t, R_t, Z_t)$, the conditional distributions satisfy

$$\mathbb{P}_\theta(S \mid C_t, R_t, Z_t) = \mathbb{P}_\theta(S \mid C_{t+1}), \qquad t < T.$$

Consequently, $H(S \mid C_t, R_t, Z_t) = H(S \mid C_{t+1})$, so summing over all steps gives the telescoping identity

$$\sum_{t=1}^{T} \left(A_t^{\text{orch}} + A_t^{\text{agent}}\right) = H(S \mid C_1) - H(S \mid C_T, R_T, Z_T).$$

Since $C_1 = x$ is fixed, $H(S \mid C_1) = H(S)$. Moreover, S is fully determined by (C_T, R_T, Z_T), so $H(S \mid C_T, R_T, Z_T) = 0$. Substituting these boundary conditions proves the theorem. □

B.3 Unbiasedness of the System Uncertainty Estimate

For independent main outputs $Y^{(i)}$ and $Y^{(j)}$, Equation 1 gives $\mathbb{E}[1 - k(Y^{(i)}, Y^{(j)})] = H(S)$. Averaging over all unordered pairs therefore yields $\mathbb{E}[\widehat{H}(S)] = H(S)$. Combining this equality with Proposition 1 and Theorem 2 gives the expectation identity in Appendix D.4.

B.4 Proof of Proposition 1

Proof. For each main trajectory b, the main output and each branch output are conditionally independent under their shared information. The kernel representation therefore gives

$$\begin{aligned}
\mathbb{E}[k(Y^{(b)}, \widetilde{Y}_{t,C}^{(b)})] &= \mathbb{E}[\|\mu_t(C_t^{(b)})\|^2], \\
\mathbb{E}[k(Y^{(b)}, \widetilde{Y}_{t,R}^{(b)})] &= \mathbb{E}[\|\mu_t(C_t^{(b)}, R_t^{(b)})\|^2], \\
\mathbb{E}[k(Y^{(b)}, \widetilde{Y}_{t,Z}^{(b)})] &= \mathbb{E}[\|\mu_t(C_t^{(b)}, R_t^{(b)}, Z_t^{(b)})\|^2].
\end{aligned}$$

Taking the differences between adjacent identities and applying the tower property yields

$$\mathbb{E}[\widehat{a}_{t,b}^{\text{orch}}] = A_t^{\text{orch}}, \qquad \mathbb{E}[\widehat{a}_{t,b}^{\text{agent}}] = A_t^{\text{agent}}.$$

Averaging over the B main trajectories proves the proposition. □

B.5 Proof of Theorem 3

Proof. For fixed t and ℓ, the averaging formula in Equation 5 and the unbiasedness established in Proposition 1 give

$$\widehat{A}_t^\ell - A_t^\ell = \frac{1}{B} \sum_{b=1}^{B} \left(\widehat{a}_{t,b}^\ell - \mathbb{E}[\widehat{a}_{t,b}^\ell]\right).$$

Since $k \in \{0, 1\}$, each kernel difference satisfies

$$-1 \le \widehat{a}_{t,b}^\ell \le 1.$$

The B variables are independent across main trajectories. Applying Hoeffding's inequality with an interval length of 2 gives

$$\mathbb{P}\left(\left|\widehat{A}_t^\ell - A_t^\ell\right| \geq \varepsilon\right) \leq 2\exp\left(-\frac{2B^2\varepsilon^2}{\sum_{b=1}^{B}(1-(-1))^2}\right) = 2\exp\left(-\frac{B\varepsilon^2}{2}\right).$$

Dependence between different steps on the same main trajectory does not affect this componentwise bound. Applying the union bound over the $2T$ pairs (t, ℓ) yields

$$\mathbb{P}\left(\max_{\substack{1\leq t\leq T\\ \ell\in\{\text{orch},\text{agent}\}}}\left|\widehat{A}_t^\ell - A_t^\ell\right| \geq \varepsilon\right) \leq \sum_{t=1}^{T}\sum_{\ell}\mathbb{P}\left(\left|\widehat{A}_t^\ell - A_t^\ell\right| \geq \varepsilon\right) \leq 4T\exp\left(-\frac{B\varepsilon^2}{2}\right).$$

Setting $\varepsilon = \sqrt{2\log(4T/\delta)/B}$ yields Equation 6. □

B.6 Proof of Theorem 4

Proof. Each single-trajectory estimate is the difference between two same-cluster kernel values. Since $k(\cdot,\cdot) \in \{0,1\}$,

$$\widehat{a}_{t,b}^\ell \in \{-1,0,1\}, \qquad \left(\widehat{a}_{t,b}^\ell\right)^2 = \mathbf{1}\left\{\widehat{a}_{t,b}^\ell \neq 0\right\}.$$

Hence $\mathbb{E}[(\widehat{a}_{t,b}^\ell)^2] = \beta_t^\ell$. Together with the unbiasedness identity $\mathbb{E}[\widehat{a}_{t,b}^\ell] = A_t^\ell$, this gives

$$\mathrm{Var}\left(\widehat{a}_{t,b}^\ell\right) = \mathbb{E}\left[\left(\widehat{a}_{t,b}^\ell\right)^2\right] - \left(\mathbb{E}[\widehat{a}_{t,b}^\ell]\right)^2 = \beta_t^\ell - (A_t^\ell)^2.$$

For concentration, define $X_{t,b}^\ell := \widehat{a}_{t,b}^\ell - A_t^\ell$. Independence of the sampling bundles implies that $X_{t,1}^\ell, \ldots, X_{t,B}^\ell$ are independent, with

$$\mathbb{E}[X_{t,b}^\ell] = 0, \qquad |X_{t,b}^\ell| \leq 2, \qquad \mathrm{Var}(X_{t,b}^\ell) = \beta_t^\ell - (A_t^\ell)^2.$$

Since $\widehat{A}_t^\ell - A_t^\ell = B^{-1}\sum_{b=1}^{B} X_{t,b}^\ell$, the two-sided Bernstein inequality yields

$$\mathbb{P}\left(\left|\widehat{A}_t^\ell - A_t^\ell\right| \geq \varepsilon\right) \leq 2\exp\left(-\frac{B\varepsilon^2}{2[\beta_t^\ell - (A_t^\ell)^2] + 4\varepsilon/3}\right).$$

□

B.7 Brier Score Equivalence

For $S^\star \sim q^\star$, expanding the expected Brier score gives

$$\mathbb{E}\left[\sum_{s\in\mathcal{S}}(q_\theta(s) - \mathbf{1}\{S^\star = s\})^2\right] = \|q_\theta\|^2 - 2\langle q_\theta, q^\star\rangle + 1 = \|q_\theta - q^\star\|^2 + 1 - \|q^\star\|^2.$$

The last expression equals $\mathcal{L}_{\text{cal}}(\theta) + H(S^\star)$, proving the identity in Appendix D.6.

B.8 Proof of Theorem 5

Proof. Since $q^\star$ is independent of θ, differentiating the calibration objective coordinatewise gives

$$\nabla_\theta \mathcal{L}_{\text{cal}}(\theta) = 2\sum_{s\in\mathcal{S}}[q_\theta(s) - q^\star(s)]\,\nabla_\theta q_\theta(s).$$

By the system formulation, the agent output distributions and deterministic execution mechanisms are fixed with respect to θ, while Assumption 4 ensures that the orchestration policy is differentiable. Applying the likelihood-ratio identity to $q_\theta(s) = \mathbb{E}_\theta[\mathbf{1}\{\gamma(Y) = s\}]$ yields

$$\nabla_\theta q_\theta(s) = \sum_{t=1}^{T}\mathbb{E}_\theta\left[\mathbf{1}\{\gamma(Y) = s\}\nabla_\theta \log \pi_\theta(R_t \mid C_t)\right].$$

Substituting this identity into the objective gradient and using the definition of ϕ gives

$$\nabla_\theta \mathcal{L}_{\text{cal}}(\theta) = 2 \sum_{t=1}^{T} \mathbb{E}_\theta \left[\langle q_\theta - q^\star, \phi(Y) \rangle \nabla_\theta \log \pi_\theta(R_t \mid C_t) \right].$$

Conditioning on (C_t, R_t) replaces $\phi(Y)$ by $\mu_t(C_t, R_t)$:

$$\nabla_\theta \mathcal{L}_{\text{cal}}(\theta) = 2 \sum_{t=1}^{T} \mathbb{E}_\theta \left[\langle q_\theta - q^\star, \mu_t(C_t, R_t) \rangle \nabla_\theta \log \pi_\theta(R_t \mid C_t) \right].$$

The conditional score has zero expectation because

$$\mathbb{E}_\theta \left[\nabla_\theta \log \pi_\theta(R_t \mid C_t) \mid C_t \right] = \sum_{r=1}^{N} \nabla_\theta \pi_\theta(r \mid C_t) = \nabla_\theta 1 = 0.$$

Consequently,

$$\mathbb{E}_\theta \left[\langle q_\theta - q^\star, \mu_t(C_t) \rangle \nabla_\theta \log \pi_\theta(R_t \mid C_t) \right] = 0.$$

Subtracting this zero-expectation baseline from each stepwise term proves the theorem. □

B.9 Proof of Theorem 6

Proof. For every b, the leave-one-out estimator depends only on the other $B - 1$ main trajectories. It is therefore independent of the main trajectory and continuation branches associated with b, and

$$\mathbb{E}[\widehat{q}_{-b}] = \frac{1}{B-1} \sum_{j=1, j \neq b}^{B} \mathbb{E}[\phi(Y^{(j)})] = q_\theta.$$

By the conditional sampling construction, the two reused branches satisfy

$$\mathbb{E}\left[\phi(\widetilde{Y}_{t,R}^{(b)}) \mid C_t^{(b)}, R_t^{(b)} \right] = \mu_t(C_t^{(b)}, R_t^{(b)}), \quad \mathbb{E}\left[\phi(\widetilde{Y}_{t,C}^{(b)}) \mid C_t^{(b)}, R_t^{(b)} \right] = \mu_t(C_t^{(b)}).$$

The second identity holds because the context-conditioned branch resamples the agent independently of $R_t^{(b)}$ given $C_t^{(b)}$. Combining these identities with the independence of $\widehat{q}_{-b}$ gives

$$\begin{aligned} &\mathbb{E}\left[\left\langle \widehat{q}_{-b} - q^\star, \phi(\widetilde{Y}_{t,R}^{(b)}) - \phi(\widetilde{Y}_{t,C}^{(b)}) \right\rangle \nabla_\theta \log \pi_\theta(R_t^{(b)} \mid C_t^{(b)}) \right] \\ &= \mathbb{E}\left[\left\langle q_\theta - q^\star, \mu_t(C_t^{(b)}, R_t^{(b)}) - \mu_t(C_t^{(b)}) \right\rangle \nabla_\theta \log \pi_\theta(R_t^{(b)} \mid C_t^{(b)}) \right]. \end{aligned}$$

Summing over t and multiplying the sum over b by $2/B$ recovers the calibration gradient in Equation 8. □

C Implementation Details

C.1 Agent Configurations

Table 3 lists the five agents and the execution horizon for each dataset. Each agent receives a specialized system prompt, the question, and at most four previous agent outputs. The exact prompts and worked examples are provided in `ecomas/registry.py`.

C.2 Generation, Parsing, and Evaluation

The reported RQ1 run uses temperature 0.7 and at most 768 generated tokens; formal RQ3 uses temperature 0.7, at most 768 tokens, $B = 32$ for attribution, $B = 64$ for intervention, and five repetitions. Positive-temperature sampling uses top-$p = 0.9$ and top-$k = 20$. Seeds and backbone-specific settings are retained in the run manifests. RQ4 trains each learned policy on the same 300 examples and evaluates 100 fixed questions with 10 stochastic executions per question.

The final answer is parsed from the last valid `FINAL ANSWER` marker: an option for MMLU-Pro, an exact expression for MATH-500, or one of the three NLI labels for ChaosNLI. Parser success is

Table 3: Agent configurations of the MAS for each dataset.

Dataset	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Steps
MMLU-Pro	Mathematics and Computation	Life Sciences	Social Institutions	Humanities and Behavior	Option Discrimination	4
MATH-500	Algebra	Geometry	Number Theory	Combinatorics and Probability	Functions and Inequalities	6
ChaosNLI	Formal Semantics	Lexical Syntax	Commonsense Reasoning	Pragmatic Explanation	Label Boundary Judgment	5

close to 100% in the reported formal experiments; invalid parses are recorded. In RQ3, an orchestration intervention replaces sampling at the selected step with the policy's most probable agent; an execution intervention uses greedy decoding. Paired continuations resample unrevealed decisions and responses with independent randomness. The branch cache is invalidated after policy updates and between intervention questions.

Semantic equivalence and calibration. The analysis assumes a fixed semantic mapping γ. Valid multiple-choice options and NLI labels have canonical identities; mathematical expressions use deterministic symbolic normalization. Bidirectional entailment and complete-link clustering are applied to remaining free-form answers, with pairwise decisions and conflicts recorded. Batch-dependent fallback decisions do not define a fixed γ; their incidence is recorded in the clustering audit. Invalid responses form a separate cluster for uncertainty estimation. For closed-label calibration, invalid predictions are recorded but omitted from valid-label coordinates; MATH-500 assigns out-of-support answers to its `__other__` bucket. The near-complete parser success makes the impact of invalid outputs small in the reported evaluations; strict equivalence to the full-space $\mathcal{L}_{\mathrm{cal}}$ requires zero invalid-output probability.

D Further Discussions

The following sections follow the main text: semantic representation, paired sampling and its bounds, calibration and sample reuse, evidence auditing, and diagnostic efficiency.

D.1 Properties of Second-Order Tsallis Entropy

The entropy $H(S) = 1 - \sum_{s\in\mathcal{S}} q_\theta(s)^2$ has five basic properties:

Non-negativity, $H(S) \geq 0$;

Definiteness, $H(S) = 0$ if and only if $q_\theta(s_0) = 1$ for some s_0;

Invariance under permutations of cluster labels;

Continuity in the probability vector (in the ℓ_1 norm if $\mathcal{S}$ is countable); and

Expansibility when zero-probability clusters are added. Hence uncertainty depends on semantic probabilities rather than arbitrary cluster names or unused categories.

Verification. Since $q_\theta(s) \geq 0$ and $\sum_s q_\theta(s) = 1$, we have $\sum_s q_\theta(s)^2 \leq 1$, with equality exactly when all probability mass lies in one cluster. Permuting labels only reorders the sum, and inserting a zero-probability cluster leaves it unchanged. For any probability vectors p, q, $\left|\sum_s p(s)^2 - \sum_s q(s)^2\right| \leq 2\|p-q\|_1$, which also establishes continuity for countably many clusters.

D.2 Positive Semidefiniteness of the Same-Cluster Kernel

The same-cluster kernel admits the feature representation used in the main text. We record its positive semidefiniteness explicitly because it justifies treating semantic disagreement through feature-space inner products.

Proposition 2. *The same-cluster kernel $k(y, y') = \mathbf{1}\{\gamma(y) = \gamma(y')\}$ is positive semidefinite.*

Proof. Since $k(y, y') = \langle \phi(y), \phi(y') \rangle$, for any finite collection $y_1, \ldots, y_m$ and real coefficients $a_1, \ldots, a_m$,

$$\sum_{i=1}^{m} \sum_{j=1}^{m} a_i a_j k(y_i, y_j) = \left\| \sum_{i=1}^{m} a_i \phi(y_i) \right\|^2 \geq 0.$$

Thus the Gram matrix of k is positive semidefinite. □

D.3 Paired Sampling for Information Attribution

Algorithm 1 implements the three conditional continuations introduced in the main text. Its stored main outputs also support the separate system uncertainty estimate in Appendix D.4. Each fresh continuation follows the transition F and the final output extraction rule from the retained variables onward.

Algorithm 1 Unbiased Paired Sampling for Information Attribution

Input: Task input x, policy π_θ, agent distributions $\{p_r\}_{r=1}^N$, transition F, final output extraction rule, steps T, kernel k, and budget $B \geq 2$
Sampling rule: Use independent randomness across main trajectories and fresh randomness for each conditional continuation
for $b = 1, \ldots, B$ **do**
  Sample a complete main execution under $\mathbb{P}_\theta(\cdot \mid C_1 = x)$ and obtain $Y^{(b)}$
  Retain $\{C_t^{(b)}, R_t^{(b)}, Z_t^{(b)}\}_{t=1}^T$ ▷ Store the main trajectory
  for $t = 1, \ldots, T$ **do**
    Generate $\widetilde{Y}_{t,C}^{(b)}$ by retaining $C_t^{(b)}$, resampling the agent and its response, and continuing to step T
    Generate $\widetilde{Y}_{t,R}^{(b)}$ by retaining $C_t^{(b)}, R_t^{(b)}$, resampling its response, and continuing to step T
    if $t < T$ **then**
      Retain $(C_t^{(b)}, R_t^{(b)}, Z_t^{(b)})$; independently continue to obtain $\widetilde{Y}_{t,Z}^{(b)}$
    else
      $\widetilde{Y}_{T,Z}^{(b)} \leftarrow Y^{(b)}$ ▷ No remaining randomness
    end if
    $\widehat{a}_{t,b}^{\mathrm{orch}} \leftarrow k(Y^{(b)}, \widetilde{Y}_{t,R}^{(b)}) - k(Y^{(b)}, \widetilde{Y}_{t,C}^{(b)})$
    $\widehat{a}_{t,b}^{\mathrm{agent}} \leftarrow k(Y^{(b)}, \widetilde{Y}_{t,Z}^{(b)}) - k(Y^{(b)}, \widetilde{Y}_{t,R}^{(b)})$
  end for
end for
for $t = 1, \ldots, T$ **do**
  $\widehat{A}_t^{\mathrm{orch}} \leftarrow \frac{1}{B} \sum_{b=1}^B \widehat{a}_{t,b}^{\mathrm{orch}}, \quad \widehat{A}_t^{\mathrm{agent}} \leftarrow \frac{1}{B} \sum_{b=1}^B \widehat{a}_{t,b}^{\mathrm{agent}}$ ▷ Average stepwise contributions
end for
Output: $\{(\widehat{A}_t^{\mathrm{orch}}, \widehat{A}_t^{\mathrm{agent}})\}_{t=1}^T$, main outputs $Y^{(b)}$, and stored $C_t^{(b)}, R_t^{(b)}, \widetilde{Y}_{t,C}^{(b)}, \widetilde{Y}_{t,R}^{(b)}$ for calibration

D.4 Additional Estimation Identities and Concentration Bounds

For $B \geq 2$, the independent main outputs provide a second-order U-statistic estimator of system output uncertainty:

$$\widehat{H}(S) = \binom{B}{2}^{-1} \sum_{1 \leq i < j \leq B} \left[1 - k(Y^{(i)}, Y^{(j)}) \right].$$

The expectation identities proved in Appendix B.3 are $\mathbb{E}[\widehat{H}(S)] = H(S)$ and

$$\mathbb{E}\left[\widehat{H}(S) - \sum_{t=1}^{T} \left(\widehat{A}_t^{\mathrm{orch}} + \widehat{A}_t^{\mathrm{agent}} \right) \right] = 0.$$

For completeness, the componentwise Hoeffding bound underlying Theorem 3 is

$$\mathbb{P}\left(\left|\widehat{A}_t^\ell - A_t^\ell\right| \geq \varepsilon\right) \leq 2\exp\left(-\frac{B\varepsilon^2}{2}\right),$$

and the corresponding simultaneous form is:

$$\mathbb{P}\left(\max_{\substack{1\leq t\leq T\\ \ell\in\{\text{orch},\text{agent}\}}} \left|\widehat{A}_t^\ell - A_t^\ell\right| \geq \varepsilon\right) \leq 4T\exp\left(-\frac{B\varepsilon^2}{2}\right).$$

Theorem 4 also yields the looser bound obtained by replacing $\beta_t^\ell - (A_t^\ell)^2$ with β_t^ℓ. Taking the tighter of the Hoeffding and Bernstein guarantees gives

$$\begin{aligned}\mathbb{P}\left(\left|\widehat{A}_t^\ell - A_t^\ell\right| \geq \varepsilon\right) &\leq \min\left\{2\exp\left(-\frac{B\varepsilon^2}{2}\right), 2\exp\left(-\frac{B\varepsilon^2}{2[\beta_t^\ell - (A_t^\ell)^2] + 4\varepsilon/3}\right)\right\}\\ &= 2\exp\left[-\frac{B\varepsilon^2}{\min\{2,\ 2[\beta_t^\ell - (A_t^\ell)^2] + 4\varepsilon/3\}}\right].\end{aligned} \tag{10}$$

D.5 Concentration of System Uncertainty Estimates

In addition to the guarantees for information contribution estimates, we establish a concentration bound for the system uncertainty estimator. Recall the U-statistic definition in Appendix D.4. Although the pairwise terms share outputs, the following bound follows by expressing this estimator as an average over disjoint pairings.

Theorem 7. *For any $B \geq 2$ and $\varepsilon > 0$,*

$$\mathbb{P}\left(\left|\widehat{H}(S) - H(S)\right| \geq \varepsilon\right) \leq 2\exp\left(-2\left\lfloor\frac{B}{2}\right\rfloor\varepsilon^2\right).$$

Proof. Let $m = \lfloor B/2\rfloor$. For any permutation σ of $\{1,\ldots,B\}$, define

$$V_\sigma := \frac{1}{m}\sum_{r=1}^{m}\left[1 - k\left(Y^{(\sigma(2r-1))}, Y^{(\sigma(2r))}\right)\right].$$

Averaging over all permutations assigns equal weight to every unordered sample pair, giving

$$\widehat{H}(S) = \frac{1}{B!}\sum_{\sigma} V_\sigma.$$

For each fixed σ, the m summands involve disjoint pairs of independent outputs. They are therefore independent, take values in $[0,1]$, and have expectation $H(S)$. Hoeffding's lemma gives, for every $\lambda \in \mathbb{R}$,

$$\mathbb{E}\left[e^{\lambda(V_\sigma - H(S))}\right] \leq \exp\left(\frac{\lambda^2}{8m}\right).$$

By convexity of the exponential function,

$$\mathbb{E}\left[e^{\lambda(\widehat{H}(S) - H(S))}\right] \leq \frac{1}{B!}\sum_{\sigma}\mathbb{E}\left[e^{\lambda(V_\sigma - H(S))}\right] \leq \exp\left(\frac{\lambda^2}{8m}\right).$$

The Chernoff bound, optimized at $\lambda = 4m\varepsilon$, yields

$$\mathbb{P}\left(\widehat{H}(S) - H(S) \geq \varepsilon\right) \leq \inf_{\lambda>0}\exp\left(-\lambda\varepsilon + \frac{\lambda^2}{8m}\right) = e^{-2m\varepsilon^2}.$$

Applying the same argument to $H(S) - \widehat{H}(S)$ and combining the two tails gives the stated bound. □

D.6 Calibration Objective and Leave-One-Out Estimation

Let $S^\star = \gamma(Y^\star)$ for $Y^\star \sim \mathbb{P}^\star$. Since ϕ is a one-hot feature map, q_θ and $q^\star$ are the semantic-cluster probability vectors of the system and true-answer distributions. Consequently,

$$\mathcal{L}_{\mathrm{cal}}(\theta) = \|q_\theta - q^\star\|^2 = \sum_{s\in\mathcal{S}} [q_\theta(s) - q^\star(s)]^2.$$

The objective is also the policy-dependent part of the expected Brier score:

$$\mathbb{E}_{S^\star\sim q^\star}\left[\sum_{s\in\mathcal{S}} \left(q_\theta(s) - \mathbf{1}\{S^\star = s\}\right)^2\right] = \mathcal{L}_{\mathrm{cal}}(\theta) + H(S^\star).$$

Here $H(S^\star) = 1 - \sum_{s\in\mathcal{S}} q^\star(s)^2$ is fixed by the true-answer distribution. Thus minimizing the expected Brier score with respect to θ is equivalent to minimizing $\mathcal{L}_{\mathrm{cal}}(\theta)$.

For $B \geq 2$, the leave-one-out estimate used for the b-th gradient term is

$$\widehat{q}_{-b} := \frac{1}{B-1} \sum_{\substack{1\leq j\leq B \\ j\neq b}} \phi(Y^{(j)}). \tag{11}$$

Excluding trajectory b makes the mean independent of its main execution and continuation branches. The formal unbiasedness proof is provided in Appendix B.9.

D.7 Sample Reuse for Calibration Gradient Estimation

Algorithm 2 reuses the main outputs and the context- and agent-conditioned branches produced by Algorithm 1. The leave-one-out mean prevents the output-distribution estimate from sharing trajectory-specific randomness with its gradient term.

Algorithm 2 Unbiased Calibration Gradient Estimation

Input: Orchestration policy π_θ, true-answer distribution vector $q^\star$, semantic feature map ϕ, and sampling budget $B \geq 2$
Reused samples: Main outputs $Y^{(b)}$, contexts $C_t^{(b)}$, selected agents $R_t^{(b)}$, and branch outputs $\widetilde{Y}_{t,R}^{(b)}, \widetilde{Y}_{t,C}^{(b)}$ from Algorithm 1 under π_θ
Initialize: $\widehat{\nabla_\theta \mathcal{L}_{\mathrm{cal}}} \leftarrow \mathbf{0}$
for $b = 1, \ldots, B$ **do**
  $\widehat{q}_{-b} \leftarrow \frac{1}{B-1} \sum_{j=1,\, j\neq b}^{B} \phi(Y^{(j)})$ ▷ Exclude the current trajectory
  for $t = 1, \ldots, T$ **do**
    $\widehat{\nabla_\theta \mathcal{L}_{\mathrm{cal}}} \leftarrow \widehat{\nabla_\theta \mathcal{L}_{\mathrm{cal}}} + \left\langle \widehat{q}_{-b} - q^\star,\, \phi(\widetilde{Y}_{t,R}^{(b)}) - \phi(\widetilde{Y}_{t,C}^{(b)}) \right\rangle \nabla_\theta \log \pi_\theta(R_t^{(b)} \mid C_t^{(b)})$
  end for
end for
$\widehat{\nabla_\theta \mathcal{L}_{\mathrm{cal}}} \leftarrow \frac{2}{B} \widehat{\nabla_\theta \mathcal{L}_{\mathrm{cal}}}$ ▷ Scale the accumulated gradient
Output: Calibration gradient estimate $\widehat{\nabla_\theta \mathcal{L}_{\mathrm{cal}}}$

D.8 Calibration Reachability under Orchestration

The preceding estimator is unbiased for the calibration gradient, but achieving zero calibration error also depends on which semantic answer distributions the orchestration policy can induce. We examine this representational constraint and distinguish it from the error due to incomplete optimization.

For the fixed input $C_1 = x$, let Θ denote the orchestration parameter space and define the set of attainable semantic-cluster probability vectors as

$$\mathcal{Q}_\Theta := \{q_\theta : \theta \in \Theta\}.$$

As in the calibration analysis, the agent output distributions, context update function, and final output rule remain fixed. Thus, $\mathcal{Q}_\Theta$ contains the distributions attainable by varying only the orchestration policy. Their minimum distance from the true-answer distribution is

$$\mathrm{dist}(q^\star, \mathcal{Q}_\Theta) := \inf_{q \in \mathcal{Q}_\Theta} \|q - q^\star\|.$$

Proposition 3. *Suppose that $\mathcal{S}$ is finite and $\mathcal{Q}_\Theta$ is nonempty and closed. Then*

$$\min_{\theta \in \Theta} \mathcal{L}_{\mathrm{cal}}(\theta) = \mathrm{dist}^2(q^\star, \mathcal{Q}_\Theta),$$

and zero calibration error is attainable if and only if $q^\star \in \mathcal{Q}_\Theta$.

Let $\widehat{\theta} \in \Theta$ denote the parameters returned by an optimizer. If its objective value is within $\varepsilon_{\mathrm{opt}} \geq 0$ of the minimum, then

$$\mathrm{dist}^2(q^\star, \mathcal{Q}_\Theta) \leq \mathcal{L}_{\mathrm{cal}}(\widehat{\theta}) \leq \mathrm{dist}^2(q^\star, \mathcal{Q}_\Theta) + \varepsilon_{\mathrm{opt}}.$$

Proof. Since $\mathcal{S}$ is finite, its probability simplex is compact. The nonempty closed subset $\mathcal{Q}_\Theta$ is therefore compact, so the continuous function $\|q - q^\star\|^2$ attains its minimum on this set. By the definitions of the calibration objective and $\mathcal{Q}_\Theta$,

$$\min_{\theta \in \Theta} \mathcal{L}_{\mathrm{cal}}(\theta) = \min_{q \in \mathcal{Q}_\Theta} \|q - q^\star\|^2 = \mathrm{dist}^2(q^\star, \mathcal{Q}_\Theta).$$

The attained minimum is zero exactly when $q = q^\star$ for some $q \in \mathcal{Q}_\Theta$.

Because $q_{\widehat{\theta}} \in \mathcal{Q}_\Theta$, its calibration error is at least this minimum. The assumed optimization accuracy gives

$$\mathcal{L}_{\mathrm{cal}}(\widehat{\theta}) - \mathrm{dist}^2(q^\star, \mathcal{Q}_\Theta) = \mathcal{L}_{\mathrm{cal}}(\widehat{\theta}) - \min_{\theta \in \Theta} \mathcal{L}_{\mathrm{cal}}(\theta) \leq \varepsilon_{\mathrm{opt}},$$

which establishes the upper bound. □

Remark. The distance term characterizes the minimum calibration error permitted by the chosen orchestration policy class and fixed execution mechanisms, while $\varepsilon_{\mathrm{opt}}$ bounds the additional error due to incomplete optimization. Increasing the sampling budget can improve gradient estimation but leaves the attainable set $\mathcal{Q}_\Theta$ unchanged. Reducing a positive minimum error requires changing this set, for example by enlarging the policy class or modifying the agents. The assumed optimization accuracy is separate from the finite-sample unbiasedness established above.

D.9 Evidence Ledger

Table 4 separates the runs supporting benchmark claims from checks of the estimator and execution protocol. All counts refer to questions unless otherwise stated.

Table 4: Evaluation scope and sampling budgets. The short probe is diagnostic only.

Study	Dataset	Questions	Protocol
RQ1: exact-tree validation	MMLU-Pro	28	$T = 4$; $B = 4, 8, 16, 32, 64$; 200 repeats
RQ3: faithfulness	all three	100 each	32 attribution; 64 intervention; five repeats
RQ4: calibration	all three	300 train; 100 test each	10 test executions/question
Execution probe	ChaosNLI	10 selected	$B = 4$; two repeats

The separate ChaosNLI probe exercises generation, clustering, and cumulative interventions with real-model inference; four questions have nonzero initial uncertainty. Its small sample does not support inference about the formal 100-question comparisons. Repeated executions within a question estimate its output distribution and are not independent question-level observations. Fixed question identifiers and random seeds are retained in the corresponding run configurations.

The benchmarks and backbones cover the stated multiple-choice, mathematical, and NLI settings. Claims about substantially larger models, tool use, or adaptive horizons require separate evaluations.

Table 5: Runtime on RQ1 diagnostic trees (seconds). Speedup is relative to exact enumeration.

Sampling Budget	Runtime (Orch)	Speedup (Orch)	Runtime (Agent)	Speedup (Agent)
Exact	2030.724	1×	2030.729	1×
$B = 4$	0.004082	497,461×	0.003613	562,112×
$B = 8$	0.008108	250,460×	0.007207	281,754×
$B = 16$	0.016153	125,717×	0.014248	142,526×
$B = 32$	0.032311	62,849×	0.028545	71,141×
$B = 64$	0.064190	31,636×	0.056640	35,853×

D.10 Computational Efficiency of Paired Sampling

Table 5 shows that paired sampling is over $30{,}000\times$ faster than exact enumeration on the RQ1 diagnostic trees for every reported budget. Runtime grows approximately linearly with B, as predicted by $O(BT^2)$ for fixed T. These timings cover the diagnostic tree computation; end-to-end costs also include LLM generation and semantic equivalence checks.