Uncertainty Propagation on LLM Agent

Anonymous ACL submission

Abstract

Large language models (LLMs) integrated into multi-step agent systems enable complex decision-making processes across various applications. However, their outputs often lack reliability, making uncertainty estimation crucial. Existing uncertainty estimation methods primarily focus on final-step outputs, which fail to account for cumulative uncertainty over the multi-step decision-making process and the dynamic interactions between agents and their environments. To address these limitations, we propose SAUP (Situation Awareness Uncertainty Propagation), a novel framework that propagates uncertainty through each step of an LLM-based agent's reasoning process. SAUP incorporates situational awareness by assigning situational weights to each step's uncertainty during the propagation. Our method, compatible with various one-step uncertainty estimation techniques, provides a comprehensive and accurate uncertainty measure. Extensive experiments on benchmark datasets demonstrate that SAUP significantly outperforms existing state-of-the-art methods, achieving up to 20% improvement in AUROC.

1 Introduction

Large language models (LLMs) (Minaee et al., 2024) have demonstrated remarkable capabilities, and when integrated into agent systems (Wang et al., 2024), they enable complex decision-making processes and broader applications. However, while LLM-based agents are increasingly effective, their outputs are not always reliable, which can lead to significant issues, particularly in high-stakes environments such as healthcare or autonomous systems. This makes uncertainty estimation critical, as it evaluates the reliability of an agent's decisions and outputs (Chang et al., 2024; Raiaan et al., 2024). Understanding and quantifying uncertainty is essential because it offers insight into potential system failures, providing a safeguard for sensitive applications. Current methods for estimating uncertainty in LLM-based agents remain limited. For example, UALA (Han et al., 2024) proposes a one-step uncertainty measurement to estimate the uncertainty of the final step before the agent provides an answer.

A key challenge is that uncertainty accumulates over time in multi-step processes, rather than in isolated actions, and is further exacerbated in dynamic environments where external factors are uncontrollable. These interactions can significantly impact the system's overall uncertainty. Therefore, robust methods that account for various information sources and interaction complexities are necessary to accurately capture the uncertainty across an agent's entire decision-making process. As illustrated in Figure 1, in sensitive contexts, solely observing the final step's uncertainty may lead to overconfidence in the outcome, resulting in adverse consequences and highlighting the importance of considering intermediate uncertainties and the quality of interaction between the agent and its environment.

To estimate LLM uncertainty, previous approaches focus mainly on the variance of the final step's output at the token, sentence, or semantic level. Predictive entropy (Gal and Ghahramani, 2016; Gal et al., 2017), initially used in image data, was extended to language models to predict uncertainty in output tokens (Xiao and Wang, 2021). While likelihood can also indicate uncertainty, (Malinin and Gales, 2020) introduces Normalized Entropy, accounting for output length. (Kuhn et al., 2023) proposes semantic entropy, incorporating linguistic invariances within shared meanings. (Kadavath et al., 2022; Yin et al., 2023) explore self-assessment by LLMs to estimate uncertainty. However, these methods, designed for traditional one-step QA, do not directly apply to LLM agents. *They face two key issues: first, they only consider the final step's uncertainty, ignoring the accumula-*

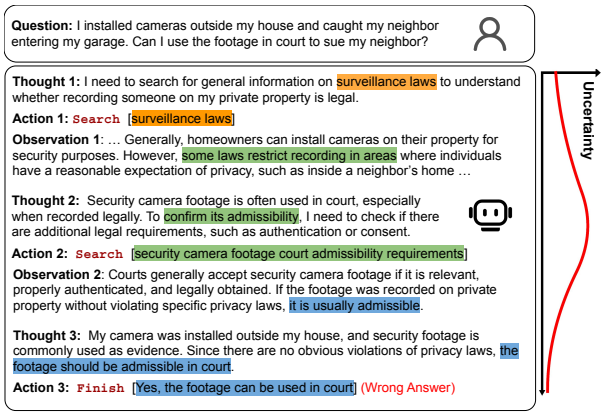

Figure 1: The overall uncertainty of an agent based on large language models (LLMs). In this example, although the large model ultimately arrived at the exact answer, the high uncertainty during the intermediate reasoning process caused it to fail to consider critical legal factors such as privacy laws and rules on admissibility of evidence, leading to an incorrect result.

tion of uncertainty throughout the process; second, they overlook the reasoning process of LLM agents, which is critical in multi-step decision-making and the agent's interaction with its environment.

To address the challenges of uncertainty in multi-step processes within complex environments, we introduce **SAUP** (Situation-Awareness Uncertainty Propagation). SAUP comprehensively estimates uncertainty in LLM-based agents by propagating uncertainty through the multi-step reasoning and decision-making process. It builds upon frameworks like ReACT (Yao et al., 2022), which integrates LLMs' reasoning into problem-solving by decomposing tasks into thinking, acting, and observing steps. SAUP propagates uncertainties from the initial stages to the final step and aggregates them using a situation-weighting scheme, where each step's uncertainty is weighted based on the agent's situation, progress, and observation quality. Since directly measuring an agent's situation is challenging, we design effective surrogates that are adaptable to various scenarios.

The primary contribution of this paper can be summarized as follows: Firstly, We propose SAUP, a simple yet effective pipeline for providing comprehensive situation-aware uncertainty estimation in multi-step agents within complex environments. Unlike existing single-step uncertainty estimation methods, SAUP accounts for the agent's situational context throughout problem-solving, rather than focusing solely on the final step. Secondly, To estimate the agent's unobservable situation, we introduce surrogate methods, which excel in estimating situational uncertainty and offer potential applications in related fields. Lastly, We evaluate SAUP on benchmark datasets such as HotpotQA (Yang et al., 2018), StrategyQA (Geva et al., 2021), and MMLU (Hendrycks et al., 2020). SAUP outperforms state-of-the-art methods, achieving up to a 20% improvement in AUROC, demonstrating its effectiveness.

2 Related Works

2.1 LLM-based Agent

The reasoning capabilities of LLMs have prompted researchers to explore their use as the core of agent reasoning. Nakano et al. (Nakano et al., 2021) made an early attempt to employ LLMs as agents with web search and information retrieval capabilities, transitioning LLMs from passive tools to proactive agents interacting with complex environments. Subsequent works (Wang et al., 2021; Chen et al., 2021) explored LLMs in code generation for software development. Yao et al. (Yao et al., 2022) introduced the ReAct pipeline, utilizing LLMs for decision-making where agents retrieve external information before making decisions. This framework, mirroring human decision-making, became foundational for decision-making agents, inspiring improvements by Shinn et al. (Shinn et al., 2023) and Renze et al. (Renze and Guven, 2024) through self-reflection. Li et al. (Li et al., 2023) proposed CAMEL, which expanded the framework to enable communication between agents, fostering collaboration. Similarly, AutoGen (Wu et al., 2023) allows agents to converse and collaborate with customizable interactions in natural language and code. To further enhance decision-making, Qiao et al. (Qiao et al., 2023) incorporated tool-based monitoring to refine agent behaviors.

2.2 Uncertainty in Large Language Models

LLMs dominate numerous fields, including as agents (Zhao et al., 2023; Xi et al., 2023), but targeted uncertainty estimation methods for LLM-based agents remain unexplored. Existing techniques focus on one-step output uncertainty, originating from traditional language models, such as methods to improve model calibration (Xiao and Wang, 2019, 2021; Jiang et al., 2021). Token-level uncertainty estimation in "white-box" LLMs (Malinin and Gales, 2020; Fomicheva et al., 2020; Darrin et al., 2022; Duan et al., 2024) has advanced, with Kuhn et al. (Kuhn et al., 2023) introducing

semantic equivalence into these calculations. Additionally, self-estimation of uncertainty in both "white-box" and "black-box" LLMs, accessed via APIs, has been explored (Kadavath et al., 2022; Yin et al., 2023; Chen et al., 2024). These methods focus on one-step uncertainty estimation, which can be integrated into the SAUP framework as the backbone for uncertainty assessments.

3 SAUP: Situational Awareness Uncertainty Propagation

We propose our pipeline, SAUP, with the goal of accurately estimating the overall agent's uncertainty by comprehensively considering the uncertainty at each step and the corresponding situational weights, as described in Figure 3. In the following sections, we delve into the details, elucidating how we aggregate the uncertainty from each step and estimate the corresponding situational weights.

3.1 Weighted Uncertainty Propagation

Uncertainty Propagation. As depicted in Figure 3, for each step i, the agent provides the thinking/action with the corresponding uncertainty U_i based on the previous state Z_{i-1} and the question Q. Considering only the uncertainty of the last step as the overall uncertainty U_{agent} is unreasonable and not comprehensive. Instead, we should comprehensively consider and propagate the uncertainties of all steps. The simplest example is using an arithmetic mean of the uncertainty across the steps before the agent gives the final answer. For robustness against outliers, accurate reflection of central tendency, and consistency in proportional changes, the geometric mean or Root Mean Square (RMS) can be a better choice compared to the arithmetic mean.

Situational Uncertainty Weights. Based on the intuitive logic of information flow and experimental observations, we have identified that the contribution of uncertainty at different steps to the overall agent uncertainty is not uniform. Therefore, in addition to the uniform aggregation scheme introduced earlier, it is essential to design a more comprehensive weighting aggregation scheme for overall uncertainty, tailored to the characteristics of the agent.

During the process of obtaining the final answer, the LLM-based agent produces uncertainty. We refer to the contribution of the current step's uncertainty to the overall uncertainty, due to the agent's situation, as the situational weights. Situational weights are determined by factors, such as deviations from the appropriate logical path and the quality of interactions between the agent and the environment, which influence the correctness of the final answer. These situational weights are variable during the agent's problem-solving process and its interaction with the environment. Assume that the uncertainty at step i is U_i and the corresponding situational weight is W_i, the formula of weighted uncertainty propagation is:

$$U_{\text{agent}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}((W_i U_i)^2)} \quad (1)$$

Here we choose the RMS as the propagation method. In the practical application of SAUP, besides the above linear term, we also utilize an extra logical term for numerical stability. We designed the SAUP formula based on the following considerations. First, SAUP relies on a comprehensive consideration of all steps of the agent based on propagation. Second, by introducing situational weights for the uncertainty of different steps, SAUP allows for a more complete assessment of the impact of specific steps on the overall uncertainty of the agent. In the following section 3.2 and 3.3, we will introduce the method for calculating the uncertainty U_i and the situational weight W_i corresponding to each step.

3.2 Single-Step Uncertainty Estimation

From equation 1, we can see that essentially, our SAUP is compatible with all single-step uncertainty estimation methods applicable to various scenarios, including but not limited to the ones we mentioned. SAUP is built upon these one-step methods.

In the practical implementation, we utilize the normalized entropy (Malinin and Gales, 2020), with some modifications to adapt it to the characteristics of the React Agent pipeline. This choice is based on the consideration that normalized entropy has broad applicability. It can not only be applied to open-source LLMs, such as LLAMA, where complete logits of the output are accessible, but can also be utilized with LLMs that are accessible only via API, such as the CHATGPT series. In addition, it is computationally efficient and demonstrates strong predictive performance for single-step uncertainty estimation.

For step n and question Q, we denote the agent's thinking as T_n, and the corresponding action as

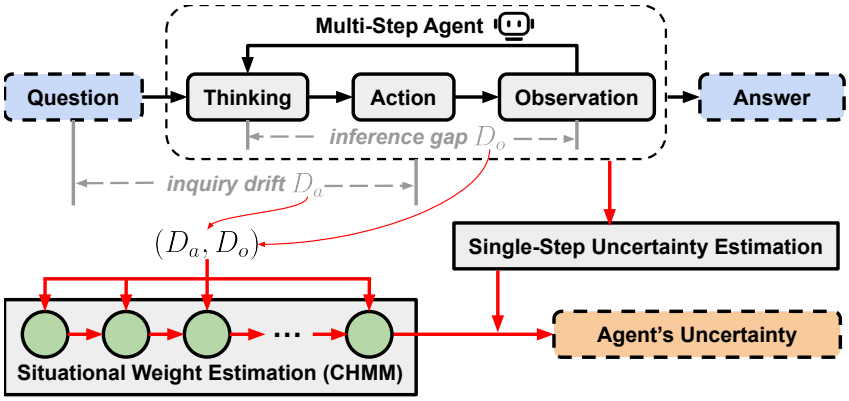

Figure 2: Overview of our proposed SAUP. The general pipeline of LLM-based multi-step agents is represented by **black** arrows, which typically involves three behaviors: thinking, action, and observation. The process of uncertainty propagation is represented by **red** arrows. For each step in the agent, off-the-shelf methods are used to estimate the uncertainty at that step, while also generating a global inquiry drift D_a and a local inference gap D_o. Together, these two components form continuous states that feed into CHMM for situational weight prediction. They, along with single-step uncertainty, jointly derive the agent's overall uncertainty.

A_n. The observation O_n is the information gained from the environment through the action A_n. Let the LLM be denoted as L_θ, and the trajectory of the previous $n-1$ steps as Z_{n-1}, where $Z_{n-1} = \{(A_1, T_1, O_1), \ldots, (A_{n-1}, T_{n-1}, O_{n-1})\}$. The LLM will output the response of thinking T_n and the action A_n together as:

$$(T_n, A_n) = L_\theta(Q, Z_{n-1}). \tag{2}$$

the step uncertainty $U_n = U_n^T + U_n^A$ is designed as the combination of thinking uncertainty U_n^T and action uncertainty U_n^A. And here we consider Predictive entropy (Kadavath et al., 2022) with the length normalization to estimate the thinking uncertainty and action uncertainty as follows:

$$\begin{aligned} U_n^R &= H(R_n \mid Q, Z_{n-1}) \\ &= \mathbb{E}_{R_n}[\frac{1}{|R_n|} - \log \mathbb{P}(R_n \mid Q, Z_{n-1})] \\ &= \mathbb{E}_{R_n}\left[\frac{1}{|R_n|}\sum_{a_i}^{R_n} - \log \mathbb{P}\left(a_i \mid Q, Z_{n-1}, a_0, \cdots a_{i-1}\right)\right] \end{aligned} \tag{3}$$

where U_n^R is the LLM response uncertainty (either U_n^T or U_n^A), a_i is the token of R_n, R_n is the LLM response (either T_n or A_n), $H(\cdot)$ is the entropy function, and $\mathbb{P}$ represent the probability.

3.3 Agent Uncertainty Estimation

Assigning weights W_i to each step's uncertainty U_i in a multi-step reasoning process is crucial for accurate overall uncertainty estimation. In LLM-based agents, effective reasoning significantly influences decision-making. However, these agents may exhibit overconfidence, making it essential to evaluate their situational state properly. Since the situational state is not directly observable, surrogate measures are used to approximate it. One idea is to assign greater weight to steps closer to the final answer or to measure deviation from an ideal trajectory. While these approaches have merit, they do not fully capture the agent's true situational state.

To address this limitation, we introduce learning-based surrogate models to capture agents' latent situational dynamics. Our proposed SAUP-HMMD leverages Hidden Markov Models (HMMs) (Baum and Petrie, 1966) to learn stepwise transitions and adaptively weight observations. HMMs are particularly suitable due to their (1) minimal data requirements and (2) computational efficiency, making them ideal under data constraints. While LSTMs (Hochreiter, 1997) and Transformers (Vaswani, 2017) excel at modeling temporal dependencies, they require significantly more data and computation. Although our framework supports various sequence models, HMMs serve as the baseline for small datasets. Section 4.3 provides a detailed performance comparison.

As a probabilistic framework for modeling sequential patterns, HMMs operate by estimating latent system states through observable evidence, governed by regular state transition dynamics. Formally, a HMM is parameterized by following core components: (i) the hidden state space $\mathcal{S} = \{s_1, ..., s_N\}$ and observation space $\mathcal{O} = \{o_1, ..., o_M\}$, (ii) transition matrix $A = [a_{ij}]$ encoding $P(s_j|s_i)$, (iii) emission matrix $B = [b_{ij}]$ characterizing $P(o_k|s_j)$, and (iv) prior distribution $\pi = [\pi_i]$ specifying initial state probabilities.

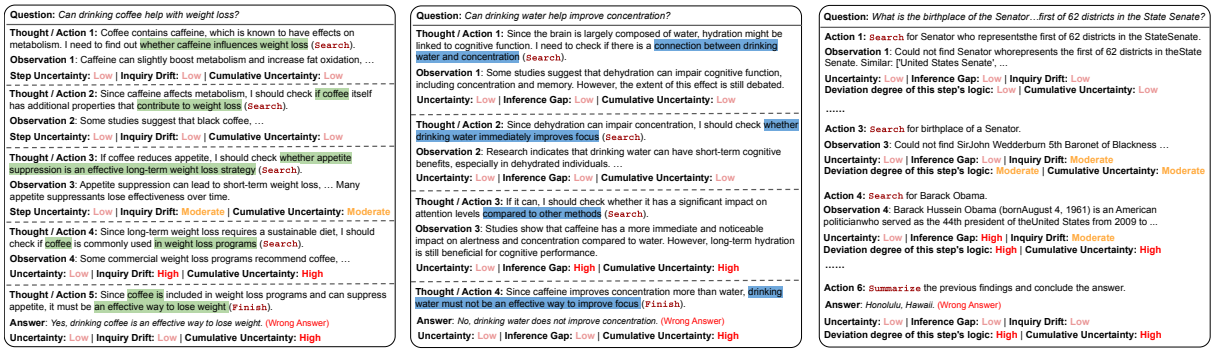

Figure 3: Examples for (a) inquiry drift, (b) inference gap and (c) the hidden states of CHMM. (a) and (b) respectively illustrate that even when the single-step uncertainty remains low throughout the process, excessively high inquiry drift and inference gap in certain steps provide additional situational information that increases the uncertainty of those steps. This, in turn, causes the LLM's reasoning process to deviate from the correct direction, ultimately leading to an incorrect result. (c) illustrates how the hidden states of CHMM, influenced by both inquiry drift and inference gap, affect the agent's logical behavior.

Extending HMMs to continuous observation domains, Continuous HMMs (CHMMs) leverage Gaussian Mixture Models (GMMs) to model emission probabilities within continuous feature spaces. We adopt CHMM as the backbone model for HMMD, where the hidden states are designed to assess the alignment between the large language model's chain-of-thought reasoning and the original problem context. Specifically, we define three discrete hidden states: correct trajectory, moderately deviated trajectory, and highly deviated trajectory, capturing varying degrees of deviation in the reasoning process. To systematically quantify these deviations, we introduce two key distance metrics:

- **Inquiry Drift** D_a : Measures the semantic shift between the original question and the agent's evolving thought-action-observation trajectory, capturing the global deviation across multiple reasoning steps.
- **Inference Gap** D_o : Captures the local transition dynamics by quantifying the discrepancy between a thought and observation process within one step.

These two metrics comprehensively characterize both long-term and short-term variations in the agent's decision trajectory, offering a structured perspective on its situational awareness. To utilize D_a and D_o, we adopt two approaches: (1) directly computing plain distances using a RoBERTa (Liu, 2019) model fine-tuned on SQuAD v2 (Rajpurkar et al., 2018), and (2) integrating the (D_a, D_o) pairs as observable variables into the CHMM framework, leveraging its latent structure to infer the hidden states, which in turn serve as situational weights. This dual approach enhances the robustness of our distance estimation by combining direct metric-based evaluation with probabilistic inference. The CHMM is trained via the Baum-Welch algorithm, enabling a more structured assessment of reasoning deviations.

The SAUP algorithm employs different surrogate configurations. We illustrate the SAUP using distance as the surrogate in Algorithm 1. Initially, uncertainty U_n is computed for step n, along with the corresponding distances D_{a_n} and D_{o_n}. This is repeated for N steps. Subsequently, based on the surrogate choice, either plain or HMM-based, the situational weights W_n are determined. Finally, the uncertainties U and weights W are aggregated to estimate the agent's overall uncertainty U_{agent}.

4 Experiments

In this section, we evaluate the performance of SAUP, aiming to answer the following questions: **Q1**: Does SAUP outperform previous state-of-the-art approaches for uncertainty estimation? **Q2**: Given the comprehensive process of Uncertainty Propagation, does SAUP provide more accurate uncertainty estimation compared to single-step methods? **Q3**: Are the situational weights for specific steps effective in improving overall uncertainty estimation? Since obtaining precise situational weights is impractical, we designed surrogates, including distance-based and position-based methods. Are these surrogates reliable for accurately assessing the agent's current situation?

Algorithm 1 Situational Awareness Uncertainty Propagation (SAUP)

Initialize the N-Step LLM-based Agent L_θ with the problem Q, and the $Z_n = \{(A_1, T_1, O_1), (A_2, T_2, O_2), \ldots, (A_n, T_n, O_n)\}$.

for step n in the problem solving process **do**
  The Uncertainty for current step $U_n^R \leftarrow H(R_n \mid Q, Z_{n-1})$
  $D_{a_n} \leftarrow Dis(Z_n, Q)$
  $D_{o_n} \leftarrow Dis(A_n, O_n)$
  if using the *CHMM* as the surrogates **then**
    Add the $(D_{a_n} + D_{o_n})$ into the D_L
  else
    Using the *Plain-Distance* as the surrogates $W_n \leftarrow D_{a_n} + D_{o_n}$
  end if
end for
if using the *CHMM* as the surrogates **then**
  $(W_1, W_2, \ldots, W_N) \leftarrow H(D_L) = H((D_{a_1} + D_{o_1}), \ldots, (D_{a_N} + D_{o_N}))$
end if
The Uncertainty for the agent $U_{agent} \leftarrow SAUP((U_1, W_1), (U_2, W_2), \ldots, (U_N, W_N))$
return Situational Awareness Agent Uncertainty U_{agent}

4.1 Experimental Setup

LLM-based Agent Framework. Our experiments focus on evaluating SAUP's ability to improve uncertainty estimation for multi-step LLM-based agents. While various multi-step agents follow different pipeline designs, they generally adhere to the thinking-acting-observation workflow. We chose the React (Yao et al., 2022) framework, a widely-used agent model, for its alignment with this workflow.

Backbone LLMs. We selected two categories of LLMs for the React agents: the open-source LLAMA3 (Dubey et al., 2024) series (8B and 70B models) with entropy access, and GPT-4o (Achiam et al., 2023) (available via API), which restricts internal information. This selection ensures broad coverage of real-world scenarios.

Dataset and Task. We evaluated three challenging agent-based QA tasks. The first, **HotpotQA** (Yang et al., 2018), focuses on multi-hop QA with diverse free-form answers. We randomly sampled 2,000 questions from the development set, assessed by both human evaluators and ChatGPT. The second, **MMLU** (Hendrycks et al., 2020), involves multiple-choice questions across diverse fields like law and mathematics. Ten questions were sampled per subtask from the test set. Lastly, **StrategyQA** (Geva et al., 2021) requires implicit reasoning, evaluated with true/false questions from its development set (229 questions).

Environment for External Information. LLM-based agents often need external sources to solve these tasks. For HotpotQA and StrategyQA, we provided access to the Wikipedia API, which retrieves relevant entity-based information. For MMLU, we used SerpAPI (SerpAPI, 2024) for structured Google search results.

Baselines. We evaluated SAUP against several uncertainty estimation methods. For entropy-based approaches, we used predictive and semantic entropy (Xiao and Wang, 2019; Kuhn et al., 2023). Likelihood-based methods (Malinin and Gales, 2020) included plain likelihood and normalized entropy, the latter accounting for token length. We also implemented P(True) (Kadavath et al., 2022; Yin et al., 2023), which prompts agents to self-assess their confidence.

Evaluation Metrics. We used AUROC (Bradley, 1997) to measure the ability of uncertainty methods to distinguish between correct and incorrect responses. Higher AUROC values indicate better differentiation, with a perfect score of 1 representing complete distinction and 0.5 representing random chance.

4.2 Superior Discriminative Performance of SAUP

In this section, we compare various uncertainty measurement methods in assessing whether an LLM-based agent's final response to QA questions is correct or incorrect. The evaluation involves the following steps: (1) The agent, using the ReACT framework, answers the QA questions; (2) Multiple versions of our SAUP method and other baseline uncertainty estimation methods calculate uncertainty scores for each response; (3) Responses are classified as correct (0) or incorrect (1); (4) AUROC is calculated based on classification accuracy and uncertainty scores. Ideally, incorrect answers should correspond to higher uncertainty scores.

We employed several state-of-the-art LLMs, including *{LLAMA3 8B, LLAMA3 70B, GPT4O}*, and conducted evaluations on challenging datasets, namely *{StrategyQA, MMLU, HotpotQA}*. Table 1 presents the results, demonstrating that our

Table 1: Results for SAUP. The best results and second best results are **bold** and underlined, respectively.

Method	HotpotQA			MMLU			StrategyQA		
	LLAMA3 8B	LLAMA3 70B	GPT4O	LLAMA3 8B	LLAMA3 70B	GPT4-O	LLAMA3 8B	LLAMA3 70B	GPT4-O
Predictive Entropy	0.631	0.617	N.A.	0.531	0.585	N.A.	0.542	0.589	N.A.
Likelihood	0.653	0.622	0.764	0.550	0.592	0.610	0.525	0.591	0.641
Normalised Entropy	0.664	0.635	0.772	0.555	0.579	0.607	0.554	0.557	0.710
P(True)	0.601	0.618	0.749	0.528	0.560	0.588	0.533	0.577	0.689
Semantic Entropy	0.702	0.669	N.A.	0.548	0.605	N.A.	0.599	0.610	N.A.
SAUP-Learned	**0.771**	**0.755**	**0.778**	**0.669**	**0.638**	**0.626**	**0.787**	**0.783**	**0.809**

Table 2: Results for SAUP with various Surrogates. The best results and second best results are **bold** and underlined, respectively.

Method	HotpotQA			MMLU			StrategyQA		
	LLAMA3 8B	LLAMA3 70B	GPT4O	LLAMA3 8B	LLAMA3 70B	GPT4-O	LLAMA3 8B	LLAMA3 70B	GPT4-O
SAUP-P	0.723	0.739	**0.797**	0.634	0.636	0.614	0.668	0.641	0.734
SAUP-D	0.762	0.726	0.773	0.660	0.619	0.624	0.755	**0.809**	0.806
SAUP-PD	0.759	0.745	0.782	0.651	0.625	0.619	0.732	0.756	0.785
SAUP-HMMD(Learned)	**0.771**	**0.755**	0.778	**0.669**	**0.638**	**0.626**	**0.787**	0.783	**0.809**

SAUP method, consistently achieves higher AUROC scores across all datasets compared to state-of-the-art methods. These findings indicate that SAUP offers superior performance in distinguishing between correct and incorrect agent responses based on uncertainty estimation, leading to important conclusions.

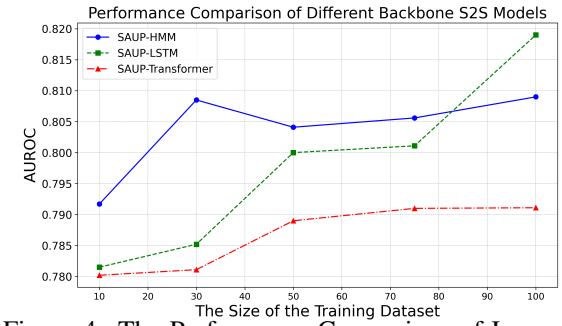

Figure 4: The Performance Comparison of Learned-based Surrogates with Various S2S Backbone Models

4.3 In-Depth Dissection of SAUP

Given the superiority of our proposed *SAUP*, we further dissect its performance by addressing the following questions. This analysis highlights the advantages of SAUP in various aspects and offers insights into its applicability and performance under different conditions.

Q1: Is the uncertainty measurement of the internal steps beneficial for the overall uncertainty measurement of the agent?

Yes, measuring uncertainty at each internal step significantly contributes to a more accurate overall uncertainty estimation. By considering intermediate uncertainties, we capture the cumulative effect of uncertainty propagation throughout the interaction process. As shown in Table 1, SAUP-based methods consistently outperform traditional single-step methods in AUROC scores across datasets and models. The internal step uncertainties provide meaningful information that, when aggregated, enhance the overall uncertainty measurement. Even basic uncertainty propagation methods, such as algorithmic averaging or root mean square (RMS), used to aggregate the uncertainty across all steps, have demonstrated significant improvements over single-step baselines, as shown in Table 3.

Q2: What is the quality of the surrogates, and how do they benefit the overall uncertainty measurement?

High-quality surrogates ensure that situational weights accurately reflect each step's impact on the overall uncertainty. We propose the Position Surrogate (SAUP-P), which assigns greater weight to steps closer to the final answer, and the Plain Distance Surrogate (SAUP-D), which uses only the plain distance. The Hybrid Surrogate (SAUP-PD) combines both approaches with a factor for better balance. As shown in Table 2 and Table 3, different surrogates improve AUROC scores compared to simple uncertainty propagation baselines, which assign equal weights to all steps. In addition, the HMMD-based (learned) surrogate outperforms others by a clear margin, validating its effectiveness in capturing the agent's situational context.

Q3: Can SAUP demonstrate its superiority in separating correct and incorrect results?

Yes, SAUP provides more discriminative uncertainty scores, leading to higher AUROC values across datasets and models, as evidenced in Table 1. The step-by-step propagation of uncertainty allows SAUP to capture the accumulation of uncertainty throughout the reasoning process, enabling better separation of correct and incorrect results.

In addition, we performed a visualization analysis on the StrategyQA dataset (Figure 5). The

Table 3: Results for Simple Uncertainty Propagation. The best results and second best results are **bold** and underlined, respectively.

Method	HotpotQA			MMLU			StrategyQA		
	LLAMA3 8B	LLAMA3 70B	GPT4O	LLAMA3 8B	LLAMA3 70B	GPT4-O	LLAMA3 8B	LLAMA3 70B	GPT4-O
Arithmetic Mean	0.695	0.676	0.781	0.621	0.596	0.609	0.576	0.611	0.711
Geometric Mean	0.713	0.714	**0.785**	0.614	0.591	0.610	0.601	0.627	0.714
RMS	0.717	0.728	0.782	0.624	0.615	0.612	0.584	0.629	0.723
SAUP-Learned	**0.771**	**0.755**	0.778	**0.669**	**0.638**	**0.626**	**0.787**	**0.783**	**0.809**

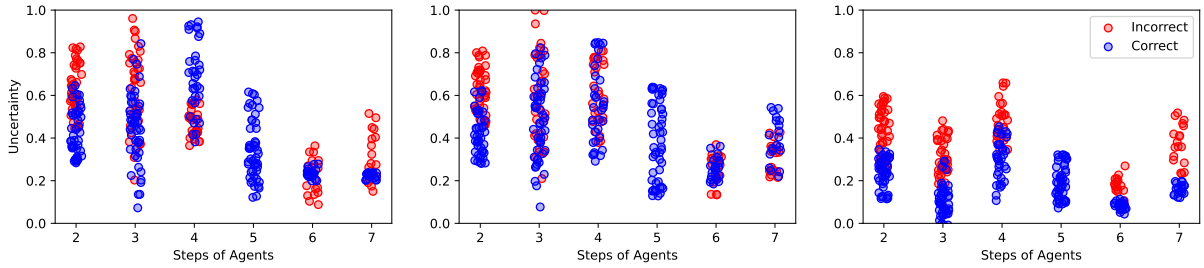

Figure 5: Visualization analysis of one-step methods (left), simple uncertainty propagation methods (middle) and SAUP (right) on the StrategyQA dataset. Detailed explanations of this figure are provided in the Q3 of Section 4.3.

X-axis represents the steps taken, and the Y-axis shows normalized uncertainty values. Red points indicate incorrect answers, and blue points indicate correct answers. SAUP (right) shows the clearest separation between correct and incorrect answers, outperforming the one-step (left) and simple uncertainty propagation methods (middle), highlighting its advantage in uncertainty estimation.

Q4: Is the HMM reasonable, and how does its performance change with different dataset sizes? Why not use gradient-based models like RNNs or Transformers?

Learned-based surrogates rely on manually annotated data. During training, we map data groups D_{a_n} and D_{o_n} to the agent's situational context, enabling SAUP to infer states in unseen scenarios. We use a Hidden Markov Model (HMM) in the main experiment, but also explore LSTM and Transformer models, analyzing their theoretical and experimental advantages.

Theoretical Perspective: HMMs are efficient and interpretable, ideal for limited data but weak in modeling long-range dependencies. LSTMs capture temporal dependencies better but require more data and resources. Transformers handle both local and global dependencies effectively but are computationally expensive and data-intensive.

Experimental Comparison: On the StrategyQA dataset, we evaluated HMM-based, LSTM-based, and mini-size Transformer-based surrogates varying training dataset sizes. Figure 4 shows that HMMs perform well with smaller datasets, while LSTMs and Transformers improve with more data. However, Transformer-based surrogates require impractically large datasets for uncertainty measurement tasks, making them less suitable.

HMMs are practical for uncertainty propagation in LLM-based agents due to their simplicity and efficiency, particularly with limited data. LSTMs are viable alternatives when data and computational resources are sufficient, while Transformers are generally not feasible for most scenarios.

Q5: Does the question difficulty influence the effectiveness of uncertainty propagation?

Yes, complex questions lead to longer, nuanced decision-making, increasing uncertainty accumulation. SAUP's situational awareness framework excels in such cases, effectively propagating uncertainty at each step. As shown in Table 1, SAUP's advantage is most evident in more challenging datasets like StrategyQA, with greater AUROC improvements.

5 Conclusion

In this paper, we propose Situational Awareness Uncertainty Propagation (SAUP), a novel framework for estimating uncertainty in LLM-based multi-step agents. Unlike traditional methods focused on single-step uncertainty, SAUP propagates uncertainty across all reasoning steps while integrating situational awareness. Experiments on challenging datasets show that SAUP outperforms state-of-the-art methods, achieving up to 20% improvements in AUROC scores, demonstrating its effectiveness in enhancing reliability for complex decision-making. This research underscores the importance of multi-step uncertainty estimation and situational awareness in ensuring the trustworthy deployment of LLM-based agents.

6 Limitations

Despite the effectiveness of SAUP in improving uncertainty estimation for multi-step LLM-based agents, several limitations remain. First, the learning-based surrogate version of SAUP relies on manually annotated datasets for situational weights, which is time-consuming, costly, and may not generalize well to very complex scenarios—especially when manual labels are still prone to errors. Additionally, the complexity of diverse environments could exacerbate the difficulty in ensuring accurate situational labeling. Second, the SAUP framework assumes that uncertainty at each step can be accurately captured. Although this is beyond the scope of our study, errors in single-step uncertainty estimation can compromise the propagation of uncertainty, thereby diminishing the benefits of the SAUP framework. Future work should focus on developing more robust situational weight estimation methods that reduce dependence on manually annotated datasets—potentially leveraging LLM-generated labels—to enhance SAUP's applicability and reliability across diverse use cases.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Leonard E Baum and Ted Petrie. 1966. Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37(6):1554–1563.

Andrew P Bradley. 1997. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

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

Pei Chen, Soumajyoti Sarkar, Leonard Lausen, Balasubramaniam Srinivasan, Sheng Zha, Ruihong Huang, and George Karypis. 2024. Hytrel: Hypergraph-enhanced tabular data representation learning. *Advances in Neural Information Processing Systems*, 36.

Maxime Darrin, Pablo Piantanida, and Pierre Colombo. 2022. Rainproof: An umbrella to shield text generators from out-of-distribution data. *arXiv preprint arXiv:2212.09171*.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2024. 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.

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

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

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

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep bayesian active learning with image data. In *International conference on machine learning*, pages 1183–1192. PMLR.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Jiuzhou Han, Wray Buntine, and Ehsan Shareghi. 2024. Towards uncertainty-aware language agent. *arXiv preprint arXiv:2401.14016*.

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

S Hochreiter. 1997. Long short-term memory. *Neural Computation MIT-Press*.

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

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

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

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

Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Andrey Malinin and Mark Gales. 2020. Uncertainty estimation in autoregressive structured prediction. *arXiv preprint arXiv:2002.07650*.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey. *arXiv preprint arXiv:2402.06196*.

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

Shuofei Qiao, Honghao Gui, Chengfei Lv, Qianghuai Jia, Huajun Chen, and Ningyu Zhang. 2023. Making language models better tool learners with execution feedback. *arXiv preprint arXiv:2305.13068*.

Mohaimenul Azam Khan Raiaan, Md Saddam Hossain Mukta, Kaniz Fatema, Nur Mohammad Fahad, Sadman Sakib, Most Marufatul Jannat Mim, Jubaer Ahmad, Mohammed Eunus Ali, and Sami Azam. 2024. A review on large language models: Architectures, applications, taxonomies, open issues and challenges. *IEEE Access*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

Matthew Renze and Erhan Guven. 2024. Self-reflection in llm agents: Effects on problem-solving performance. *arXiv preprint arXiv:2405.06682*.

SerpAPI. 2024. Real-time search api for google results. https://serpapi.com.

Noah Shinn, Beck Labash, and Ashwin Gopinath. 2023. Reflexion: an autonomous agent with dynamic memory and self-reflection. *arXiv preprint arXiv:2303.11366*, 2(5):9.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345.

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

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. 2023. Autogen: Enabling next-gen llm applications via multi-agent conversation. *Preprint*, arXiv:2308.08155.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. 2023. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*.

Yijun Xiao and William Yang Wang. 2019. Quantifying uncertainties in natural language processing tasks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7322–7329.

Yijun Xiao and William Yang Wang. 2021. On hallucination and predictive uncertainty in conditional language generation. *arXiv preprint arXiv:2103.15025*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do large language models know what they don't know? *arXiv preprint arXiv:2305.18153*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.