Sequential Probabilistic Uncertainty Estimation for Parallel Multi-Agent Reasoning Systems

Tunyu Zhang[1,†] **Zihao Zhao**[2] **Yusong Zhao**[2] **Haizhou Shi**[1] **Zhuohang Li**[3]

Haoxian Chen[4] **Hao Wang**[2,†,‡] **Dimitris N. Metaxas**[1,†,‡]

Abstract

LLM-based multi-agent systems (MAS) have attracted growing attention for improving reasoning through interaction among multiple agents. In this work, we focus on *parallel multi-agent reasoning systems*, where several agents solve the same problem over multiple rounds and aggregate their outputs into a final answer. Despite their strong reasoning performance, uncertainty estimation for such systems remains underexplored: the reliability of a MAS depends not only on individual generations, but also on how agents interact and evolve across rounds. We propose **SAUCE** (Sequential Agent Uncertainty through Consensus Evolution), a lightweight, training-free uncertainty estimator that formulates MAS uncertainty as sequential inference over a latent system-level belief. SAUCE aggregates round-level agreement and generation-uncertainty signals through a filtering-style update. Across five backbones, five benchmarks, and two MAS protocols, SAUCE improves misclassification detection, selective prediction, and calibration over a broad set of uncertainty estimation baselines, including standard log-likelihood-based methods and MAS-specific estimators.

Code: github.com/Tyrion58/SAUCE

Data: Tyrion279/SAUCE-Qwen3-4B-Rollouts

1 Introduction

Multi-agent systems (MAS) built with large language models (LLMs) have demonstrated strong capabilities in complex reasoning tasks [31, 51, 22, 67, 55]. A common design is *parallel multi-agent reasoning* [9, 12, 23, 36], such as debate-style systems [4, 12], where multiple agents solve the same problem and exchange responses over several rounds. In these systems, agents act in parallel within each round, while interactions evolve sequentially across rounds. As such systems are explored for complex reasoning, planning, and decision-making tasks [64, 7, 19, 30], this motivates the need for reliable system-level uncertainty estimation for multi-agent interactions.

Uncertainty estimation has been extensively studied for single LLMs, including sampling-based [14, 38, 28, 37], Bayesian [66, 61, 49], and internal signal-based approaches [63, 28, 46, 15]. Extending these methods to MAS, however, is non-trivial. Sampling-based and Bayesian approaches become costly when repeated across agents and rounds. Signal-based methods are more computationally practical because they rely on quantities already available during interaction, but they introduce a different challenge: these signals are heterogeneous across agents and fluctuate over rounds. For parallel reasoning MAS, uncertainty estimation must aggregate heterogeneous agent-level signals into a coherent system-level estimate while tracking how these signals change across rounds.

[1]Rutgers University. [2]University of Illinois Urbana-Champaign. [3]Vanderbilt University. [4]Columbia University.
[†]Correspondence to: Tunyu Zhang (ty.zhang@rutgers.edu), Hao Wang (hwangml@illinois.edu), Dimitris N. Metaxas (dmn@cs.rutgers.edu). [‡]Equal advising.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

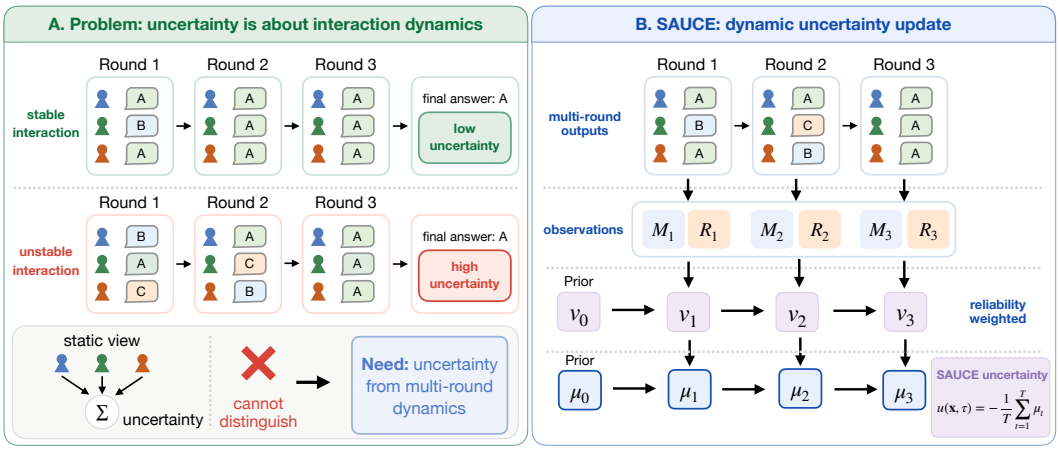


Figure 1: Overview. **(A)** Two MAS trajectories can reach the same final answer but differ greatly in reliability: a *stable* run whose outputs converge across rounds (top) is more trustworthy than an *unstable* run whose outputs oscillate (bottom). A static view that only considers the final output cannot distinguish them, motivating uncertainty estimation from interaction dynamics. **(B)** SAUCE models multi-round interaction as a sequence of round-level observations (M_t, R_t), where agreement is weighted by generation uncertainty, and performs a filtering-style update over a latent system-level belief. The resulting trajectory summary defines the SAUCE uncertainty score $u(\boldsymbol{x}, \tau)$ (Eqn. 11).

Crucially, multi-agent interaction is a *sequential* process [32, 12]. As agents exchange information over rounds, reliability may change, so uncertainty should be updated progressively rather than inferred from a single-step signal or a static summary (Fig. 1 (A)). Thus, MAS uncertainty is not merely a property of the final output, but of the *interaction process* itself. This raises a key question:

Do the interaction dynamics of a MAS inform its internal uncertainty?

We address this question by modeling MAS uncertainty as a **sequential inference problem** over interaction rounds. We propose **SAUCE** (**S**equential **A**gent **U**ncertainty through **C**onsensus **E**volution), a lightweight, training-free uncertainty estimator for parallel reasoning MAS. As shown in Fig. 1 (B), inspired by hierarchical Bayesian deep learning [57, 58], SAUCE treats each round as noisy evidence about an evolving system-level belief, rather than as an independent snapshot. Round-level agreement provides the evidence, while generation uncertainty determines how strongly the evidence should update the belief. This yields a filtering-style update that propagates the previous belief state and revises uncertainty according to the current round's generation uncertainty. As a result, uncertainty is inferred from the full interaction trajectory rather than from a single final answer. SAUCE uses only signals extracted from the MAS inference trace, requiring no additional sampling or access to model weights or hidden states. Empirically, SAUCE outperforms log-likelihood baselines and MAS-specific uncertainty estimators across Debate [12] and DyLAN [36] protocols.

Contributions. Our contributions are summarized as follows:

1. **A system-dynamics perspective for MAS uncertainty estimation.** We argue that uncertainty in parallel reasoning MAS should be estimated from how consensus and reliability evolve across rounds, rather than from a static final-output signal.
2. **A sequential inference framework for consensus aggregation.** We formulate MAS uncertainty estimation as a filtering-style update over a latent system-level belief, where round-level evidence is weighted by its estimated generation uncertainty.
3. **A lightweight estimator for practical MAS.** We instantiate SAUCE with majority-vote consensus and token-level generation entropy, yielding a training-free method that requires minimal model access and applies to both open- and closed-source models with top-k log-probabilities.

2 Preliminaries

In this section, we formalize the multi-agent reasoning process and define the problem of uncertainty estimation for MAS.

2.1 Multi-Agent Reasoning Trajectory

Given an input problem $\boldsymbol{x}$, a MAS consists of N agents $\mathcal{A} = \{A_1, \ldots, A_N\}$ that solve the problem through T rounds of interaction. At round t, agent A_i produces an output

$$\boldsymbol{z}_t^i \sim \pi_i(\cdot \mid \boldsymbol{x}, \boldsymbol{h}_t^i), \tag{1}$$

where $\boldsymbol{h}_t^i$ denotes the information available to agent A_i before round t, including its own previous outputs and the messages observed from other agents. The full multi-agent interaction trajectory is

$$\tau = \left(\boldsymbol{z}_1^{1:N}, \boldsymbol{z}_2^{1:N}, \ldots, \boldsymbol{z}_T^{1:N}\right), \tag{2}$$

which is sampled from the MAS rollout distribution induced by the agents and the interaction protocol. The final prediction is obtained by applying an aggregation function g to the input and the trajectory:

$$\widehat{\boldsymbol{y}} = g(\boldsymbol{x}, \tau). \tag{3}$$

Scope: Parallel Reasoning MAS. This paper focuses on *parallel* multi-agent reasoning [12, 32, 36], where all N agents generate responses in parallel at each round, conditioned on the same input $\boldsymbol{x}$ and the prior interaction history. In the protocols considered in this paper, the final prediction is typically obtained by aggregating the round-T parsed answers, e.g., by majority vote [59, 25, 6]. Debate [12] and DyLAN [36] are representative protocols of this form.

2.2 Sequential Uncertainty Estimation for MAS

Given an input $\boldsymbol{x}$, a reasoning trajectory τ, and the final prediction $\widehat{\boldsymbol{y}} = g(\boldsymbol{x}, \tau)$, our goal is to estimate a trajectory-level uncertainty score

$$u(\boldsymbol{x}, \tau) \in \mathbb{R}, \tag{4}$$

where a larger value indicates lower confidence in $\widehat{\boldsymbol{y}}$.

Dynamics in MAS. Unlike static confidence measures that summarize only the final output or aggregate signals without temporal structure, we model MAS uncertainty as a quantity induced by the interaction process. At each round, agent outputs induce round-level evidence about the system state, and the uncertainty is updated sequentially as new evidence arrives:

$$u_t = \Phi(u_{t-1}, e_t), \qquad u(\boldsymbol{x}, \tau) = \mathcal{A}(u_{1:T}), \tag{5}$$

where e_t denotes the evidence extracted from round t, u_t denotes the system uncertainty state after round t, Φ is a sequential update rule, and $\mathcal{A}$ maps the uncertainty trajectory to the final score.

Challenges. This dynamic view is essential because the final prediction arises from multi-round communication, where agents may reinforce, revise, or contradict one another. A static summary, such as the log-likelihood of the final answer or an order-invariant average of per-round signals, is blind to the *temporal order* of evidence: two trajectories with identical average signals but opposite dynamics may receive the same score despite carrying very different reliability. MAS uncertainty should therefore be updated progressively as new evidence emerges across rounds, rather than inferred from a single-step signal or a static aggregation.

Motivated by this view, we formulate MAS uncertainty estimation as a sequential inference problem. SAUCE instantiates this principle by integrating round-level signals into the uncertainty state u_t, leading to the method in Sec. 3 and the probabilistic interpretation in Sec. 4.

3 Method

We present SAUCE, a lightweight uncertainty estimator for parallel reasoning MAS. SAUCE extracts round-level evidence from generations of agents, and updates uncertainty across rounds through a filtering-style recursion. This recursion can be interpreted as **posterior uncertainty inference over a latent system-level belief**, with the full probabilistic view given in Sec. 4.

3.1 Round-Level Signals from Agent Outputs

At each interaction round t, the N agents generate outputs in parallel, $\{\boldsymbol{z}_t^i\}_{i=1}^N$. This parallel structure naturally exposes two round-level signals at no extra cost: a cross-agent *agreement* signal and a per-agent *generation-uncertainty* signal.

Agreement Signal M_t. For each output $\boldsymbol{z}_t^i$, we extract the parsed answer a_t^i. Let m_t denote the modal answer at round t. We define M_t as the *modal-answer share*, i.e., the fraction of agents whose answer matches m_t:

$$M_t \;=\; \tfrac{1}{N}\sum\nolimits_{i=1}^{N} \mathbb{1}[\,a_t^i = m_t\,]. \tag{6}$$

A higher M_t reflects stronger cross-agent consensus.

Generation-Uncertainty Signal R_t. Cross-agent agreement should be trusted more when the underlying generations are confident. We measure round-level generation uncertainty by averaging token-level predictive entropy across agents:

$$R_t \;=\; \tfrac{1}{N}\sum\nolimits_{i=1}^{N} H_{\text{tok}}(\boldsymbol{z}_t^i), \tag{7}$$

where $H_{\text{tok}}(\boldsymbol{z}_t^i)$ averages predictive entropy [37] over the tokens of $\boldsymbol{z}_t^i$, computed from the top-k log-probabilities. Larger R_t indicates higher generation uncertainty, so the round-level agreement should be trusted less. Together, (M_t, R_t) captures two sources of MAS uncertainty: cross-agent disagreement, reflected by low M_t, and generation uncertainty, reflected by high R_t.

3.2 Sequential Uncertainty Update

SAUCE maintains a round-level confidence estimate μ_t throughout the multi-agent interaction, together with a stability weight v_t that controls how strongly new evidence updates this estimate. Both quantities are initialized by hyperparameters, μ_0 and v_0. The final uncertainty score is obtained by negating the temporal average of μ_t. This sequential estimation process alternates between two steps: confidence *propagation* and reliability-weighted confidence *update*.

Propagation. At round t, SAUCE first propagates the previous state:

$$\widetilde{\mu}_t = \lambda \mu_{t-1}, \qquad \widetilde{v}_t = \lambda^2 v_{t-1} + q, \qquad t \in [1, T], \tag{8}$$

where $\lambda \in (0, 1)$ controls temporal coupling across rounds, and q captures additional uncertainty introduced during interaction. The two limits motivate this open interval: as $\lambda \to 0$ the state resets each round (no temporal memory), and as $\lambda \to 1$ the previous belief is retained without decay; an interior λ balances these two extremes.

SAUCE then combines the propagated state with the current round signals through a reliability-dependent weight:

$$\alpha_t \;=\; \widetilde{v}_t/(\widetilde{v}_t + R_t), \quad t \in [1, T]. \tag{9}$$

Update. The weight α_t determines how much SAUCE trusts the current round. When the round is reliable, R_t is small and α_t is large, so the estimate moves quickly toward the current consensus signal M_t. When the round is noisy, α_t becomes small, and SAUCE retains more of the propagated state. The update is

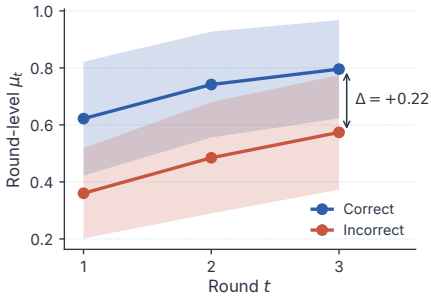


Figure 2: Round-level μ_t on Llama-3.1-8B / MATH-500. Both classes grow more confident with each round, but correct trajectories converge to a consistently higher consensus. Shading shows ± 1 standard deviation across trajectories.

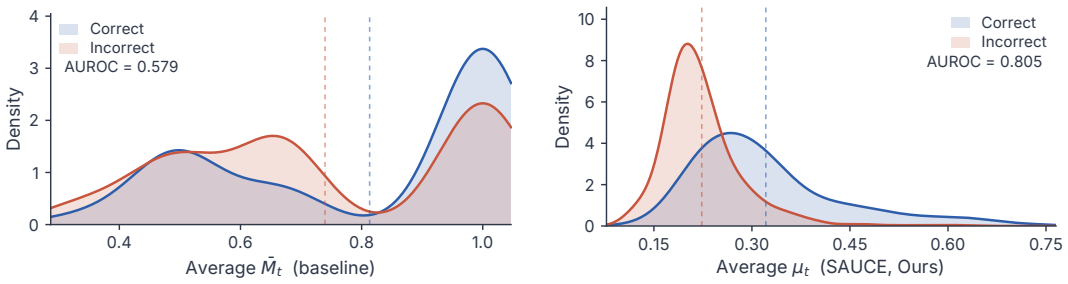
$$\mu_t \;=\; \widetilde{\mu}_t + \alpha_t(M_t - \widetilde{\mu}_t), \quad v_t \;=\; (1 - \alpha_t)\,\widetilde{v}_t. \tag{10}$$

This recursion has the same form as **the posterior mean and variance update** of a linear-Gaussian filter with effective noise R_t, as formally derived in Sec. 4.

We show the evolution of round-level confidence μ_t in Fig. 2. As the debate proceeds, μ_t gradually increases, indicating that the system becomes more confident over rounds. Moreover, correct samples consistently exhibit higher average μ_t than incorrect samples, suggesting that the sequential confidence estimate remains informative throughout the interaction.

Finally, SAUCE aggregates the consensus estimate across the trajectory into a per-record uncertainty score:

$$u(\boldsymbol{x}, \tau) \;=\; -\tfrac{1}{T}\sum\nolimits_{t=1}^{T} \mu_t. \tag{11}$$



(a) Baseline $\bar{M}_t$. (b) SAUCE μ_t.

Figure 3: Distribution of (a) the baseline agreement signal $\bar{M}_t$ and (b) SAUCE's posterior-mean estimate μ_t (each averaged over rounds) for correct vs. incorrect MAS outputs (MMLU-Pro, Gemini-2.5-FL, DyLAN). SAUCE sharpens the separation between classes (AUROC $0.579 \rightarrow 0.805$). Densities are normalized independently in each panel.

We use the temporal average rather than the terminal value μ_T so that the score reflects the full deliberation process.

Remark. As shown in Fig. 3, directly averaging M_t yields mixed and highly overlapping distributions for correct and incorrect trajectories, whereas SAUCE produces a much cleaner separation. This highlights the importance of sequentially modeling how confidence evolves across rounds.

4 Theoretical Analysis

We provide a probabilistic interpretation of SAUCE by viewing multi-agent interaction as a sequential inference process over a latent system belief. This perspective is inspired by the hierarchical Bayesian deep learning framework [57, 58], where observable neural representations are connected to latent probabilistic variables through a hierarchical probabilistic graphical model [24]. As illustrated in Fig. 4, we treat agent outputs at each round as noisy observations of a latent belief and estimate uncertainty through posterior inference.

4.1 Sequential Inference over Latent System Beliefs

Let $s_t \in \mathbb{R}$ denote the latent belief of the multi-agent system at round t, representing the system's underlying belief about the correctness of its current solution.

Transition Model. We assume that the latent belief evolves over rounds according to a first-order Markov process [44, 39, 47],

$$p(s_t \mid s_{t-1}), \tag{12}$$

which captures the temporal evolution of the system belief during interaction.

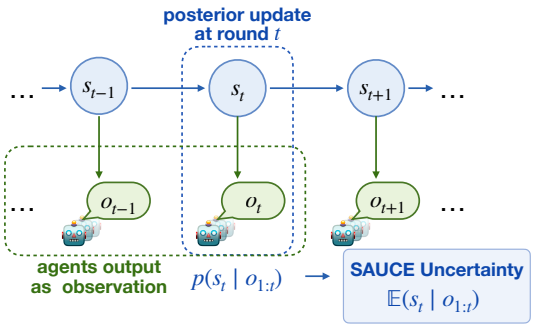


Figure 4: Probabilistic view of SAUCE. A latent system belief s_t evolves across rounds and emits a noisy observation o_t formed from the agents' outputs. SAUCE's per-round estimate is the posterior expectation $\mathbb{E}[s_t \mid o_{1:t}] = \mu_t$.

Observation Model. At round t, we extract a round-level signal o_t from the collection of agent outputs and treat it as a noisy observation of the latent belief state s_t. Formally, the observation process is described by

$$p(o_t \mid s_t), \tag{13}$$

which specifies the conditional distribution of the observed signal given the latent belief.

Sequential Inference. Given the transition and observation models, posterior inference over the latent belief follows the Bayesian filtering [47, 11, 26] recursion

$$p(s_t \mid o_{1:t}) \propto p(o_t \mid s_t) \int p(s_t \mid s_{t-1})\, p(s_{t-1} \mid o_{1:t-1})\, ds_{t-1}. \tag{14}$$

4.2 Linear-Gaussian Instantiation and Posterior Variance Update

To obtain a tractable update rule, we adopt a *linear-Gaussian* instantiation of the framework above. Specifically, we assume the latent belief evolves as

$$s_t = \lambda\, s_{t-1} + \epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0, q), \tag{15}$$

where $\lambda \in (0, 1)$ controls the temporal dependence across rounds and q is the process noise that captures additional uncertainty introduced during interaction. The latent belief therefore evolves smoothly while being continually shaped by the accumulated interaction history. Treating λ as a constant is a deliberate simplification, which we justify empirically through ablations in Appendix C.1.

Proposition 4.1. *Under the linear-Gaussian instantiation above, let the observation be $o_t \mid s_t \sim \mathcal{N}(s_t, r_t^2)$. If the posterior belief remains Gaussian, $s_t \mid o_{1:t} \sim \mathcal{N}(\mu_t, \sigma_t^2)$, then the posterior mean and variance satisfy*

$$\mu_t = \widetilde{\mu}_t + \alpha_t\big(o_t - \widetilde{\mu}_t\big), \tag{16}$$

$$\sigma_t^2 = \left(\tfrac{1}{\widetilde{\sigma}_t^2} + \tfrac{1}{r_t^2}\right)^{-1} = (1 - \alpha_t)\,\widetilde{\sigma}_t^2, \tag{17}$$

where the predictive moments and Kalman gain [26] are $\widetilde{\mu}_t = \lambda\,\mu_{t-1}$, $\widetilde{\sigma}_t^2 = \lambda^2\sigma_{t-1}^2 + q$, and $\alpha_t = \widetilde{\sigma}_t^2/(\widetilde{\sigma}_t^2 + r_t^2)$. See Appendix D.1 for the proof.

Proposition 4.1 shows that the SAUCE recursion (Eqn. 8–Eqn. 10) is the exact posterior of this linear-Gaussian model under the identification $o_t \leftrightarrow M_t$ and $r_t^2 \leftrightarrow R_t$. Substituting these into the proposition recovers the SAUCE update verbatim, with the agreement signal M_t acting as a noisy observation of the latent belief and the generation-uncertainty signal R_t as its observation-noise variance. The running estimate μ_t is therefore the posterior mean $\mathbb{E}[s_t \mid o_{1:t}]$, and the stability weight v_t its posterior variance $\mathrm{Var}[s_t \mid o_{1:t}]$.

Takeaway. SAUCE carries two quantities with a clean probabilistic reading. The round-level estimate μ_t *is* the posterior mean of the latent system belief s_t, so the per-record score $u(\boldsymbol{x}, \tau) = -\frac{1}{T}\sum_t \mu_t$ summarizes how the system's belief evolves across the entire trajectory rather than relying on the terminal round alone.

5 Experiments

We evaluate SAUCE on two standard uncertainty-estimation tasks for large language models: *(i)* **misclassification detection** and *(ii)* **selective prediction**. We further analyze **calibration** via the Brier score after Platt scaling, and include two diagnostic studies on long-debate robustness and the contribution of the sequential update.

5.1 Experimental Setup

Models. We evaluate five backbones spanning both open- and closed-source models. The open-source models include **Qwen3-4B** [65], **Llama-3.1-8B-Instruct** [17], and **Gemma-4-E4B-it** [16]; the closed-source models include **Gemini-2.5-Flash-Lite** [8] and **GPT-5.4-mini** [42]. Open-source models are served locally with vLLM [29], which provides per-token log-probabilities [40, 15]; for closed-source models, we use the top-k log-probabilities returned by the API.

MAS Architectures. We run two representative MAS protocols on every backbone. *(i)* **Debate** [12] uses $N = 3$ agents that first solve each problem independently and then exchange their full responses for $T = 3$ rounds. *(ii)* **DyLAN** [36] places the same agents in a dynamic agent network with a listwise ranker that retains the top $k = 2$ replies each round and an early-stopping consensus threshold of 0.667. For the main experiments, both protocols use up to $T = 3$ rounds; the consensus answer is the majority vote of the surviving agents in the final round. We probe longer debates separately in Sec. 5.4.

Benchmarks. We evaluate on five widely used reasoning suites covering mathematical problem solving, broad knowledge, and hard instruction-following. For *open-source* models we use **MATH-500** [33], **MMLU-Pro** [60], and **BBH** [50]. For *closed-source* models we use **OlympiadBench** [21], **MMLU-Pro** (subset), and **BBEH-mini** [27]. Benchmark sizes, formats, and answer verifiers are summarized in Appendix B.2.

Table 1: Misclassification detection (AUROC ↑) and selective prediction (AUARC ↑) on open-source models. **Boldface** and underlining denote the best and second-best performance.

Method	MATH-500				MMLU-Pro				BBH				Avg	
	Debate		DyLAN		Debate		DyLAN		Debate		DyLAN			
	ROC	ARC	ROC	ARC	ROC	ARC	ROC	ARC	ROC	ARC	ROC	ARC	ROC	ARC
Qwen3-4B														
PE	64.89	83.45	64.58	82.70	59.54	73.63	63.53	74.92	68.90	87.13	59.13	83.79	63.43	80.94
LL	65.93	83.14	66.36	84.08	68.39	78.49	69.88	79.27	69.56	87.62	60.61	83.82	66.79	82.74
UDPO	57.29	79.14	72.50	84.10	58.22	66.43	67.95	73.43	59.18	84.38	64.97	**88.00**	63.35	79.25
MATU (N=1)	62.11	82.90	63.30	84.05	58.55	73.11	57.81	73.46	58.59	84.56	63.62	85.76	60.66	80.64
SAUCE (Ours)	**68.81**	**84.12**	**73.11**	**85.60**	**77.05**	**82.72**	**78.14**	**83.37**	**73.68**	**89.09**	**70.18**	87.93	**73.50**	**85.47**
Llama-3.1-8B-Instruct														
PE	35.69	34.10	46.96	43.83	45.90	46.51	58.68	57.33	53.92	58.99	63.45	71.06	50.77	51.97
LL	39.20	35.68	47.62	46.44	48.69	47.12	57.69	56.99	57.78	61.44	63.66	71.53	52.44	53.20
UDPO	70.94	63.13	70.21	66.81	66.96	61.75	68.71	66.58	66.71	71.44	62.82	72.22	67.72	66.99
MATU (N=1)	55.55	49.77	73.85	70.71	52.85	53.23	63.26	65.32	61.31	64.20	66.24	72.51	62.18	62.62
SAUCE (Ours)	**83.63**	**72.12**	**80.69**	**74.46**	**75.83**	**71.00**	**74.44**	**72.30**	**74.07**	**74.81**	**72.35**	**76.94**	**76.84**	**73.61**
Gemma-4-E4B-it														
PE	58.20	87.47	**66.08**	88.45	69.33	83.64	75.50	87.58	53.67	84.69	54.47	79.97	62.88	85.30
LL	**62.43**	87.55	65.12	**88.58**	74.04	86.63	76.19	87.89	66.28	89.59	56.53	80.70	66.77	86.82
UDPO	55.83	85.47	55.53	84.45	62.01	75.64	72.28	81.36	64.01	93.83	80.36	**92.30**	65.00	85.51
MATU (N=1)	54.01	86.99	60.64	87.72	55.37	77.41	65.76	82.36	57.73	90.82	74.98	88.82	61.41	85.69
SAUCE (Ours)	62.29	**87.95**	64.05	87.89	**82.24**	**89.75**	**82.57**	**90.08**	**83.62**	**96.33**	**83.53**	91.94	**76.38**	**90.66**

Table 2: Misclassification detection and selective prediction on closed-source models. Metrics and notation follow Table 1. **Boldface** and underlining denote the best and second-best performance.

Method	Olympiad				MMLU-Pro				BBEH-mini				Avg	
	Debate		DyLAN		Debate		DyLAN		Debate		DyLAN			
	ROC	ARC	ROC	ARC	ROC	ARC	ROC	ARC	ROC	ARC	ROC	ARC	ROC	ARC
Gemini-2.5-Flash-Lite														
PE	65.51	66.86	66.35	65.23	48.71	73.35	70.22	89.09	49.98	19.13	41.73	16.06	57.08	54.95
LL	68.67	68.13	67.42	66.67	64.90	82.82	72.18	90.05	46.71	18.50	39.41	15.46	59.88	56.94
UDPO	63.08	48.89	81.58	71.05	54.59	77.52	57.83	82.71	59.23	25.12	**75.41**	**27.29**	65.29	55.43
MATU (N=1)	68.96	70.73	64.31	70.23	54.37	78.22	61.06	85.46	56.50	23.83	66.08	25.37	61.88	58.97
SAUCE (Ours)	**81.27**	**76.82**	**81.85**	**77.13**	**79.60**	**89.84**	**78.72**	**91.95**	**65.68**	**28.07**	67.48	24.44	**75.77**	**64.71**
GPT-5.4-mini														
PE	55.56	54.90	72.58	63.59	54.61	80.26	63.26	84.77	46.36	16.34	39.92	13.00	55.38	52.14
LL	70.25	61.87	75.03	65.69	61.57	83.56	65.51	85.97	41.67	14.18	45.03	13.76	59.84	54.17
UDPO	65.99	46.61	77.73	59.55	56.29	81.71	67.70	86.20	**58.58**	**19.59**	54.73	16.08	63.50	51.62
MATU (N=1)	53.45	54.91	65.23	63.74	56.44	81.52	60.31	85.04	48.64	16.96	**55.78**	**18.93**	56.64	53.51
SAUCE (Ours)	**85.15**	**74.46**	**83.46**	**72.57**	**70.63**	**86.44**	**73.83**	**88.29**	50.08	17.72	47.36	14.85	**68.42**	**59.06**

Baselines. We compare SAUCE with four training-free uncertainty estimators: **PE**, the predictive entropy of the final-round output [37]; **LL**, the log-likelihood of the final answer [40]; **UDPO** [52], a system-level variance score for debate; and **MATU** [5], an uncertainty estimator designed for multi-agent systems. The original MATU samples multiple MAS runs at substantially higher inference cost, so for budget parity we use its single-run variant ($N=1$).

Metrics. Following standard practice in uncertainty estimation [14, 5, 69], we measure ranking quality with **AUROC** ↑, which captures how well an estimator separates correct from incorrect MAS outputs, and **AUARC** ↑ [20, 2], which summarizes the accuracy–coverage curve obtained by rejecting increasingly uncertain inputs. For calibration, we report the **Brier score after Platt scaling** [3, 43, 18] with 5-fold cross-validation (lower is better). All three metrics are reported as percentages ($\times 100$). Formal definitions are given in Appendix B.1.

Implementation Details. All experiments use 3 agents and 3 interaction rounds, decoding with temperature 0.7, top-$p=0.9$, and an 8192-token budget per agent. SAUCE uses transferred hyperparameters: for each MAS architecture, we tune on a single calibration cell and reuse the same values across every other model–dataset combination. The transferred values are listed in Appendix B.3, and per-axis sensitivity is reported in Appendix C.2.

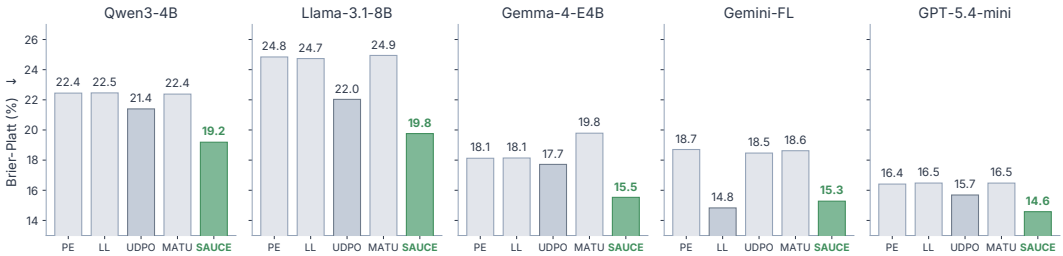


Figure 5: Brier score after Platt scaling (5-fold cross-validation) on MMLU-Pro under the Debate protocol, across all five backbones. SAUCE attains the lowest Brier-Platt score on four of the five backbones.

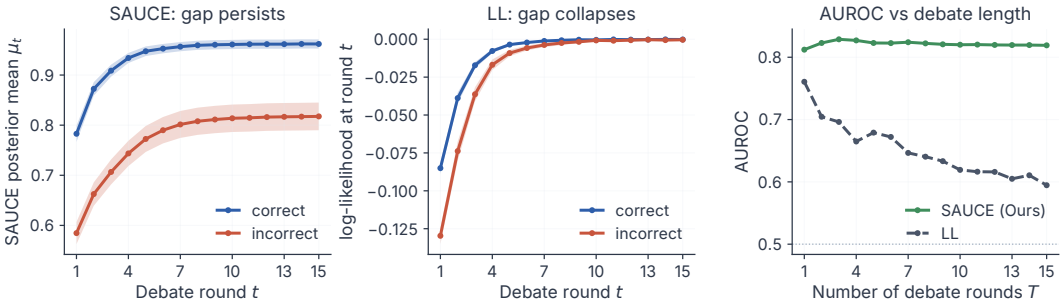


Figure 6: Per-round behavior on a 15-round Qwen3-4B Debate run on MMLU-Pro, grouped by final-answer correctness. ***Left:*** SAUCE posterior mean μ_t; the correct/incorrect gap persists through round 15. ***Middle:*** Average per-agent log-likelihood at round t; both classes become increasingly token-confident and the gap vanishes. ***Right:*** AUROC against final-consensus correctness as a function of debate length T; SAUCE plateaus while LL degrades toward chance (dashed line).

5.2 Uncertainty Estimation for Multi-Agent Systems

Tables 1 and 2 report AUROC and AUARC on the full grid of backbones, MAS protocols, and benchmarks of Sec. 5.1. SAUCE leads the large majority of cells and every per-backbone average; the gap to the strongest baseline holds under both ranking (AUROC) and accuracy–coverage (AUARC) views and on both Debate and DyLAN. Among the baselines, single-LLM scores (PE, LL) frequently land near or below chance, and UDPO and MATU each lead only on isolated (model, protocol) cells.

Results. Single-round token statistics break down under MAS: interaction reshapes the very entropy and log-likelihood that PE and LL rely on (Sec. 5.4). Specialized MAS estimators tie their gains to a single protocol and do not transfer across backbones or datasets. Only SAUCE's filtering-style aggregation across rounds wins consistently across backbones, datasets, and protocols, and the same property lets it carry over to the closed-source setting where only top-k token log-probabilities are exposed.

5.3 Uncertainty Estimation for Calibration

Fig. 5 reports the Brier score after Platt scaling on MMLU-Pro under the Debate protocol across all five backbones. SAUCE attains the lowest Brier-Platt score on four of the five backbones; on Gemini-FL it is within 0.5 pp of the strongest baseline (LL). Calibration results on additional benchmarks and settings are deferred to Appendix C.3.

Results. Brier penalizes both miscalibrated magnitudes and unresolved predictions, so this view complements AUROC/AUARC: a method can rank well yet pay a Brier penalty when its confidences cluster in a narrow band—trivially calibrated, but unable to separate reliable from unreliable trajectories. SAUCE's calibrated probabilities, by contrast, remain well spread across the probability range while still landing close to empirical accuracy. The score therefore has the right *shape* as well as the right *order*, and a two-parameter Platt transform suffices to turn it into a per-trajectory confidence usable as a downstream gating signal.

5.4 Robustness in Long Debates

Our main experiments use $T = 3$. To test whether SAUCE remains reliable in longer interactions, we run Qwen3-4B Debate on MMLU-Pro up to $T = 15$ and compare SAUCE with a last-round log-likelihood (LL) baseline at each truncation. As the debate proceeds, agents become increasingly token-confident on both correct and incorrect trajectories, causing the LL scores to concentrate and the correct/incorrect gap to vanish (Fig. 6, *middle*). In contrast, SAUCE's posterior mean μ_t also rises as agreement accumulates, but preserves a clear class separation throughout the sweep (*left*); correspondingly, SAUCE's AUROC plateaus after $T = 3$, whereas LL steadily degrades toward chance (*right*). These results suggest that retaining earlier-round evidence makes SAUCE more robust than last-round confidence scores when debate length increases.

5.5 Contribution of the Sequential Update

A natural concern is that SAUCE's improvement may come simply from combining agreement M_t with generation uncertainty R_t, rather than from the sequential update itself. Row (b) of Table 3 tests this possibility with a static precision-weighted aggregation $\frac{\sum_t M_t/R_t}{\sum_t 1/R_t}$.

Table 3: Component-wise ablation on Qwen3-4B / MMLU-Pro / Debate. Row (b) is the static joint of M_t and R_t; row (c) is the full sequential update with $\bar{\mu}$ readout. AUROC ↑ and Brier-Platt ↓.

Method	AUROC	Brier-Platt
Last round, $1-M_t$	58.4	21.3
(a) Round-mean of M_t	68.6	20.0
(b) Static joint with R_t	68.4	20.9
(c) **SAUCE** (Ours)	**77.1**	**19.2**

Results. Table 3 shows that the static joint score performs closely to the M_t-only round mean, while the full SAUCE update yields a clear gain. This indicates that SAUCE's improvement comes from the sequential update rather than a static combination of M_t and R_t. To further probe the contribution of interaction history, we compare simpler trajectory summaries and alternative round orders (Appendix C.4; statistical analyses in Appendix C.5) and examine whether answer-switch histories remain informative when final agreement is held fixed (Appendix C.6).

6 Related Work

Uncertainty Estimation for LLMs. Uncertainty estimation [56, 62] has been widely studied for hallucination detection and calibration in LLMs [14, 34, 1, 35, 54]. Existing methods include sampling-based approaches, such as Semantic Entropy [14] and SAR [13], which estimate uncertainty from multiple generations and their semantic variation; Bayesian approaches, which approximate posterior uncertainty through weight perturbations or low-rank adaptation [69, 49, 61, 66]; and internal-signal methods, which use token entropy, log-likelihood, or related generation statistics with little additional cost [15, 53, 45, 10, 13]. While effective for single LLMs, these methods do not directly address the structure of multi-agent reasoning, where uncertainty depends not only on individual generations but also on how agents interact and evolve across rounds. Our work addresses this gap by formulating uncertainty estimation in parallel multi-agent reasoning systems as a sequential inference problem over the interaction trajectory.

Parallel Multi-Agent Reasoning Systems. LLM-based multi-agent systems (MAS) have emerged as a promising paradigm for complex reasoning, where multiple agents collaborate, debate, or critique one another [51, 31, 48, 68]. We focus on *parallel multi-agent reasoning systems*, such as debate-style methods [6, 12, 9] and DyLAN [36], where agents solve the same input at each round, exchange information, and produce a verifiable final prediction. This setting naturally exposes system-level signals [41, 52], such as cross-agent agreement, disagreement, and their evolution over rounds. Recent work has begun to study uncertainty in MAS, including general frameworks for uncertainty quantification in LLM agents [41], uncertainty-driven optimization for mitigating debate collapse [52], and MATU [5], which estimates trajectory-level uncertainty from repeated MAS runs. In contrast, SAUCE is a lightweight, training-free estimator that updates uncertainty online from standard rollout signals within a single parallel multi-agent reasoning trajectory.

7 Conclusion

We studied uncertainty estimation in parallel multi-agent reasoning systems. We formulated MAS uncertainty as a sequential inference problem over the multi-round interaction trajectory, and proposed SAUCE: a filtering-style update that propagates a latent consensus state across rounds. Across Debate and DyLAN protocols on multiple reasoning benchmarks, SAUCE consistently outperforms log-likelihood baselines and recent MAS-specific uncertainty estimators on both ranking and calibration, and remains stable as debates extend over more rounds where last-round confidence scores collapse. More broadly, our results suggest that interaction dynamics are a useful primitive for reasoning about the reliability of multi-agent systems, opening the way toward uncertainty-aware orchestration and adaptive control of MAS.

Limitations. SAUCE focuses on *parallel multi-agent reasoning systems*, where agents produce comparable per-round answers. This assumption limits its direct applicability to more general agentic systems with heterogeneous roles, tool use, or open-ended planning, where round-level consensus is not naturally defined. Our current instantiation also uses simple agreement and uncertainty signals, namely modal-answer share and token-level predictive entropy; developing principled alternatives remains an important direction for future work.

Acknowledgement

We thank all reviewers, AC, and SAC for their valuable comments. HW is supported by Amazon Faculty Research Award, Microsoft AI & Society Fellowship, Nokia Faculty Research Award, NSF CAREER Award IIS-2340125, NIH grant R01CA297832, and NSF grant IIS-2127918. DM is funded by NSF: 2310966, 2235405, 2212301, 2003874, 1951890 and NIH 2R01HL127661.

Competing Interests. The authors declare no competing interests.

References

[1] Y. F. Bakman, D. N. Yaldiz, S. Kang, T. Zhang, B. Buyukates, S. Avestimehr, and S. P. Karimireddy. Reconsidering LLM uncertainty estimation methods in the wild. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 29531–29556, 2025.

[2] K. Boyd, K. H. Eng, and C. D. Page. Area under the precision-recall curve: point estimates and confidence intervals. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 451–466. Springer, 2013.

[3] G. W. Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

[4] C.-M. Chan, W. Chen, Y. Su, J. Yu, W. Xue, S. Zhang, J. Fu, and Z. Liu. Chateval: Towards better LLM-based evaluators through multi-agent debate. *arXiv preprint arXiv:2308.07201*, 2023.

[5] T. Chen, H. Yao, J. Chen, E. E. Papalexakis, and H. Wei. Every response counts: Quantifying uncertainty of LLM-based multi-agent systems through tensor decomposition. *arXiv preprint arXiv:2604.08708*, 2026.

[6] H. K. Choi, J. Zhu, and S. Li. Debate or vote: Which yields better decisions in multi-agent large language models? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[7] M. Coeckelbergh. LLMs, truth, and democracy: An overview of risks. *Science and Engineering Ethics*, 31(1):4, 2025.

[8] G. Comanici, E. Bieber, M. Schaekermann, I. Pasupat, N. Sachdeva, I. Dhillon, M. Blistein, O. Ram, D. Zhang, E. Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

[9] Y. Cui, H. Fu, H. Zhang, L. Wang, and C. Zuo. FREE-MAD: Consensus-free multi-agent debate. *arXiv preprint arXiv:2509.11035*, 2025.

[10] M. Darrin, P. Piantanida, and P. Colombo. Rainproof: An umbrella to shield text generator from out-of-distribution data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5831–5857, 2023.

[11] A. Doucet, N. De Freitas, and N. Gordon. An introduction to sequential monte carlo methods. In *Sequential Monte Carlo methods in practice*, pages 3–14. Springer, 2001.

[12] Y. Du, S. Li, A. Torralba, J. B. Tenenbaum, and I. Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first international conference on machine learning*, 2024.

[13] J. Duan, H. Cheng, S. Wang, A. Zavalny, C. Wang, R. Xu, B. Kailkhura, and K. 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)*, pages 5050–5063, 2024.

[14] S. Farquhar, J. Kossen, L. Kuhn, and Y. Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

[15] Y. Fu, X. Wang, Y. Tian, and J. Zhao. Deep think with confidence. In *NeurIPS 2025 Workshop on Efficient Reasoning*, 2025.

[16] Google DeepMind. google/gemma-4-e4b-it. https://huggingface.co/google/gemma-4-E4B-it, 2026. Accessed: 2026-05-07.

[17] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[18] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

[19] T. Guo, X. Chen, Y. Wang, R. Chang, S. Pei, N. V. Chawla, O. Wiest, and X. Zhang. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*, 2024.

[20] J. A. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

[21] C. He, R. Luo, Y. Bai, S. Hu, Z. L. Thai, J. Shen, J. Hu, X. Han, Y. Huang, Y. Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 2024.

[22] J. He, C. Treude, and D. Lo. LLM-based multi-agent systems for software engineering: Literature review, vision, and the road ahead. *ACM Transactions on Software Engineering and Methodology*, 34(5):1–30, 2025.

[23] T. Hu, Z. Tan, S. Wang, H. Qu, and T. Chen. Multi-agent debate for LLM judges with adaptive stability detection. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[24] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. In *Learning in graphical models*, pages 105–161. Springer, 1998.

[25] L. B. Kaesberg, J. Becker, J. P. Wahle, T. Ruas, and B. Gipp. Voting or consensus? decision-making in multi-agent debate. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11640–11671, 2025.

[26] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[27] M. Kazemi, B. Fatemi, H. Bansal, J. Palowitch, C. Anastasiou, S. V. Mehta, L. K. Jain, V. Aglietti, D. Jindal, Y. P. Chen, et al. Big-bench extra hard. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 26473–26501, 2025.

[28] J. Kossen, J. Han, M. Razzak, L. Schut, S. Malik, and Y. Gal. Semantic entropy probes: Robust and cheap hallucination detection in LLMs. *arXiv preprint arXiv:2406.15927*, 2024.

[29] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, and I. Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[30] X. Y. Lee, S. Akatsuka, L. Vidyaratne, A. Kumar, A. Farahat, and C. Gupta. Reliable decision-making for multi-agent LLM systems. Workshop paper, 2025.

[31] X. Li, S. Wang, S. Zeng, Y. Wu, and Y. Yang. A survey on LLM-based multi-agent systems: workflow, infrastructure, and challenges. *Vicinagearth*, 1(1):9, 2024.

[32] T. Liang, Z. He, W. Jiao, X. Wang, Y. Wang, R. Wang, Y. Yang, S. Shi, and Z. Tu. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 17889–17904, 2024.

[33] H. Lightman, V. Kosaraju, Y. Burda, H. Edwards, B. Baker, T. Lee, J. Leike, J. Schulman, I. Sutskever, and K. Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pages 39578–39601, 2024.

[34] L. Liu, Y. Pan, X. Li, and G. Chen. Uncertainty estimation and quantification for LLMs: A simple supervised approach. *arXiv preprint arXiv:2404.15993*, 2024.

[35] X. Liu, T. Chen, L. Da, C. Chen, Z. Lin, and H. Wei. Uncertainty quantification and confidence calibration in large language models: A survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 6107–6117, 2025.

[36] Z. Liu, Y. Zhang, P. Li, Y. Liu, and D. Yang. Dynamic LLM-agent network: An LLM-agent collaboration framework with agent team optimization. *arXiv preprint arXiv:2310.02170*, 2023.

[37] A. Malinin and M. Gales. Uncertainty estimation in autoregressive structured prediction. *arXiv preprint arXiv:2002.07650*, 2020.

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

[39] K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[40] K. Murray and D. Chiang. Correcting length bias in neural machine translation. In *Proceedings of the third conference on machine translation: research papers*, pages 212–223, 2018.

[41] C. Oh, S. Park, T. E. Kim, J. Li, W. Li, S. Yeh, S. Du, H. Hassani, P. Bogdan, D. Song, and S. 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)*, pages 16219–16250, 2026.

[42] OpenAI. GPT-5.4 mini and nano officially released. https://openai.com/zh-Hans-CN/index/introducing-gpt-5-4-mini-and-nano/, Mar. 2026. Accessed: 2026-05-07.

[43] J. Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[44] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[45] J. Ren, J. Luo, Y. Zhao, K. Krishna, M. Saleh, B. Lakshminarayanan, and P. J. Liu. Out-of-distribution detection and selective generation for conditional language models. *arXiv preprint arXiv:2209.15558*, 2022.

[46] J. Ren, Y. Zhao, T. Vu, P. J. Liu, and B. Lakshminarayanan. Self-evaluation improves selective generation in large language models. In *Proceedings on "I Can't Believe It's Not Better: Failure Modes in the Age of Foundation Models" at NeurIPS 2023 Workshops*, pages 49–64. PMLR, 2023.

[47] S. Särkkä and L. Svensson. *Bayesian filtering and smoothing*, volume 17. Cambridge university press, 2023.

[48] H. Shi, Y. Liu, B. Pang, Z. L. Liu, H. Wang, S. Savarese, C. Xiong, Y. Zhou, and S. Yavuz. Ssr: Socratic self-refine for large language model reasoning. *arXiv preprint arXiv:2511.10621*, 2025.

[49] H. Shi, Y. Wang, L. Han, H. Zhang, and H. Wang. Training-free bayesianization for low-rank adapters of large language models. *Advances in Neural Information Processing Systems*, 38, 2025.

[50] M. Suzgun, N. Scales, N. Schärli, S. Gehrmann, Y. Tay, H. W. Chung, A. Chowdhery, Q. V. Le, E. H. Chi, D. Zhou, and J. Wei. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.

[51] Y. Talebirad and A. Nadiri. Multi-agent collaboration: Harnessing the power of intelligent LLM agents. *arXiv preprint arXiv:2306.03314*, 2023.

[52] L. Tang, Y. Meng, J. Costa, Y. Zhang, M. Ye, and Z. Xi. The value of variance: Mitigating debate collapse in multi-agent systems via uncertainty-driven policy optimization. *arXiv preprint arXiv:2602.07186*, 2026.

[53] L. Van der Poel, R. Cotterell, and C. Meister. Mutual information alleviates hallucinations in abstractive summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5956–5965, 2022.

[54] R. Vashurin, E. Fadeeva, A. Vazhentsev, L. Rvanova, D. Vasilev, A. Tsvigun, S. Petrakov, R. Xing, A. Sadallah, K. Grishchenkov, et al. Benchmarking uncertainty quantification methods for large language models with lm-polygraph. *Transactions of the Association for Computational Linguistics*, 13:220–248, 2025.

[55] V. Venkataramani, H. Shi, Z. Ke, A. Xu, X. He, Y. Zhou, S. Yavuz, H. Wang, and S. Joty. Mas-prove: Understanding the process verification of multi-agent systems. In *ICML*, 2026.

[56] H. Wang, S. Xingjian, and D.-Y. Yeung. Natural-parameter networks: A class of probabilistic neural networks. In *NIPS*, pages 118–126, 2016.

[57] H. Wang and D.-Y. Yeung. Towards bayesian deep learning: A framework and some existing methods. *TKDE*, 28(12):3395–3408, 2016.

[58] H. Wang and D.-Y. Yeung. A survey on bayesian deep learning. *CSUR*, 53(5):1–37, 2020.

[59] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

[60] Y. Wang, X. Ma, G. Zhang, Y. Ni, A. Chandra, S. Guo, W. Ren, A. Arulraj, X. He, Z. Jiang, et al. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024.

[61] Y. Wang, H. Shi, L. Han, D. Metaxas, and H. Wang. BLOB: Bayesian low-rank adaptation by backpropagation for large language models. *Advances in neural information processing systems*, 37:67758–67794, 2024.

[62] Z. Wang and H. Wang. Variational imbalanced regression: Fair uncertainty quantification via probabilistic smoothing. In *NeurIPS*, 2023.

[63] Z. Xia, J. Xu, Y. Zhang, and H. Liu. A survey of uncertainty estimation methods on large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21381–21396, 2025.

[64] R. Xu, X. Li, S. Chen, and W. Xu. Nuclear deployed!: Analyzing catastrophic risks in decision-making of autonomous LLM agents. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 1226–1310, 2025.

[65] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[66] A. X. Yang, M. Robeyns, X. Wang, and L. Aitchison. Bayesian low-rank adaptation for large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[67] Y. Yu, Z. Yao, H. Li, Z. Deng, Y. Jiang, Y. Cao, Z. Chen, J. W. Suchow, Z. Cui, R. Liu, et al. Fincon: A synthesized LLM multi-agent system with conceptual verbal reinforcement for enhanced financial decision making. *Advances in Neural Information Processing Systems*, 37:137010–137045, 2024.

[68] G. Zhang, Y. Yue, Z. Li, S. Yun, G. Wan, K. Wang, D. Cheng, J. X. Yu, and T. Chen. Cut the crap: An economical communication pipeline for LLM-based multi-agent systems. *arXiv preprint arXiv:2410.02506*, 2024.

[69] T. Zhang, H. Shi, Y. Wang, H. Wang, X. He, Z. Li, H. Chen, L. Han, K. Xu, H. Zhang, D. Metaxas, and H. Wang. TokUR: Token-level uncertainty estimation for large language model reasoning. In *International Conference on Learning Representations*, 2026.

Appendix

In Appendix A, we present the full algorithmic description of SAUCE as a single-pass procedure over a multi-agent reasoning trajectory. In Appendix B, we provide our implementation details, including:

- **evaluation metrics** (Appendix B.1),
- **benchmark formats and subset construction** (Appendix B.2),
- and **hyperparameter configuration** (Appendix B.3).

In Appendix C, we present additional empirical studies, including:

- a **comparison with a learned, context-dependent transition coefficient** (Appendix C.1),
- **per-axis hyperparameter sensitivity** (Appendix C.2),
- the **full Brier-Platt calibration grid** (Appendix C.3),
- **trajectory-summary comparisons and temporal-order ablations** (Appendix C.4),
- **bootstrap comparisons and repeated-rollout variability** (Appendix C.5),
- and **trajectory analysis under unanimous final agreement** (Appendix C.6).

Finally, in Appendix D, we provide proofs and derivations supporting our theoretical claims, including:

- the **proof of Proposition 4.1** (Appendix D.1),
- the **innovations form of the marginal likelihood** (Appendix D.2),
- and the **equivalence with the evidence lower bound** (Appendix D.3).

A Algorithm

Algorithm 1 summarizes SAUCE as a single-pass procedure over a parallel-MAS rollout. Given $\tau = (\boldsymbol{z}_t^{1:N})_{t=1}^{T}$, SAUCE traverses the rounds once, extracts the round-level signals (M_t, R_t) defined in Sec. 3.1, and applies the propagation–update recursion of Sec. 3.2. The procedure is training-free, requires no additional forward passes, and operates entirely on quantities available in a standard rollout (parsed answers and top-k token log-probabilities). All hyperparameters are scalars and are shared across (model, dataset) cells within each MAS protocol (Appendix C.2).

Algorithm 1 SAUCE: Sequential Agent Uncertainty through Consensus Evolution.

Require: Rollout $\tau = (\boldsymbol{z}_t^{1:N})_{t=1}^{T}$ for input $\boldsymbol{x}$; hyperparameters $\lambda \in (0,1)$, $q > 0$, $v_0 > 0$, μ_0.
Ensure: Per-record uncertainty score $u(\boldsymbol{x}, \tau)$.

$(\mu, v) \leftarrow (\mu_0, v_0)$
for $t = 1, \dots, T$ **do**
  Extract parsed answers $\{a_t^i\}_{i=1}^{N}$ and per-token log-probabilities from $\{\boldsymbol{z}_t^i\}_{i=1}^{N}$
  $m_t \leftarrow \mathrm{mode}(\{a_t^i\}_{i=1}^{N})$ ▷ round modal answer
  $M_t \leftarrow \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}[a_t^i = m_t]$ ▷ agreement signal (Eqn. 6)
  $R_t \leftarrow \frac{1}{N}\sum_{i=1}^{N} H_{\mathrm{tok}}(\boldsymbol{z}_t^i)$ ▷ generation uncertainty (Eqn. 7)
  $\widetilde{\mu} \leftarrow \lambda\,\mu, \quad \widetilde{v} \leftarrow \lambda^2 v + q$ ▷ predict (Eqn. 8)
  $\alpha \leftarrow \widetilde{v}/(\widetilde{v} + R_t)$ ▷ Kalman gain (Eqn. 9)
  $\mu \leftarrow \widetilde{\mu} + \alpha\,(M_t - \widetilde{\mu}), \quad v \leftarrow (1-\alpha)\,\widetilde{v}$ ▷ update (Eqn. 10)
  $\mu_t \leftarrow \mu$
end for
return $u(\boldsymbol{x}, \tau) = -\dfrac{1}{T}\sum_{t=1}^{T} \mu_t$ ▷ (Eqn. 11)

B Implementation Details

This appendix collects the metric definitions, benchmark formats, and hyperparameter values used in the main results.

B.1 Metrics

Each estimator outputs an *uncertainty score* $s_i = u(\boldsymbol{x}_i, \tau_i)$ per record (higher = less reliable), evaluated against a binary correctness label $y_i \in \{0, 1\}$ on a dataset of size N. Let $\mathcal{P} = \{i : y_i = 1\}$ and $\mathcal{N} = \{i : y_i = 0\}$.

AUROC. The probability that a uniformly drawn correct trajectory has a lower uncertainty score than a uniformly drawn incorrect one, with ties contributing one half:

$$\text{AUROC} = \frac{1}{|\mathcal{P}||\mathcal{N}|} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} \Big(\mathbb{1}[s_i < s_j] + \tfrac{1}{2} \mathbb{1}[s_i = s_j] \Big), \tag{18}$$

equivalent to the Mann–Whitney U statistic, with tied pairs contributing $\frac{1}{2}$.

AUARC. The area under the accuracy–coverage curve obtained by rejecting the most uncertain trajectories first. Let π be the permutation that orders records by ascending uncertainty, $s_{\pi(1)} \leq s_{\pi(2)} \leq \ldots \leq s_{\pi(N)}$. Then

$$\text{AUARC} = \frac{1}{N} \sum_{k=1}^{N} \frac{1}{k} \sum_{i=1}^{k} y_{\pi(i)}, \tag{19}$$

i.e., the average accuracy on the top-k retained subset over all coverage levels k/N.

Brier after Platt scaling. The Brier score evaluated on calibrated probabilities,

$$\text{Brier} = \frac{1}{N} \sum_{i=1}^{N} \big(\widehat{p}_i - y_i \big)^2, \qquad \widehat{p}_i = \sigma\big(a \cdot (-s_i) + b \big), \tag{20}$$

where Platt parameters (a, b) are fit by minimising the binary cross-entropy on training data. We use 5-fold cross-validation: (a, b) is refit on each training fold, the Brier score is computed on the corresponding held-out fold, and we report the average across folds.

All three metrics are reported as percentages ($\times 100$).

B.2 Benchmarks

We follow each benchmark's standard test split or, where listed in Table 4, a fixed random subset shared across all backbones. The same prompt template and decoding configuration are used for every (model, dataset, MAS) cell. The consensus answer of a MAS rollout is judged by an automatic verifier matched to the answer format:

Benchmark Choice. Open-source models in our suite (Qwen3-4B, Llama-3.1-8B-Instruct, Gemma-4-E4B-it) are at the smaller-scale end of LLMs and already exhibit a wide accuracy spread on MATH-500, MMLU-Pro, and BBH, yielding a well-balanced split of correct and incorrect MAS outputs that uncertainty estimation can be meaningfully evaluated against. Closed-source models (Gemini-2.5-Flash-Lite, GPT-5.4-mini) largely saturate the original MATH-500 and BBH, leaving too little correct/incorrect variance for the same purpose. We therefore replace those with the harder variants, e.g., OlympiadBench for mathematics and BBEH-mini for hard instruction-following, to restore an informative spread at this stronger model class. MMLU-Pro spans a wide enough difficulty range to remain useful for both classes; we use the full split for open-source models and the $5{,}000$ subset above for closed-source models to control API cost.

Multiple-Choice (MMLU-Pro, BBH, BBEH-mini): canonicalised exact match on the parsed option after format-aware normalisation (whitespace, parenthesisation, casing).

Mathematical Reasoning (MATH-500, OlympiadBench): symbolic equivalence via the Math-Verify framework, which handles algebraic simplification (e.g. fraction/decimal forms, sign).

Subset Construction. For closed-source models we use a $5{,}000$-question random subset of the MMLU-Pro test split, sampled uniformly from its $12{,}032$ items with a fixed seed and shared across all closed-source backbones and MAS protocols. Drawing a subset (rather than the full split) keeps API inference cost manageable while preserving the broad subject coverage of MMLU-Pro. BBEH-mini is used as released by [27], a compact ≈ 460-question collection spanning 23 Big-Bench Extra Hard tasks; we use it without further filtering.

Table 4: Benchmark formats and evaluation sizes used in the main results. "–" denotes a benchmark not used for that model class.

Dataset	Format	Open-source size	Closed-source size
MATH-500	Free-form math	500	–
MMLU-Pro	Multiple-choice	12,032	5,000 (subset)
BBH	MCQ / short string	1,250	–
OlympiadBench	Free-form math	–	674
BBEH-mini	MCQ / short string	–	$\approx$460

B.3 Hyperparameter Configuration

SAUCE uses a single tuple $(\lambda, q, \sigma_0^2, \mu_0)$ per MAS protocol, transferred across every (model, dataset) cell of Tables 1–2. We tune on a single calibration cell (Qwen3-4B / MMLU-Pro) by joint grid search; per-axis sensitivity around these defaults is reported in Appendix C.2.

Table 5: SAUCE hyperparameter defaults transferred across all cells.

MAS protocol	λ	q	σ_0^2	μ_0
Debate	0.95	0.10	0.25	0.0
DyLAN	0.70	0.001	0.25	0.0

C Additional Experiments

This appendix collects additional empirical studies supporting the main results: a comparison between constant and context-dependent transition coefficients (Appendix C.1), hyperparameter sensitivity (Appendix C.2), the full calibration grid (Appendix C.3), comparisons with simpler trajectory summaries and temporal-order ablations (Appendix C.4), statistical analyses (Appendix C.5), and a diagnostic study of trajectories with identical final-round agreement (Appendix C.6).

C.1 Constant vs. Context-Dependent Transition Coefficient

The main text instantiates the latent belief as

$$s_t = \lambda\, s_{t-1} + \epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0, q), \tag{21}$$

with a constant transition coefficient λ. A natural generalisation is to let λ depend on the interaction history $\boldsymbol{h}_t$, since how strongly an agent should carry its prior belief into the next round may plausibly vary with the conversation so far. This appendix shows the extra capacity is not exercised: under a tight variational objective, a learnable $\lambda_\theta(\boldsymbol{h}_t) = \sigma\big(\mathrm{MLP}(\boldsymbol{h}_{t-1})\big)$ collapses to a near-degenerate constant whose downstream AUROC is close to that of a fixed scalar, so we adopt the constant λ^* as the paper's main configuration.

Training Objective: An ELBO. We learn θ by maximising the evidence lower bound (ELBO) on the marginal likelihood of the observation sequence, an objective that does not use correctness labels. For any variational posterior $q(s_{1:T})$,

$$\log p_\theta(o_{1:T}) = \mathcal{L}(\theta, q) + \mathrm{KL}\big(q(s_{1:T}) \,\|\, p_\theta(s_{1:T} \mid o_{1:T})\big), \tag{22}$$

$$\mathcal{L}(\theta, q) = \mathbb{E}_q[\log p_\theta(o_{1:T}, s_{1:T}) - \log q(s_{1:T})]\,. \tag{23}$$

Setting the variational distribution to the exact smoothing posterior makes the ELBO tight ($\mathrm{KL} = 0$). The resulting log marginal likelihood can be evaluated by the Kalman filter via the innovations decomposition:

$$\mathcal{L}(\theta) = -\tfrac{T}{2}\log(2\pi) - \tfrac{1}{2}\sum_{t=1}^{T}\left[\log S_t + \tfrac{e_t^2}{S_t}\right], \quad e_t = o_t - \lambda_\theta(\boldsymbol{h}_t)\,\mu_{t-1}, \quad S_t = \widetilde{\sigma}_t^2 + R_t. \tag{24}$$

The full derivation, including the equivalence between the ELBO and the innovations form, is given in Appendix D.3. Maximising Eqn. 24 is therefore variational learning of the transition coefficient, with no looser surrogate to blame if the procedure fails to extract context-dependent structure.

Table 6: Learned vs. constant transition coefficient on Qwen3-4B / Debate (AUROC↑, $\times 100$). The two filter variants achieve similar AUROC on both datasets; both improve substantially over the no-filter baseline.

Method	MATH-500	MMLU-Pro
mean(R_t) – no filter	77.00	64.93
Learned $\lambda_\theta(\boldsymbol{h}_t)$	82.85	**75.18**
Const. λ^* (ours)	**82.94**	74.76

Downstream Comparison. We optimise Eqn. 24 end-to-end with respect to θ on Qwen3-4B Debate trajectories from MATH-500 and MMLU-Pro and compare against the constant λ^* on the same trajectories. Table 6 reports downstream AUROC: the learned context-dependent $\lambda_\theta(\boldsymbol{h}_t)$ and constant λ^* differ by only 0.09 and 0.42 AUROC points on MATH-500 and MMLU-Pro, respectively, and both substantially outperform the no-filter baseline. Because the bound is tight, the absence of an improvement cannot be attributed to a loose variational surrogate, and the optimiser nonetheless fails to extract any context-dependent signal that downstream uncertainty estimation can use. Inspection of the trained network confirms this picture: the learned $\lambda_\theta(\boldsymbol{h}_t)$ concentrates on a near-degenerate spike with no measurable conditioning on correctness. We therefore adopt the constant transition coefficient as the paper's main configuration: it achieves similar AUROC to the learned variant while remaining fully training-free, and it side-steps the additional engineering overhead (hidden-state access, optimiser hyperparameters, validation splits) that the learnable alternative drags along.

Implementation Details. The MLP has a 64-dimensional bottleneck and a 32-dimensional ReLU hidden layer; q, μ_0 and σ_0^2 are fixed buffers shared with the constant baseline. We optimise the innovations NLL (the negative of Eqn. 24) with Adam at learning rate 10^{-3} and early-stop on validation AUROC, which uses correctness labels. Trajectories are split $70/15/15$ into train, validation and test; sample sizes are $n = 500$ for MATH-500 and $n = 12{,}032$ for MMLU-Pro.

C.2 Hyperparameter Sensitivity

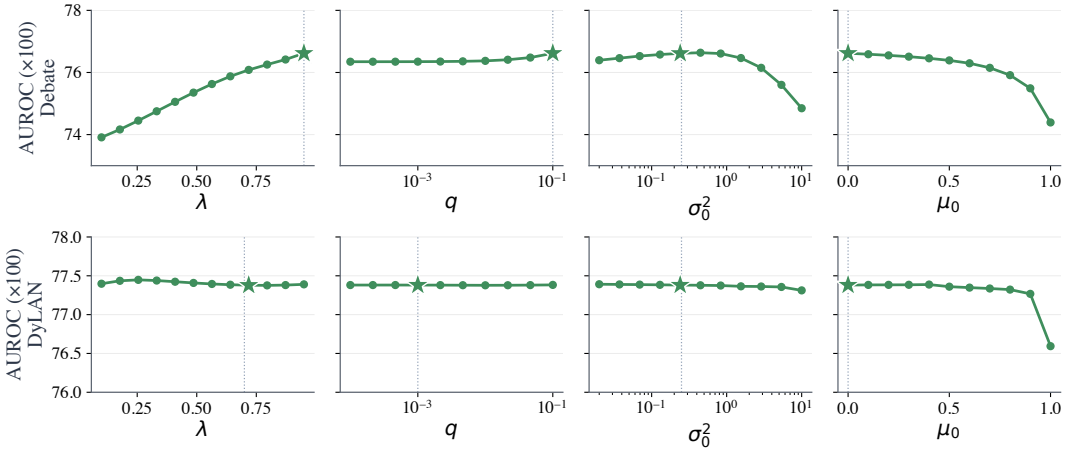


Figure 7: Per-axis sensitivity of SAUCE's four hyperparameters at the calibration cell (Qwen3-4B / MMLU-Pro). Top row: Debate, selected values $(\lambda, q, \sigma_0^2, \mu_0) = (0.95, 0.10, 0.25, 0.0)$. Bottom row: DyLAN, selected values $(0.70, 0.001, 0.25, 0.0)$. In each panel, one hyperparameter is swept while the other three are held at their per-protocol selected value (marked by ★ and a dotted vertical line).

Setup. SAUCE's four scalar hyperparameters $(\lambda, q, \sigma_0^2, \mu_0)$ are tuned by joint grid search on the Qwen3-4B / MMLU-Pro calibration cell, separately for each MAS protocol. The resulting tuples, which are $(0.95, 0.10, 0.25, 0.0)$ for Debate and $(0.70, 0.001, 0.25, 0.0)$ for DyLAN, are then transferred verbatim to every other (model, dataset) cell in Tables 1–2.

Fig. 7 sweeps each axis independently around the selected tuple. In Debate, AUROC drops by roughly three points as λ decreases from 0.95 to 0.1. Decreasing λ erodes the temporal coupling that

Table 7: Calibration quality on open-source models, measured by Brier score after Platt scaling with 5-fold cross-validation (↓, ×100). **Bold** denotes the lowest score per (model, protocol, dataset) cell, and per-backbone Avg.

Method	MATH-500		MMLU-Pro		BBH		Avg
	Debate	DyLAN	Debate	DyLAN	Debate	DyLAN	
Qwen3-4B							
PE	16.92	17.06	22.44	21.64	15.23	16.48	18.30
LL	17.12	16.97	22.46	21.65	15.41	16.55	18.36
UDPO	17.16	15.13	21.40	19.84	15.54	15.85	17.49
MATU	17.86	17.94	22.38	22.34	16.32	16.16	18.83
SAUCE (Ours)	**15.98**	**15.06**	**19.19**	**18.52**	**14.77**	**15.62**	**16.52**
Llama-3.1-8B-Instruct							
PE	24.02	25.26	24.84	24.45	24.59	22.83	24.33
LL	24.71	22.51	24.74	23.65	24.56	22.49	23.78
UDPO	20.70	22.38	22.03	21.67	21.94	22.51	21.87
MATU	25.02	20.76	24.94	23.47	23.68	22.32	23.36
SAUCE (Ours)	**16.57**	**17.71**	**19.76**	**20.18**	**20.67**	**20.73**	**19.27**
Gemma-4-E4B-it							
PE	13.43	12.79	18.12	16.79	11.68	17.23	15.01
LL	13.41	**12.78**	18.14	16.78	11.69	17.21	15.00
UDPO	13.45	14.10	17.72	16.20	10.91	13.05	14.24
MATU	13.52	13.79	19.79	18.77	11.71	14.95	15.42
SAUCE (Ours)	**13.30**	13.79	**15.53**	**15.29**	**10.06**	**12.40**	**13.40**

Table 8: Calibration quality on closed-source models, measured by Brier score after Platt scaling with 5-fold cross-validation (↓, ×100). Notation follows Table 7.

Method	OlympiadBench		MMLU-Pro		BBEH-mini		Avg
	Debate	DyLAN	Debate	DyLAN	Debate	DyLAN	
Gemini-2.5-Flash-Lite							
PE	22.35	21.87	18.70	14.84	17.21	15.05	18.34
LL	22.47	22.18	**14.83**	**13.53**	17.18	15.56	17.63
UDPO	21.60	**16.63**	18.47	15.93	16.80	**14.21**	17.27
MATU	22.02	23.45	18.62	15.93	17.06	14.56	18.61
SAUCE (Ours)	**17.10**	16.71	15.28	15.36	**16.30**	14.55	**15.88**
GPT-5.4-mini							
PE	24.87	20.50	16.41	16.12	15.67	**13.80**	17.90
LL	24.86	20.67	16.48	16.03	15.59	13.83	17.91
UDPO	20.92	17.92	15.69	14.57	**15.30**	13.97	16.40
MATU	25.06	22.96	16.48	16.47	15.63	13.83	18.41
SAUCE (Ours)	**16.38**	**17.04**	**14.58**	**14.18**	15.54	13.98	**15.28**

aggregates evidence across rounds, consistent with the role of sequential dynamics highlighted in Table 3. The curves also show sensitivity to the initial mean and variance over the full swept range; these axes are not uniformly flat.

We transfer the selected tuple to every other (model, dataset) cell instead of tuning each cell separately. The sensitivity study characterises the calibration cell; it does not bound the potential benefit of per-cell tuning on other models or datasets.

C.3 Calibration on All Benchmarks

Table 7 and Table 8 report the full Brier-Platt grid across all backbones, MAS protocols, and benchmarks. SAUCE attains the best per-backbone average on every model and is the leading or near-leading method on the large majority of cells, in line with the headline result of Fig. 5.

Table 9: Simpler trajectory summaries under Debate (AUROC ↑, ×100). **Boldface** and underlining denote the best and second-best performance. Evaluation sizes and score definitions follow Appendix C.4.

Method	Qwen3-4B			GPT-5.4-mini		
	MATH-500	MMLU-Pro	BBH	OlympiadBench	MMLU-Pro	BBEH-mini
EWMA	66.80	68.43	69.56	81.63	66.95	53.46
Slope	37.79	38.76	40.75	32.36	39.56	**57.25**
Switches	61.76	60.59	61.47	71.08	57.58	52.43
TTC	66.47	68.29	68.29	79.97	66.31	56.94
Mean $1-M_t$	66.60	68.56	69.72	82.33	67.08	55.95
Mean H_t	66.78	68.35	69.33	82.37	66.80	56.63
SAUCE (Ours)	**68.59**	**76.42**	**72.86**	**83.83**	**69.53**	49.96
n	500	12,032	1,250	674	4,999	460

C.4 Comparison with Simpler Trajectory Summary Methods

In Sec. 5.5, we compared the full update with static agreement and reliability-weighted aggregation. Here, we extend the analysis to simpler trajectory summaries on Qwen3-4B and GPT-5.4-mini across six Debate settings.

The experiments in this section follow the same protocol-specific hyperparameter settings as the main experiments, as specified in Appendix B.3.

Evaluation Records. For the single-rollout comparisons, MMLU-Pro uses 12,032 Qwen examples and 4,999 evaluable GPT examples from the 5,000-question subset; the remaining evaluation sizes are listed in Table 9.

Alternative Summaries. All scores are oriented so that higher values indicate greater uncertainty:

- **EWMA**: a fixed-weight moving average of agreement, initialized with $h_1 = M_1$ and updated as $h_t = (1-\beta)h_{t-1} + \beta M_t$ with $\beta = 0.3$. Its score is $-T^{-1}\sum_t h_t$, matching SAUCE's trajectory-averaged readout.
- **Slope**: the negative least-squares slope of M_t over rounds.
- **Switches**: the number of adjacent rounds in which the plurality answer changes.
- **TTC** (time to consensus): the first round with unanimous agreement, with trajectories that never reach unanimity assigned $T+1$.
- **Mean disagreement**: $T^{-1}\sum_t(1-M_t)$.
- **Mean answer entropy**: $T^{-1}\sum_t H_t$, where H_t is the entropy of the empirical distribution of parsed answers at round t. This measures cross-agent answer diversity and is distinct from token-level generation entropy R_t.

Results. SAUCE achieves the highest AUROC among the listed summaries in five of the six settings. It also outperforms EWMA in those five settings, indicating that its gains generally extend beyond fixed-weight temporal smoothing. The exception is GPT-5.4-mini on BBEH-mini, where SAUCE is near chance and the slope summary performs best. The paired-bootstrap comparison with EWMA is reported in Appendix C.5.

Temporal-Order Ablation. We keep each trajectory and its final prediction fixed, but process the paired signals (M_t, R_t) in their original order, in reverse order, or after a random permutation. Shuffled AUROC is averaged over three permutation seeds. The evaluation records and filter parameters are unchanged across conditions.

Table 10 shows that the original order outperforms both alternatives in five of the six settings. The clearest differences occur on MMLU-Pro: original processing exceeds the better alternative by 3.97 AUROC points for Qwen3-4B and 3.31 for GPT-5.4-mini. The effect is smaller elsewhere, and reversing the order improves Qwen3-4B on BBH. Thus, chronological order provides useful information in several settings, but its benefit is not uniform across tasks.

Table 10: Temporal-order ablation under Debate (AUROC ↑, ×100). Shuffled averages three seeds; Δ is original minus the better alternative. Best and second-best scores are bold and underlined.

Method	Qwen3-4B			GPT-5.4-mini		
	MATH-500	MMLU-Pro	BBH	OlympiadBench	MMLU-Pro	BBEH-mini
Original	**68.59**	**76.42**	72.86	**83.83**	**69.53**	**49.96**
Reversed	68.30	71.68	**74.26**	82.29	65.21	<u>49.72</u>
Shuffled	<u>68.41</u>	<u>72.45</u>	<u>73.84</u>	<u>82.99</u>	<u>66.22</u>	49.12
Δ	+0.19	+3.97	−1.41	+0.84	+3.31	+0.24

Table 11: Paired-bootstrap comparisons under Debate. The first two rows report SAUCE's AUROC (×100) and 95% interval; subsequent rows give its tail fraction against each baseline. **Bold** indicates $p < 0.05$; < 0.001 denotes zero tail counts in 1,000 resamples.

Statistic	Qwen3-4B			GPT-5.4-mini		
	MATH-500	MMLU-Pro	BBH	OlympiadBench	MMLU-Pro	BBEH-mini
SAUCE	68.59	76.42	72.86	83.83	69.53	49.96
95% CI	62.6–74.7	75.6–77.3	69.4–76.3	80.7–87.1	67.4–71.5	43.8–56.3
vs. PE	**0.011**	**< 0.001**	**0.019**	**< 0.001**	**< 0.001**	0.121
vs. LL	**0.020**	**< 0.001**	**0.034**	**< 0.001**	**< 0.001**	**0.002**
vs. UDPO	**< 0.001**	**< 0.001**	**< 0.001**	**< 0.001**	**< 0.001**	1.000
vs. MATU	**0.016**	**< 0.001**	**< 0.001**	**< 0.001**	**< 0.001**	0.378
vs. EWMA	0.156	**< 0.001**	**< 0.001**	**0.002**	**< 0.001**	0.998

C.5 Statistical Significance

The experiments in this section follow the same protocol-specific hyperparameter settings as the main experiments, as specified in Appendix B.3.

Prompt-Level Bootstrap. We perform 1,000 paired bootstrap resamples of the evaluation records in each setting of Table 9. Each resample selects records with replacement and applies the same indices to all estimators. The 2.5th and 97.5th percentiles give the reported 95% AUROC intervals. For each baseline B, we also report the empirical tail fraction p_B, the proportion of paired resamples in which $\mathrm{AUROC}(\mathrm{SAUCE}) - \mathrm{AUROC}(B) \leq 0$. These comparisons are one-sided and unadjusted for multiple comparisons. A reported value below 0.001 means no nonpositive difference was observed in 1,000 resamples; it does not establish a zero population probability.

In these comparisons, LL averages token log-likelihood over agents and rounds, and MATU uses a single rollout ($N = 1$).

In these supplementary comparisons, the tail fraction is below 0.05 against PE, averaged LL, UDPO, and single-run MATU in five of the six settings. Against EWMA, it is below 0.05 in four settings. Qwen3-4B / MATH-500 has a positive point-estimate difference from EWMA but a tail fraction of 0.156, while GPT-5.4-mini / BBEH-mini does not show an advantage over EWMA. These intervals quantify prompt-sampling uncertainty for a fixed rollout; they do not measure variability across independently generated rollouts.

Repeated-Rollout Variability. We separately summarize independently generated Debate rollouts at temperature 0.7 on four settings, evaluating SAUCE, averaged LL, and final-round PE on each run. Table 12 reports the arithmetic mean and sample standard deviation of AUROC, together with the actual run counts and evaluation sizes. MATH-500, BBH, and BBEH-mini each have five runs; OlympiadBench has six, with 673–674 evaluable records per run. We restrict this summary to settings with comparable evaluation sizes across runs.

SAUCE has the highest mean AUROC among the three estimators in all four settings, with sample standard deviations ranging from 0.87 to 2.29 AUROC points. On BBEH-mini its mean remains near chance, so the relative advantage over LL and PE should not be interpreted as reliable discrimination

Table 12: Repeated Debate rollouts (AUROC ↑, mean ± sample standard deviation, ×100). **Boldface** and underlining denote the best and second-best means.

Method	Qwen3-4B		GPT-5.4-mini	
	MATH-500	BBH	OlympiadBench	BBEH-mini
PE	67.04 ± 1.64	69.83 ± 0.99	53.65 ± 1.92	<u>46.11</u> ± 1.31
LL	<u>67.67</u> ± 1.22	<u>71.57</u> ± 1.59	<u>70.17</u> ± 0.68	43.34 ± 2.54
SAUCE (Ours)	**72.26** ± 2.29	**72.58** ± 1.76	**82.79** ± 0.87	**49.01** ± 1.51
Runs	5	5	6	5
n per run	500	1,250	673–674	460

Table 13: Answer-switch analysis among final-unanimous trajectories on Qwen3-4B / Debate / MMLU-Pro ($n = 10{,}696$). Switches count changes in individual agents' answers.

Switches	n	Accuracy (%)	Mean confidence
0	8,004	76.9	0.915
1	1,595	49.5	0.758
2	486	44.4	0.676
3+	611	38.8	0.622

on that task. The repeated-run summaries and prompt-level bootstrap address different sources of variability and are reported separately.

C.6 Trajectory Analysis under Unanimous Final Agreement

The experiment in this section follows the same protocol-specific hyperparameter settings as the main experiments, as specified in Appendix B.3.

The motivating example in Fig. 1 suggests that identical final agreement can conceal different interaction histories. We examine this directly on Qwen3-4B / Debate / MMLU-Pro by restricting attention to the $10{,}696$ of $12{,}032$ trajectories whose agents give the same parsed answer in the final round. We group these trajectories by the total number of individual agents' answer changes between adjacent rounds, rather than the plurality-answer switches used as a baseline in Appendix C.4.

As the number of agent-level switches increases from zero to three or more, accuracy decreases from 76.9% to 38.8%, while mean SAUCE confidence $\bar{\mu} = -u$ decreases from 0.915 to 0.622. Final-round agreement is constant across these groups, but SAUCE's score tracks their different empirical reliability. This is a descriptive association between interaction history, confidence, and correctness; the analysis does not establish a causal effect of switching, and $\bar{\mu}$ is not a calibrated correctness probability.

D Theoretical Analysis and Proofs

This appendix collects the formal derivations and proofs supporting the linear-Gaussian instantiation of SAUCE in Sec. 4. Appendix D.1 proves Proposition 4.1 (the closed-form posterior update that defines the SAUCE recursion). Appendix D.2 derives the innovations form of the marginal likelihood used as our training objective in Appendix C.1. Appendix D.3 shows that, under our linear-Gaussian model, this objective is also the tight evidence lower bound, which underwrites the variational interpretation of the empirical analysis.

D.1 Proof of Proposition 4.1

Proof. We proceed by induction on t. The base case at $t = 0$ holds by the prior $s_0 \sim \mathcal{N}(\mu_0, \sigma_0^2)$. Assume the inductive hypothesis $s_{t-1} \mid o_{1:t-1} \sim \mathcal{N}(\mu_{t-1}, \sigma_{t-1}^2)$; we show that the same Gaussian form propagates to round t with the stated mean and variance.

Step 1: Predictive Distribution. The transition $s_t = \lambda s_{t-1} + \epsilon_t$ with $\epsilon_t \sim \mathcal{N}(0, q)$ independent of $(s_{t-1}, o_{1:t-1})$ is an affine transformation of a Gaussian plus an independent Gaussian, so $s_t \mid o_{1:t-1}$ is Gaussian with

$$\begin{aligned}\mathbb{E}[s_t \mid o_{1:t-1}] &= \mathbb{E}[\lambda s_{t-1} + \epsilon_t \mid o_{1:t-1}] = \lambda \mu_{t-1} =: \widetilde{\mu}_t, \\ \mathrm{Var}[s_t \mid o_{1:t-1}] &= \lambda^2 \mathrm{Var}[s_{t-1} \mid o_{1:t-1}] + \mathrm{Var}[\epsilon_t] = \lambda^2 \sigma_{t-1}^2 + q =: \widetilde{\sigma}_t^2.\end{aligned}$$

Hence $s_t \mid o_{1:t-1} \sim \mathcal{N}(\widetilde{\mu}_t, \widetilde{\sigma}_t^2)$.

Step 2: Joint Distribution of State and Observation. The observation model $o_t \mid s_t \sim \mathcal{N}(s_t, r_t^2)$ is equivalent to $o_t = s_t + \eta_t$ with $\eta_t \sim \mathcal{N}(0, r_t^2)$ independent of s_t given $o_{1:t-1}$. Conditional on $o_{1:t-1}$, both s_t and o_t are linear functions of jointly Gaussian variables, so the conditional joint $(s_t, o_t) \mid o_{1:t-1}$ is itself Gaussian. Its first two moments are

$$\begin{aligned}\mathbb{E}[o_t \mid o_{1:t-1}] &= \mathbb{E}[s_t + \eta_t \mid o_{1:t-1}] = \widetilde{\mu}_t, \\ \mathrm{Var}[o_t \mid o_{1:t-1}] &= \mathrm{Var}[s_t \mid o_{1:t-1}] + \mathrm{Var}[\eta_t] = \widetilde{\sigma}_t^2 + r_t^2, \\ \mathrm{Cov}(s_t,\, o_t \mid o_{1:t-1}) &= \mathrm{Cov}(s_t,\, s_t + \eta_t \mid o_{1:t-1}) = \mathrm{Var}[s_t \mid o_{1:t-1}] = \widetilde{\sigma}_t^2,\end{aligned}$$

where the last line uses the conditional independence $\mathrm{Cov}(s_t, \eta_t \mid o_{1:t-1}) = 0$. Collecting these moments,

$$\begin{pmatrix} s_t \\ o_t \end{pmatrix} \,\Big|\, o_{1:t-1} \sim \mathcal{N}\Bigg(\begin{pmatrix} \widetilde{\mu}_t \\ \widetilde{\mu}_t \end{pmatrix}, \begin{pmatrix} \widetilde{\sigma}_t^2 & \widetilde{\sigma}_t^2 \\ \widetilde{\sigma}_t^2 & \widetilde{\sigma}_t^2 + r_t^2 \end{pmatrix} \Bigg). \tag{25}$$

Step 3: Conditioning on the New Observation. For any jointly Gaussian pair (X, Y) with means (μ_X, μ_Y), variances $(\Sigma_{XX}, \Sigma_{YY})$, and cross-covariance Σ_{XY}, the standard conditioning identity gives

$$X \mid Y \sim \mathcal{N}\Big(\mu_X + \Sigma_{XY}\Sigma_{YY}^{-1}(Y - \mu_Y),\ \Sigma_{XX} - \Sigma_{XY}\Sigma_{YY}^{-1}\Sigma_{YX}\Big). \tag{26}$$

Applying Eqn. 26 to the joint in Eqn. 25 with $X = s_t$ and $Y = o_t$ yields

$$s_t \mid o_{1:t} \sim \mathcal{N}\Big(\widetilde{\mu}_t + \tfrac{\widetilde{\sigma}_t^2}{\widetilde{\sigma}_t^2 + r_t^2}\big(o_t - \widetilde{\mu}_t\big),\ \widetilde{\sigma}_t^2 - \tfrac{(\widetilde{\sigma}_t^2)^2}{\widetilde{\sigma}_t^2 + r_t^2} \Big).$$

Define the Kalman gain

$$\alpha_t := \tfrac{\widetilde{\sigma}_t^2}{\widetilde{\sigma}_t^2 + r_t^2} \in [0, 1].$$

Substituting back, the posterior mean and variance simplify to

$$\mu_t = \widetilde{\mu}_t + \alpha_t\big(o_t - \widetilde{\mu}_t\big), \qquad \sigma_t^2 = \widetilde{\sigma}_t^2 \cdot \tfrac{r_t^2}{\widetilde{\sigma}_t^2 + r_t^2} = (1 - \alpha_t)\,\widetilde{\sigma}_t^2.$$

The variance further admits a harmonic-of-precisions form: starting from $\sigma_t^2 = \widetilde{\sigma}_t^2 r_t^2 / (\widetilde{\sigma}_t^2 + r_t^2)$,

$$\tfrac{1}{\sigma_t^2} = \tfrac{\widetilde{\sigma}_t^2 + r_t^2}{\widetilde{\sigma}_t^2\, r_t^2} = \tfrac{1}{\widetilde{\sigma}_t^2} + \tfrac{1}{r_t^2},$$

which is exactly the precision-addition statement in the proposition. Since $s_t \mid o_{1:t}$ is Gaussian with the stated mean and variance, the inductive hypothesis carries to round t, completing the induction. □

D.2 Innovations Form of the Marginal Likelihood

By the chain rule of probability, the log marginal likelihood factorises as $\log p(o_{1:T}) = \sum_{t=1}^{T} \log p(o_t \mid o_{1:t-1})$. Under Proposition 4.1 the posterior from the previous round is $p(s_{t-1} \mid o_{1:t-1}) = \mathcal{N}(\mu_{t-1}, \sigma_{t-1}^2)$; marginalising over the linear-Gaussian transition gives the predictive distribution

$$p(s_t \mid o_{1:t-1}) = \mathcal{N}\big(\widehat{\mu}_t,\ \widetilde{\sigma}_t^2\big), \quad \widehat{\mu}_t = \lambda_t(\boldsymbol{h}_t)\,\mu_{t-1}, \quad \widetilde{\sigma}_t^2 = \lambda_t(\boldsymbol{h}_t)^2\,\sigma_{t-1}^2 + q, \tag{27}$$

by the convolution of two Gaussians. Marginalising once more over s_t under the linear-Gaussian observation model yields

$$p(o_t \mid o_{1:t-1}) = \mathcal{N}\big(\widehat{\mu}_t,\ S_t\big), \quad S_t = \widetilde{\sigma}_t^2 + R_t, \tag{28}$$

and substituting the Gaussian log-density and summing over t produces the innovations form

$$\log p(o_{1:T}) = -\tfrac{T}{2}\log(2\pi) - \tfrac{1}{2}\sum_{t=1}^{T}\left[\log S_t + \tfrac{e_t^2}{S_t}\right], \quad e_t = o_t - \widehat{\mu}_t. \tag{29}$$

In the negative log marginal likelihood, e_t^2/S_t penalises prediction errors relative to the predictive variance, while $\log S_t$ discourages unnecessary variance inflation.

D.3 Equivalence with the Evidence Lower Bound

For the following decomposition, we assume a Markov-structured variational posterior,

$$q(s_{1:T}) = q(s_1)\prod_{t=2}^{T} q(s_t \mid s_{t-1}),$$

where dependence on the observed sequence $o_{1:T}$ is left implicit. For any variational posterior $q(s_{1:T})$ over the latent trajectory, the standard decomposition gives

$$\log p(o_{1:T}) = \mathcal{L}(q) + \mathrm{KL}\big(q(s_{1:T}) \,\|\, p(s_{1:T} \mid o_{1:T})\big), \tag{30}$$

$$\mathcal{L}(q) = \mathbb{E}_q[\log p(o_{1:T}, s_{1:T}) - \log q(s_{1:T})]\,. \tag{31}$$

Using the Markov structure $p(o_{1:T}, s_{1:T}) = p(s_1)\prod_{t=2}^{T} p(s_t \mid s_{t-1})\prod_{t=1}^{T} p(o_t \mid s_t)$, the ELBO splits into three interpretable terms,

$$\begin{aligned}\mathcal{L}(q) \;=\; &\underbrace{\sum_{t=1}^{T}\mathbb{E}_{q(s_t)}[\log p(o_t \mid s_t)]}_{\text{(I) observation fit}} \;-\; \underbrace{\mathrm{KL}\big(q(s_1) \,\|\, p(s_1)\big)}_{\text{(II) prior regularisation}} \\ &- \underbrace{\sum_{t=2}^{T}\mathbb{E}_{q(s_{t-1})}\big[\mathrm{KL}\big(q(s_t \mid s_{t-1}) \,\|\, p(s_t \mid s_{t-1})\big)\big]}_{\text{(III) transition regularisation}},\end{aligned} \tag{32}$$

each of which has a closed form under Gaussian q (omitted for brevity).

Tightness then follows from a standard property of linear-Gaussian state-space models: the Kalman smoother yields the exact posterior $p(s_{1:T} \mid o_{1:T})$, so setting $q(s_{1:T}) = p(s_{1:T} \mid o_{1:T})$ makes $\mathrm{KL}(q\|p) = 0$ and the ELBO coincides with the exact log marginal likelihood. Evaluating the three terms above at the Kalman posterior and simplifying recovers the innovations form of Appendix D.2. The training objective in Appendix C.1 is therefore simultaneously the exact marginal likelihood and the tight ELBO, a property we use implicitly when arguing that the failure of $\lambda_\theta(\boldsymbol{h}_t)$ to outperform a constant cannot be attributed to a loose variational bound.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction summarize the paper's scope, method (SAUCE), and main empirical claims, all supported by the theory in Sec. 4 and the experiments in Sec. 5.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The Limitations paragraph at the end of the main paper discusses scope (parallel reasoning MAS only) and the simplicity of our agreement and uncertainty signals.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Proposition 4.1 states the linear-Gaussian assumptions in its statement; the full proof is given in Appendix D.1, with related derivations (innovations form, ELBO equivalence) in Appendix D.2 and Appendix D.3.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Algorithm 1 gives the full SAUCE procedure; Sec. 5.1 and Appendix B list the models, MAS protocols, benchmarks (with sizes, formats, and verifiers), evaluation metrics, and the per-protocol hyperparameter values used in every cell.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code and rollout data are publicly available via our GitHub repository and Hugging Face dataset. The repository includes scripts and instructions to reproduce the main experiments. The paper also provides the full algorithm (Algorithm 1) and hyperparameter settings (Appendix B.3).

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Decoding settings, MAS architectures, and benchmark splits are stated in Sec. 5.1; Appendix B.3 lists the per-protocol SAUCE hyperparameter tuple and how it was selected (joint grid search on a single calibration cell, then transferred).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Appendix C.5 reports paired-bootstrap with 95% AUROC confidence intervals, along with the mean and sample standard deviation across independently generated rollouts. These analyses distinguish prompt-sampling uncertainty from rollout-to-rollout variability.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: We do not report compute details. The experiments require generating MAS rollouts (open-source models served via vLLM, closed-source models via API), but the rollout cost is determined by the underlying LLMs and is not part of SAUCE's contribution.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper studies uncertainty estimation methodology and conforms with the NeurIPS Code of Ethics; no human subjects, deployment, or sensitive data is involved beyond standard public reasoning benchmarks.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: SAUCE is a methodological tool for uncertainty estimation in multi-agent reasoning systems with no direct societal impact tied to a specific application; better-calibrated uncertainty can support more reliable downstream MAS deployment but introduces no new direct risks.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release any new models, datasets, or other assets with misuse risk; it proposes an analytical method evaluated on existing public benchmarks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All datasets (MATH-500, MMLU-Pro, BBH, OlympiadBench, BBEH-mini), models (Qwen3-4B, Llama-3.1-8B, Gemma-4-E4B, Gemini-2.5-Flash-Lite, GPT-5.4-mini),

and serving infrastructure (vLLM) are credited via citations in Sec. 5.1; we use them under their public licenses and within their stated terms of use.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The accompanying code and experimental rollout data are documented in the linked repositories to support reproducibility.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as an important, original, or non-standard component.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.