MEMUQ: MEMORY SPACE UNCERTAINTY QUANTIFICATION FOR LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

As Large Language Models (LLMs) are increasingly deployed as agents in multi-step reasoning and tool-based interaction, reliable uncertainty quantification (UQ) is essential for addressing hallucination and assessing trustworthiness of their decisions. However, existing UQ methods target single-step or response-level settings, failing to capture how uncertainty propagates as agent context and memory evolves, resulting in unreliable confidence estimates that overlook error accumulation and conflicting evidence across steps. To address these limitations, we propose MemUQ, a novel UQ framework leveraging the agent memory space to model both intrinsic generation uncertainty and extrinsic uncertainty from multi-step inconsistencies. MemUQ decomposes agent memory into atomic claims, aggregates them across steps and trajectories, and constructs a conflict graph to quantify claim reliability by combining graph-based importance with frequency-based support. This yields fine-grained, interpretable uncertainty signals grounded in interaction history. Experiments on HotpotQA and StrategyQA demonstrate up to 27% AUROC improvements in error detection over baselines, showing that combining intrinsic uncertainty with memory-centric, claim-level extrinsic uncertainty provides an effective approach to UQ for LLM agents in complex decision-making settings.

1 INTRODUCTION

Large Language Model (LLM) agents are rapidly becoming integral to complex decision-making systems, from multi-step reasoning tasks to autonomous tool use in real-world applications (Wang et al., 2024; Liu et al., 2024). These agents decompose complex problems into sequences of actions, maintaining context across multiple interaction steps to arrive at final answers. However, despite their impressive capabilities, LLM agents are susceptible to hallucinations and can propagate errors throughout decision trajectories (Huang et al., 2025). This fundamental reliability issue makes it crucial to quantify when and to what degree their outputs can be trusted, particularly in high-stakes applications where incorrect decisions may have serious consequences (Gawlikowski et al., 2023; Yin et al., 2024).

The key challenge in uncertainty quantification (UQ) for LLM agents lies in the accumulative nature of uncertainty across multi-step interactions. Traditional methods such as semantic entropy (Kuhn et al., 2023) and predictive entropy (Malinin & Gales, 2021) are designed for single-step question-answering with static context, and therefore fail to capture how uncertainty propagates as agents accumulate information and encounter conflicting evidence (Lin et al., 2024b; Qiu & Miikkulainen, 2024). Each reasoning step introduces its own uncertainty while also inheriting uncertainty from previously gathered external information, creating cascading errors in which early mistakes influence subsequent reasoning (Duan et al., 2025; Kirchhof et al., 2025).

Duan et al. (2025) addresses this issue through an information-theoretic decomposition, estimating extrinsic uncertainty as cumulative mutual information between sequential decisions in the action space. However, the action space provides only an indirect view of agent uncertainty: it captures how decisions constrain future decisions, but does not explicitly encode the variability, inconsistency, or contradiction among the observations that drive those decisions. Since LLM agents primarily rely on external evidence, uncertainty should also be assessed in terms of the accumulated observations themselves.

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

Measuring uncertainty directly in the observation space is challenging, as observations from external sources can be arbitrarily long, arrive in different orders across trajectories, and lack consistent structure for comparison. These properties make direct application of traditional UQ methods infeasible and motivate a structured memory-space representation for comparing accumulated evidence across trajectories.

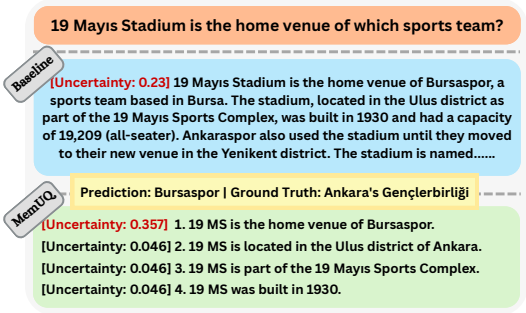


Figure 1: Interpretability comparison of our method (bottom) against baseline (top). In the final step of agentic generation, MemUQ captures fine-grained claim-level uncertainty, whereas the baseline aggregates uncertainty at the memory level, reducing interpretability and often leading to overconfident estimates.

Therefore, we propose to quantify uncertainty in the LLM agent **memory** space, leveraging the memory-based LLM agent architecture (Yu et al., 2026). The memory space serves as a structured repository for the agent's trajectory-level knowledge, capturing both accumulated evidence and inter-evidence relationships. By analyzing consistency patterns within this memory space, we can directly measure extrinsic uncertainty arising from conflicting or evolving information.

However, agent memories typically consist of lengthy, unstructured text that makes direct application of traditional UQ methods unreliable and lacks interpretability. To overcome this limitation, we propose to decompose the memory into atomic claims, which are self-contained, verifiable statements (Jiang et al., 2024), and track their semantic consistency across independently-sampled trajectories to detect conflicting evidence that single-turn graph-based methods cannot observe. As shown in Figure 1, this decomposition enables us to track how individual claims evolve, conflict, or reinforce each other across the agent's trajectory. We then construct a conflict graph over claims (with claims as nodes and conflicts as edges) and apply various graph centrality metrics to identify claims at the core of contradictory evidence, enabling detection of inconsistencies that signal high uncertainty. In summary, our main contributions are:

1. **Memory-based extrinsic uncertainty quantification:** We propose the MemUQ framework, leveraging the agent's memory space for UQ and demonstrating that it encodes extrinsic uncertainty through consistency and conflict patterns across accumulated observations.
2. **Bridging traditional UQ to agentic systems:** We enable established UQ methods (Qiu & Miikkulainen, 2024; Farquhar et al., 2024) to operate on multi-step agent interactions via memory consistency, without extensive modifications.
3. **Claim-level decomposition with graph-based aggregation:** Unlike graph-based UQ methods that operate on single-turn generation (Jiang et al., 2024), we construct a cross-trajectory conflict graph over claims extracted from K independently-sampled agent trajectories, applying centrality metrics (closeness, betweenness, eigenvector, PageRank) to identify claims central to contradictory evidence and produce fine-grained, interpretable uncertainty signals that substantially outperform existing methods.

2 Preliminaries

We introduce the foundational concepts of our MemUQ framework: uncertainty propagation in agentic interactions (§2.1) and the memory-based agent architecture (§2.2).

2.1 Uncertainty in LLM Agent Interactions

LLM agents operating in multi-step, interactive environments face fundamentally different uncertainty challenges than single-step question answering (Duan et al., 2025; Kirchhof et al., 2025). Following prior work (Liu et al., 2024; Duan et al., 2024b), multi-step agentic decision-making is modeled as a stochastic Markov Decision Process $(f_\theta, \mathcal{O}, \mathcal{Y}, T)$, where the LLM f_θ interacts with an environment continuously. Here $\mathcal{O}$ and $\mathcal{Y}$ denote the observation and output spaces, respectively, and $T : \mathcal{Y}^* \to \mathcal{O}$ is the observation transition function. At time step t, given input x, the agent receives observation

$o_t \in \mathcal{O}$ from the environment and generates output $y_t \in \mathcal{Y}$ according to:

$$y_t \sim p_\theta(y_t | o_{1:t}, y_{1:t-1}, x). \tag{1}$$

In this setting, Duan et al. (2025) decompose uncertainty into two components: (1) *Intrinsic Uncertainty*: The expected variance of y_t given all preceding context, $\mathbb{E}_{y_{1:t-1}}[\mathrm{Var}_{y_t}(y_t | y_{1:t-1}, x)]$, reflecting generation uncertainty at the current step. (2) *Extrinsic Uncertainty*: Variance of y_t introduced by prior decisions, $\mathrm{Var}_{y_{1:t-1}}[\mathbb{E}_{y_t}[y_t | y_{1:t-1}, x]]$, i.e., uncertainty inherited from prior steps.

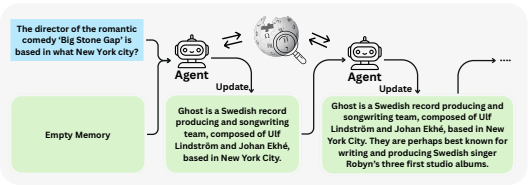


Figure 2: Overview of the memory-based agent interaction framework. At each step, the agent receives an observation from the environment and updates its memory state.

While intrinsic uncertainty can be estimated via standard MC sampling at each step, extrinsic uncertainty is more challenging to quantify. Duan et al. (2025) estimates it via pointwise mutual information between sequential decisions in the action space. MemUQ instead captures a complementary signal: rather than measuring information-theoretic dependence along the actions made, we measure the semantic consistency of evidence accumulated in the agent's memory, detecting contradictions that arise from conflicting observations across trajectories.

2.2 Memory-Based Agent Architecture

We adopt the memory-based LLM agent framework (Yu et al., 2026), in which the agent maintains a structured memory state that is iteratively updated throughout its interaction with the environment. While the original framework targets long-context document processing, we leverage the memory space as a principled substrate for uncertainty quantification.

At each interaction step t, the agent receives an observation o_t from the environment and updates its memory state according to:

$$m_t = \Phi(m_{t-1}, o_t), \quad m_0 = \varnothing, \tag{2}$$

where $\Phi(\cdot)$ denotes memory update by the LLM agent via prompting. The memory resides in token space rather than feature space, making it interpretable and comparable across trajectories.

Unlike raw observations, which vary in length and ordering across trajectories, or final answers, which discard intermediate evidence, the memory state accumulates task-relevant information in a structured and consistent form. This makes it a suitable representation for detecting inconsistencies that arise from conflicting observations. We refer to the accumulated memory as the **memory space** of the agent and leverage it as the central object for quantifying extrinsic uncertainty.

3 Memory Space Uncertainty Quantification

We present a memory-centric, claim-level uncertainty quantification framework for LLM agents in multi-step interactions. As shown in Figure 3, we decompose agent memory into atomic claims, construct conflict graph to capture cross-trajectory inconsistencies, and estimate extrinsic uncertainty by combining graph centrality (identifying contradictory evidence) with claim frequency (approximating factuality), yielding fine-grained, interpretable uncertainty signals.

3.1 Uncertainty Propagation in Memory Space

As described in §2.2, the memory space provides a unified aggregation of retrieved information that absorbs variability in the observation space, distilling relevant facts into a structured and comparable representation regardless of retrieval order or observation length. We leverage this **memory space** as the central object for uncertainty quantification. Given trajectory $\tau = (o_1, y_1), \cdots, (o_t, y_t)$, the memory state m_t evolves via Eq. equation 2 and the final output $y_T \sim p_\theta(y_T \mid m_T, x)$; prompts for memory updates are provided in Appendix D. Under this formulation, uncertainty from multi-step observations is encoded as inconsistencies in memory content across steps and trajectories.

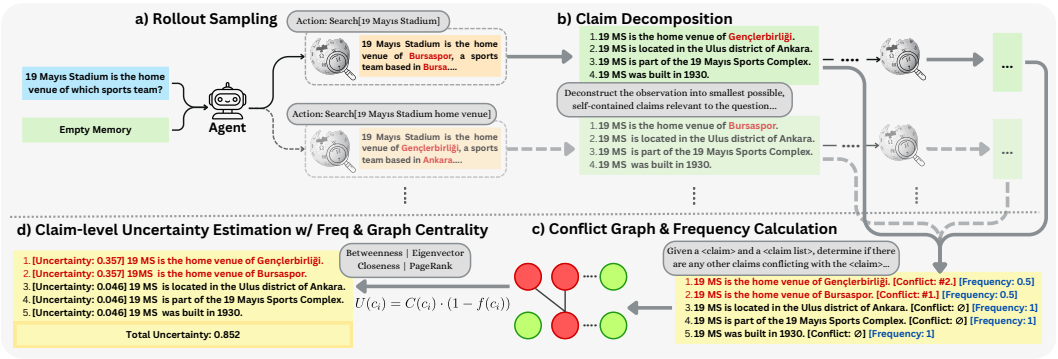


Figure 3: Overview of the MemUQ framework. (a) We sample multiple rollout trajectories per query, tracking environment interactions and storing observations in memory. (b) Agent memory is stored as atomic, self-contained claims. (c) A conflict graph is constructed by detecting contradictions between claims, and claim frequencies are computed across trajectories. (d) Claim-level uncertainty combines graph centrality with frequency-based support, yielding fine-grained, interpretable scores.

Uncertainty in Memory Across Interactions. Since y_T is generated from m_T, the entropy $H(m_T \mid x)$ characterizes variability in the terminal memory state. However, focusing solely on m_T can obscure extrinsic uncertainty signals: conflicting information retrieved at earlier steps may be overwritten by subsequent updates, leaving no trace in the final state. To capture such effects, we aggregate memory content across all interaction steps $m_{1:T}$ rather than relying only on m_T.

3.2 Claim Set Construction

Although agent memory provides a distilled abstraction of raw observations, it still takes the form of long, unstructured text unsuitable for direct UQ. Moreover, observations within a single trajectory target different sub-questions and are thus not directly comparable. To address this, we decompose each memory state into a set of **atomic claims**—minimal, self-contained units of information that can be independently verified and compared. By sampling K independent rollout trajectories, we enable cross-trajectory comparison of claims targeting the same underlying facts, revealing conflicts from retrieval stochasticity or inconsistent external evidence.

Claim Decomposition. For a memory state m_t, we apply a decomposition function $\mathcal{C}(\cdot)$ that extracts atomic claims $\mathcal{C}(m_t) = \{c_1^{(t)}, c_2^{(t)}, \ldots, c_{l_t}^{(t)}\}$, where each $c_i^{(t)}$ is a self-contained factual statement. In practice (Figure 3b), we prompt the agent to store information as discrete claims directly in memory rather than paragraph-style memory with post-hoc decomposition, incurring no additional inference cost (Appendix D.1). The trajectory-level claim set is defined as $\mathcal{C}(\tau) = \bigcup_{t=1}^{T} \mathcal{C}(m_t)$, ensuring that claims retrieved earlier but later overwritten are still retained for uncertainty estimation.

Claim Aggregation and Conflict Detection. Given K sampled trajectories $\mathcal{T} = \{\tau^{(1)}, \ldots, \tau^{(K)}\}$, we aggregate all claims into a multiset $\mathcal{C}(\mathcal{T}) = \bigcup_{k=1}^{K} \mathcal{C}(\tau^{(k)})$ and construct a **unique claim set** $\mathcal{V}$ by filtering out semantically duplicate claims through the LLM agent. To capture inconsistencies, we construct a **conflict graph** $G = (\mathcal{V}, \mathcal{E})$, where an edge $(c_i, c_j) \in \mathcal{E}$ exists if claims c_i and c_j are judged to be contradictory. Claims with no conflicts remain as isolated vertices, providing a structured representation of consistency and conflict relationships (Appendix E).

3.3 Quantifying Claim-Level Uncertainty

Uncertainty arises when retrieved observations for the same underlying facts exhibit *high variability*, such that few claims are consistently supported across sampled rollouts. At the claim level, this manifests as *conflicting claims* and *low empirical frequency* for individual claims.

Claim-Level Extrinsic Uncertainty. To obtain the final uncertainty score for each claim, we combine its structural importance within the conflict graph with its empirical support across trajectories.

Table 1: Graph centrality metrics (Jiang et al., 2024) applied to the conflict graph. Higher centrality indicates a claim is more central to the network of contradictions, reflecting greater uncertainty.

Metric	Formula	Intuition
Betweenness	$C_B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$	Proportion of shortest paths between node pairs that traverse v
Eigenvector	$C_E(v) = \frac{1}{\lambda} \sum_{u \in N(v)} A_{vu} C_E(u)$	Recursive influence measure weighted by neighbor centralities
PageRank	$C_{PR}(v) = \frac{1-d}{\lvert V \rvert} + d \sum_{u \in N(v)} \frac{C_{PR}(u)}{N(u)}$	Steady-state visitation probability under random walk with restart, where d is the damping factor
Closeness	$C_C(v) = \frac{\lvert V \rvert - 1}{\sum_{u \in V} d(v,u)} \cdot \frac{\lvert V \rvert}{\lvert V_v \rvert}$	Inverse mean shortest-path distance to all nodes

Specifically, we define the claim-level extrinsic uncertainty as:

$$U(c_i) = \bar{C}(c_i) \cdot (1 - f(c_i)), \tag{3}$$

where $\bar{C}(c_i)$ is the normalized centrality score (defined below) and $f(c_i)$ is the normalized frequency of claim c_i. A claim should be assigned high uncertainty if it satisfies two conditions: 1) it has high conflict centrality, indicating that it lies at the center of contradictory evidence, and 2) it has low frequency across trajectories, suggesting that the agent rarely retrieves this information consistently.

Conversely, claims that appear frequently across trajectories with few conflicts receive low uncertainty scores, as they represent stable, well-supported information. This multiplicative combination ensures that both signals must be present for a claim to be flagged as highly uncertain.

Graph-based Variability Measure. We apply graph centrality measures to the conflict graph to capture how strongly each claim is implicated in conflicting evidence. While Jiang et al. (2024) applies centrality to a bipartite entailment graph over a single response to measure how well claims are supported, our conflict graph operates over claims from K independently sampled *agent trajectories* and uses contradiction edges, so centrality here quantifies how central a claim is to cross-trajectory disagreement rather than within-response support. Let $C : \mathcal{V} \to \mathbb{R}^+$ denote a centrality function; we adapt four metrics: Betweenness (C_B), Eigenvector (C_E), Closeness (C_C), and PageRank (C_{PR}), with formal definitions in Table 1 (notation in Appendix B). Since these metrics live on different scales (e.g., PageRank sums to one while closeness grows with the number of claims), we normalize the centralities within each query:

$$\bar{C}(c_i) = \frac{C(c_i)}{\sum_{j=1}^{N} C(c_j)}, \tag{4}$$

so that $\bar{C}$ is a distribution over claims that we refer to as the *conflict salience*. When all centralities are zero (e.g., betweenness on an edgeless graph), we use the uniform $\bar{C}(c_i) = 1/N$. PageRank is unchanged by this normalization.

Frequency-Based Support Estimation. While centrality reveals structural importance, it does not indicate which claims are most reliable. We measure how frequently each claim appears across trajectories as a probabilistic estimate of correctness. Let $n_i = \sum_{k=1}^{K} \mathbf{1}[c_i \in \tau^{(k)}]$ count the trajectories containing claim c_i. The normalized frequency is then:

$$f(c_i) = \frac{n_i}{\sum_{j=1}^{N} n_j}, \tag{5}$$

where $N = |\mathcal{V}|$. Normalizing by total claim count rather than a fixed constant implicitly encodes task complexity: when the agent produces many unique claims, each individual claim receives lower weight, reflecting that the agent encountered diverse or noisy information. For instance, a claim appearing in all K trajectories still receives low $f(c_i)$ if the total claim set is large, appropriately down-weighting per-claim confidence in high-noise settings. This normalization consistently outperforms fixed-constant alternatives (Table 4, Appendix F).

Aggregating to Trajectory-Level Uncertainty. For downstream applications requiring a single uncertainty score per query, we aggregate claim-level scores over all claims:

$$U_{\text{ext}}(x) = \sum_{c_i \in \mathcal{V}} \bar{C}(c_i) \cdot (1 - f(c_i)) = 1 - \sum_{c_i \in \mathcal{V}} \bar{C}(c_i) f(c_i). \tag{6}$$

Since both $\bar{C}$ and f sum to one, $U_{\text{ext}}(x) \in [0, 1]$ for every centrality metric. The following proposition (proved in Appendix C) shows what this score measures.

Proposition 3.1 (Covariance decomposition)**.** *Let* $m = |\mathcal{V}|$ *and let* $\text{Cov}(\bar{C}, f) = \frac{1}{m}\sum_{i=1}^{m} \bar{C}(c_i) f(c_i) - \frac{1}{m^2}$ *denote the population covariance of the conflict salience and the frequency across claims. Then*

$$1 - U_{ext}(x) = \frac{1}{m} + m \cdot \text{Cov}(\bar{C}, f). \tag{7}$$

Both terms are properties of the agent's own behavior rather than of the graph construction. The $1/m$ term is set by the number of distinct claims produced across trajectories and reflects cross-trajectory consistency: agents that retrieve the same evidence in every rollout produce few unique claims. The covariance term measures where the agent places its support relative to where the disagreement lies. A positive covariance means that the claims at the center of the conflict structure are also the claims most trajectories repeated, so the agent's evidence backs the contested content and uncertainty decreases. A negative covariance means that the contested claims are the rare ones, appearing in few trajectories, which indicates disagreements driven by isolated, unsupported observations, and uncertainty increases. The conflict graph therefore does not inject uncertainty of its own; it reweights the frequency evidence the agent generated.

We combine $U_{\text{ext}}(x)$ with intrinsic uncertainty (e.g., semantic entropy from step-level sampling) via a simple average to obtain the total uncertainty estimate. As with established sampling-based UQ methods (Farquhar et al., 2024; Qiu & Miikkulainen, 2024), the resulting score is a discriminative signal for ranking outputs and selective prediction, not a calibrated error probability.

4 Related Work

Uncertainty Quantification in LLMs. In auto-regressive LLMs, uncertainty quantification (UQ) measures confidence within the predictive distribution (Malinin & Gales, 2021). From an information-theoretic perspective, uncertainty can be measured through the entropy of generations sampled from the output space (Gawlikowski et al., 2023). However, standard entropy often overestimates uncertainty by treating semantically equivalent outputs as distinct. Semantic Entropy (Kuhn et al., 2023; Farquhar et al., 2024) addresses this by clustering outputs by meaning and computing cluster-wise entropy. Deg (Lin et al., 2024b) models output consistency through graph-based methods using node connectivity or eigenvalues of semantic similarity graphs, which INSIDE (Chen et al., 2024) extends to LLM hidden spaces. SAR (Duan et al., 2024a) reveals token-level and sentence-level semantic imbalance, with CSL (Lin et al., 2024a) further refining token-level semantic weighting. Semantic Density (Qiu & Miikkulainen, 2024) draws from maximum probability theory to calculate uncertainty as the density of generations within a semantic space. To mitigate semantic overlapping among clusters, KLE (Nikitin et al., 2024) encodes semantic similarities via positive semidefinite unit trace kernels and quantifies uncertainty using von Neumann entropy. These methods primarily target non-agentic tasks with static context.

Claim Decomposition. Claim decomposition approaches analyze LLM outputs by breaking text into atomic, verifiable statements for fine-grained evaluation. Jiang et al. (2024) constructs an entailment graph over a single response to derive graph-based uncertainty metrics; Liu & Wu (2025) extends this to multi-group uncertainty estimation for long-form text. Both methods are designed for static, single-turn settings and are therefore not suited to modeling cross-trajectory inconsistencies. MemUQ instead compares claims across K independently sampled agent trajectories, constructs contradiction rather than entailment edges, and quantifies evidence retrieval consistency through a frequency component, providing signals tailored to multi step agentic settings.

Uncertainty in LLM Agent Interactions. LLM agents face distinct uncertainty challenges as evidence accumulates across multi-step interactions (Kirchhof et al., 2025). Duan et al. (2025)

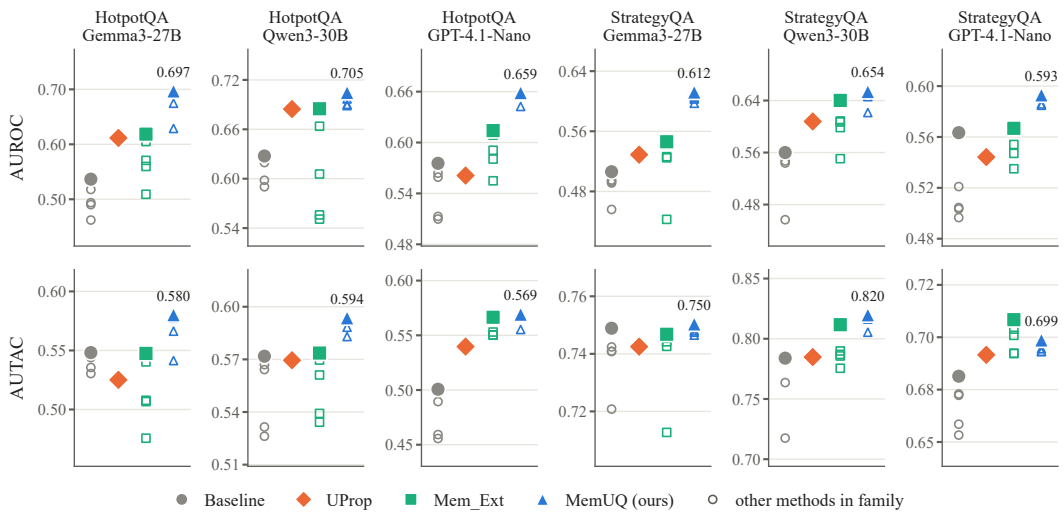


Figure 4: **MemUQ gives the strongest uncertainty signal across datasets and models.** AUROC (top) and AUTAC (bottom) per method family: Baseline (single-step UQ methods), UProp†, Mem_Ext, and MemUQ (Perplexity combined with each centrality metric). Filled markers show each family's best method; hollow markers show the rest. Exact values are in Table 8. † Customized implementation based on Duan et al. (2025).

formalizes this as extrinsic uncertainty inherited from preceding decisions, estimating it via mutual information in the action space; Zhao et al. (2025) instead aggregates per-step uncertainty using a trained Hidden Markov Model. MemAgent (Yu et al., 2026) introduces a structured, token-space memory that is updated throughout agent interactions. While originally designed for memory-augmented reasoning rather than uncertainty quantification, this formulation motivates our use of memory as a substrate for analyzing extrinsic uncertainty via cross-trajectory consistency.

5 Experiments

5.1 Experimental Setup

Models & Datasets. We evaluate MemUQ on two multi-hop question answering benchmarks that require active retrieval and reasoning over external information. **HotpotQA** (Yang et al., 2018) requires the agent to retrieve and reason across multiple Wikipedia search steps, directly testing UQ in active tool use where complex questions demand multiple pieces of evidence and resolution of potential conflicts. **StrategyQA** (Geva et al., 2021) features yes/no questions requiring multi-hop reasoning over implicit steps, making it effective for evaluating UQ in scenarios where uncertainty arises from implicit reasoning chains. We evaluate three representative LLMs: **Qwen3-30B** (Yang et al., 2025), **Gemma3-27B-Instruct** (Gemma Team et al., 2025), and **GPT-4.1-Nano** (Achiam et al., 2023), covering both open and proprietary model families. We use GPT-4.1-Nano as the GPT-5 series do not expose token probabilities required by baseline methods. We sample $K = 10$ trajectories per question with temperature 0.7 and top-$p = 0.95$, generate 10 samples per step for intrinsic uncertainty estimation, and evaluate on the first 500 test entries per dataset.

Baselines. We compare MemUQ against two categories of baselines, all using the same sampling budget of $K = 10$ trajectories and 10 samples per step. The first category consists of existing single-step UQ methods applied directly to ReAct-style agent trajectories: **Perplexity**, the length-normalized negative log-likelihood of generated answers, where lower perplexity indicates higher confidence; **Normalized Entropy (NE)** (Malinin & Gales, 2021), the entropy of the model's output distribution normalized by answer length; **Semantic Entropy (SE)** (Farquhar et al., 2024), the entropy over semantic clusters of sampled responses, capturing diversity in meaning rather than surface form; **Semantic Density (SD)** (Qiu & Miikkulainen, 2024), which measures the density of sampled outputs in semantic space, with denser clusters indicating lower uncertainty; and **Degree of Disagreement (Deg_E)** (Lin et al., 2024b), which measures sample disagreement via semantic similarity.

Table 2: Intrinsic vs. extrinsic ablation on HotpotQA. Gaps are relative to the Combined score.

	Gemma3-27B		Qwen3-30B		GPT-4.1-Nano	
Method	**AUROC**	**AUTAC**	**AUROC**	**AUTAC**	**AUROC**	**AUTAC**
Perplexity (Intrinsic)	0.6050 (-13.21%)	0.5475 (-5.66%)	0.6608 (-6.24%)	0.5685 (-4.24%)	0.6085 (-7.61%)	0.5472 (-3.65%)
PageRank (Extrinsic)	0.6786 (-2.65%)	0.5662 (-2.43%)	0.6905 (-2.03%)	0.5862 (-1.26%)	0.6566 (-0.30%)	0.5615 (-1.13%)
Combined	**0.6971**	**0.5803**	**0.7048**	**0.5937**	**0.6586**	**0.5679**

The second is **UProp** (Duan et al., 2025), an agentic UQ method, which estimates extrinsic uncertainty as the cumulative mutual information between sequential decisions in the action space; since no public implementation is available, we reimplement it following the paper.

Additionally, we include **Mem_Ext**, a memory-based adaptation that applies single-turn UQ methods to paragraph-style agent memory rather than directly to agent trajectories. This provides a consistent trajectory-level representation of accumulated observations. For MemUQ, we combine intrinsic uncertainty with our claim-level extrinsic measures. Due to space constraints, Figure 4 reports results using Perplexity as the intrinsic measure; full intrinsic–extrinsic combinations are provided in Appendix I, with detailed UQ algorithms in Appendix K.

Metrics. Following Kuhn et al. (2023), we evaluate UQ performance using two complementary metrics. **AUROC** (Area Under the Receiver Operating Characteristic Curve) measures how well uncertainty scores discriminate between correct and incorrect agent outputs. **AUTAC** (Area Under the True Accuracy Curve) measures accuracy as uncertain predictions are progressively rejected, summarizing the accuracy–rejection trade-off; higher values indicate better selective prediction.

5.2 Results

Claim-Level Uncertainty Outperforms Baselines. Figure 4 reveals a clear three-tier progression. Baseline methods perform near-randomly (e.g., AUROC 0.46–0.54 on HotpotQA for Gemma3), since each trajectory accumulates its own retrieval history and final-answer distributions across trajectories are not directly comparable. UProp partially addresses this issue by modeling uncertainty over the action space. However, action-level uncertainty provides only an indirect and limited signal, since uncertainty in retrieval-augmented agents is primarily driven by the consistency of the retrieved contextual evidence rather than by the particular actions used to acquire it.

Using memory as the UQ substrate (Mem_Ext) yields consistent gains across all models, as the memory encoding provides a stable, trajectory-level representation of accumulated evidence. MemUQ achieves further improvements (up to 27% over Baseline) by decomposing memory into atomic claims and constructing a conflict graph that surfaces claims central to contradictory evidence, capturing inconsistency signals invisible to both trajectory- and memory-level methods. The combination of intrinsic and extrinsic uncertainty is most effective, as they are complementary: intrinsic uncertainty reflects per-step model confidence, while extrinsic uncertainty captures cross-step evidential conflicts.

Effectiveness of Graph Centrality Metrics. With per-query normalization (Equation (4)), every centrality metric outperforms every Baseline method in all six settings, indicating that the conflict graph representation itself, rather than the specific centrality measure, is the primary driver of the improvement. PageRank achieves the best AUROC on both datasets for Qwen3-30B and on HotpotQA for Gemma3-27B, and is within 0.008 AUROC of the best centrality in the remaining settings. PageRank assigns higher scores to claims that conflict with other highly-contested claims, capturing a recursive notion of uncertainty: a claim is more uncertain not merely because it has conflicts, but because it conflicts with claims that are themselves central to the conflict structure. We therefore use PageRank as the default centrality.

5.3 Ablation Studies

We conduct ablation studies to analyze effectiveness of MemUQ. We examine the individual contributions of intrinsic and extrinsic uncertainty components below. Additional ablations verify task performance preservation (Appendix A), the reliability of the LLM-based claim pipeline (Appendix G), sensitivity to the number of sampled trajectories (Appendix J), memory token utilization (Appendix L), and memory representation formats (Appendix M).

Intrinsic vs Extrinsic Uncertainty. We investigate the effectiveness of intrinsic and extrinsic uncertainty individually. In Table 2, we report the AUROC and AUTAC when using each component alone on HotpotQA. Both components contribute to the overall performance, with their combination performing the best across all settings. Notably, using intrinsic uncertainty alone results in larger performance drops (up to 13.21% on Gemma3-27B) compared to using extrinsic uncertainty alone, indicating that our proposed graph-based extrinsic uncertainty is the more critical component for effective uncertainty quantification in agentic settings.

Frequency Normalization. We evaluate sensitivity to the frequency normalization in Eq. equation 5 by replacing the trajectory-dependent normalizing factor with a fixed, claim-number-independent constant ($K = 10$), which treats all claim occurrences equally regardless of trajectory structure. This alternative (`div_10`) assigns equal weight to all occurrences, discarding information about how individual trajectories distribute their support across claims.

As shown in Table 4 (Appendix F), the trajectory-dependent normalization (Freq) outperforms the fixed-constant alternative in all 48 AUROC and AUTAC cells across datasets, models, and centrality metrics. It is also the normalization under which f is a distribution over claims, which Proposition 3.1 relies on. Verbose trajectories spread their mass over many claims (lower per-claim weight), while focused trajectories concentrate mass (higher per-claim weight), implicitly reflecting trajectory-level confidence that uniform normalization discards.

Reliability of the Claim Pipeline. Since claim decomposition, deduplication, and conflict detection are performed by an LLM, we validate each stage to separate pipeline error from agent uncertainty (details in Appendix G). *Claim decomposition* is faithful: on 100 sampled memory updates from Gemma3-27B, 95.7% of claims are self-contained, no step drops a still-relevant fact from the previous memory, and only 0.7% of claims contradict the input. *Deduplication and conflict detection* match a frontier model: on a human-labeled subsample, the pipeline (Gemma3-27B) and GPT-5.5 agree with human judgment at the same rate (80%). *Residual error* is small at full scale: an NLI audit of the entire filtered claim set for Gemma3-27B on StrategyQA finds 0.91% residual duplicate pairs and 1.88% false-positive conflict edges. Since U_{ext} only reweights the agent's own claims (Proposition 3.1) and the measured residual error is below 2%, pipeline noise is unlikely to distort the score.

Computational Overhead. We measure the wall-clock cost of MemUQ-specific processing on an H200 GPU running Gemma3-27B on StrategyQA. The average claim processing time (filtering and conflict detection combined) is **2.03 seconds per question**. Collecting 10 trajectories per question requires 9.32 hours (33,552 seconds), so MemUQ's claim processing adds only $\sim$3% overhead relative to the trajectory collection cost shared by all methods. Since claim processing for each query is independent, this overhead can be trivially parallelized.

6 Conclusion

In this paper, we study UQ for LLM agents operating in multi-step, tool-based interaction settings. We introduce a memory-centric framework that treats the agent memory space as a unified representation for quantifying both intrinsic generation uncertainty and extrinsic uncertainty arising from conflicting information across interaction steps. Our approach decomposes agent memory into atomic claims, constructs a conflict graph to capture inconsistencies, and estimates claim-level uncertainty by combining graph centrality metrics with frequency-based support. Experiments on HotpotQA and StrategyQA show that our method achieves up to 27% AUROC improvements over strong baselines, with consistent gains across different model architectures.

Despite these strengths, MemUQ relies on LLM-based claim decomposition and conflict detection, which may introduce bias in identifying semantic relationships among claims. Extending to broader agentic settings with longer horizons and less structured evidence remains an important direction for future work. This work serves as an initial exploration into leveraging memory space for uncertainty quantification in LLM agents, demonstrating that memory provides a useful and expressive substrate for modeling agent uncertainty.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish writing, to draft sections of the paper, for retrieval and discovery of related work, and for research execution, specifically to write code for running experiments. We have not used generative AI tools for research ideation, for generating synthetic datasets, or for proving mathematical claims. Separately from this assistance, LLMs appear in this work as objects and components of the study: as the agents under evaluation, within the claim-processing pipeline (Section 3), and as an annotator in the pipeline validation, which we checked against human labels (Appendix G). We have reviewed all AI-assisted work: AI-drafted text was checked and revised by the authors, related work surfaced by AI tools was verified against the original papers, and AI-written experiment code was reviewed and tested by the authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

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

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. INSIDE: LLMs' internal states retain the power of hallucination detection. In *International Conference on Learning Representations*, 2024.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5050–5063, 2024a.

Jinhao Duan, Renming Zhang, James Diffenderfer, Bhavya Kailkhura, Lichao Sun, Elias Stengel-Eskin, Mohit Bansal, Tianlong Chen, and Kaidi Xu. GTBench: Uncovering the strategic reasoning capabilities of LLMs via game-theoretic evaluations. In *Advances in Neural Information Processing Systems*, volume 37, pp. 28219–28253, 2024b.

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

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, 56(Suppl 1):1513–1589, 2023.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

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

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. DeBERTa: Decoding-enhanced BERT with disentangled attention. In *International Conference on Learning Representations*, 2021.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55, 2025.

Mingjian Jiang, Yangjun Ruan, Prasanna Sattigeri, Salim Roukos, and Tatsunori B Hashimoto. Graph-based uncertainty metrics for long-form language model generations. *Advances in Neural Information Processing Systems*, 37:32980–33006, 2024.

Michael Kirchhof, Gjergji Kasneci, and Enkelejda Kasneci. Position: Uncertainty quantification needs reassessment for large language model agents. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 81665–81677, 2025.

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

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Contextualized sequence likelihood: Enhanced confidence scores for natural language generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 10351–10368, 2024a.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Generating with confidence: Uncertainty quantification for black-box large language models. *Transactions on Machine Learning Research*, 2024b. URL https://openreview.net/forum?id=DWkJCSxKU5.

Terrance Liu and Zhiwei Steven Wu. Multi-group uncertainty quantification for long-form text generation. In *Proceedings of the Forty-first Conference on Uncertainty in Artificial Intelligence*, volume 286 of *Proceedings of Machine Learning Research*, pp. 2659–2684, 2025.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. AgentBench: Evaluating LLMs as agents. In *International Conference on Learning Representations*, 2024.

Andrey Malinin and Mark Gales. Uncertainty estimation in autoregressive structured prediction. In *International Conference on Learning Representations*, 2021.

Alexander Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. Kernel language entropy: Fine-grained uncertainty quantification for llms from semantic similarities. *Advances in Neural Information Processing Systems*, 37:8901–8929, 2024.

Xin Qiu and Risto Miikkulainen. Semantic density: Uncertainty quantification for large language models through confidence measurement in semantic space. In *Advances in Neural Information Processing Systems*, volume 37, pp. 134507–134533, 2024.

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

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Zhiyuan Zeng, Xiaonan Li, Junqi Dai, Qinyuan Cheng, Xuan-Jing Huang, and Xipeng Qiu. Reasoning in flux: Enhancing large language models reasoning through uncertainty-aware adaptive guidance. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2401–2416, 2024.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and Hao Zhou. MemAgent: Reshaping long-context LLM with multi-conv RL-based memory agent. In *International Conference on Learning Representations*, 2026.

Qiwei Zhao, Dong Li, Yanchi Liu, Wei Cheng, Yiyou Sun, Mika Oishi, Takao Osaki, Katsushi Matsuda, Huaxiu Yao, Chen Zhao, Haifeng Chen, and Xujiang Zhao. Uncertainty propagation on LLM agent. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6064–6073, 2025.

A REACT-STYLE ACCURACY COMPARISON

Since our claim-level memory architecture differs from the traditional ReAct (Yao et al., 2023) framework, we verify that it does not degrade task performance. Table 3 reports exact match scores on StrategyQA across 10 trajectories with Gemma3. Our claim-memory approach achieves comparable accuracy to both the paragraph-memory baseline and the ReAct framework, confirming that our approach preserves task performance while enabling fine-grained UQ.

Table 3: StrategyQA accuracy comparison across frameworks.

Method	**Acc**
Claim-Memory	77.64 ± 1.24
Paragraph-Memory	76.80 ± 1.25
Naive-ReAct	77.91 ± 0.83

B NOTATION DEFINITION

We provide the notation used in the graph centrality metrics (Table 1). Let $G = (V, A)$ denote a graph with node set V and adjacency matrix A, where A_{vu} indicates edge presence between nodes v and u.

- v, u, s, t: Nodes in graph G, where s and t denote source and target nodes for shortest path computations.
- σ_{st}: Total number of shortest paths between s and t.
- $\sigma_{st}(v)$: Number of shortest paths between s and t passing through v.
- $N(v)$: Neighbor set of node v.
- λ: Leading eigenvalue of the adjacency matrix A.
- d: Damping factor in PageRank, typically set to 0.85.
- $d(v, u)$: Shortest path distance between v and u.
- $|V|$: Total node count in the graph.
- $|V_v|$: Node count in the connected component containing v.

C PROOF OF PROPOSITION 3.1

Proof. Both $\bar{C}$ and f sum to one over the m claims, so each has mean $1/m$, and the population covariance across claims is

$$\mathrm{Cov}(\bar{C}, f) = \frac{1}{m}\sum_{i=1}^{m} \bar{C}(c_i) f(c_i) - \frac{1}{m} \cdot \frac{1}{m} = \frac{1}{m}\sum_{i=1}^{m} \bar{C}(c_i) f(c_i) - \frac{1}{m^2}.$$

From Equation equation 6, $\sum_i \bar{C}(c_i) f(c_i) = 1 - U_{\text{ext}}(x)$. Substituting gives $\mathrm{Cov}(\bar{C}, f) = \frac{1}{m}(1 - U_{\text{ext}}(x)) - \frac{1}{m^2}$, and multiplying by m and rearranging yields $1 - U_{\text{ext}}(x) = \frac{1}{m} + m \cdot \mathrm{Cov}(\bar{C}, f)$. □

D AGENT MEMORY IMPLEMENTATIONS

In this section, we provide the details about the prompts used in our proposed framework, as well as the original MemAgent (Yu et al., 2026) implementations served as baselines in our paper.

D.1 AGENT PROMPT WITH CLAIM DECOMPOSITION.

```
You are presented with a question and a current memory formatted as self-contained claims, along with historical actions.
Solve a question answering task with interleaving Thought, Action, Observation steps. Thought can reason about the current
situation, and Action can be three types:
(1) Search[entity], which searches the exact entity on Wikipedia and returns the first paragraph if it exists. If not, it will return
some similar entities to search.
(2) Lookup[keyword], which returns the next sentence containing keyword in the current passage.
(3) Finish[answer], which returns the answer and finishes the task.
Stop generation after an action is made to wait for observation.

Question: {question}

Current Memory: {memory}

Actions Done: {prev_actions}

Generate your response in this format (For actions, start line with "Action:", FOLLOW THIS STRICTLY):

Thought: [your reasoning]
Action: [Search[entity] OR Lookup[keyword] OR Finish[answer]]
```

Figure 5: Prompt for agent decision making with claim-style memory. Given current memory and the question, the agent is instructed to interact with the environment to gather more information.

```
You are presented with a question, a current memory (as an ordered list of standalone factual claims), and a new thought,
action, and observation. Your task is to update the memory as follows:
- Carefully read the question, current memory, thought/action, and observation.
- Deconstruct all information into the smallest possible, self-contained claims relevant to the question.
- Update the memory as an ordered list of claims.
- Preserve the order of existing claims as much as possible, append to the ordered list when adding a new claim.
- If a new observation conflicts with an existing claim, directly revise that claim in place to resolve the conflict; do not create
new claims for the correction.
- The final "Updated Memory" should be a concise, self-contained, ordered list of claims useful for answering the question.

Example of updated memory after two turns, with claim 2 modified due to conflict:
<example_memory>
1. Claim 1
2. Claim 2 (revised to resolve conflict)
3. Claim 3
</example_memory>

Question: {question}

Current Memory:
{memory}

Thought and action:
{thought_action}

Observation:
{observation}

Updated Memory:
*(Provide only the ordered list of updated claims, each as a single line. Revise claims in place as needed. No explanatory text.
Preserve the order of existing claims as much as possible, append to the ordered list when adding a new claim.)*
```

Figure 6: Prompt for agent memory update with claim decomposition. Given existing thought, action and observation, the agent is instructed to only retain information related to the question into the memory.

```
You are presented with a problem and a previous memory in the format of self-contained claims. Please answer the problem
based on the previous memory and format your response as follows "Therefore, the answer is (insert answer here)".

<problem>
{prompt}
</problem>

<memory>
{memory}
</memory>

Your answer:
```

Figure 7: Prompt for generating the final answer. After agent has gathered all necessary information, the final answer is generated given only the question and updated memory.

D.2 Baseline Agent Prompt with Memory Only.

```
You are presented with a question and a current memory, along with historical actions. Solve a question answering task with interleaving Thought, Action, Observation steps. Thought can reason about the current situation, and Action can be three types:
(1) Search[entity], which searches the exact entity on Wikipedia and returns the first paragraph if it exists. If not, it will return some similar entities to search.
(2) Lookup[keyword], which returns the next sentence containing keyword in the current passage.
(3) Finish[answer], which returns the answer and finishes the task.
Stop generation after an action is made to wait for observation.

Question: {question}

Current Memory: {memory}

Actions Done: {prev_actions}

Generate your response in this format:

Thought: [your reasoning]
Action: [Search[entity] OR Lookup[keyword] OR Finish[answer]]
```

Figure 8: Prompt for agent decision making with memory. Given current memory and the question, the agent is instructed to interact with the environment to gather more information.

```
You are presented with a question, a current memory, and a new thought, action, and observation. Please read the provided observation carefully and update the memory with the new information that helps to answer the problem. Be sure to retain all relevant details from the previous memory while adding any new, useful information.

Question: {question}

Current Memory:
{memory}

Thought and action:
{thought_action}

Observation:
{observation}

Updated Memory:
```

Figure 9: Prompt for paragraph-style agent memory update. Given existing thought, action and observation, the agent is instructed to only retain information related to the question into the memory.

```
You are presented with a problem and a previous memory. Please answer the problem based on the previous memory and format your response as follows "Therefore, the answer is (insert answer here)".

<problem>
{prompt}
</problem>

<memory>
{memory}
</memory>

Your answer:
```

Figure 10: Prompt for generating the final answer. After agent has gathered all necessary information, the final answer is generated given only the question and updated memory.

E Claim Processing

This section details the prompts used for claim processing in our framework. Claim processing consists of two key operations: (1) filtering semantically duplicate claims to construct the unique claim set $\mathcal{V}$, and (2) detecting conflicts between claim pairs to construct the edge set $\mathcal{E}$ of the conflict graph. Both operations leverage LLM-based semantic reasoning to handle the nuanced relationships between natural language claims.

E.1 Claim Filtering

```
Given a claim and a list of claims, determine if there are any claims have the same meaning with the given claim. Answer with yes if any of the claims in the given list have same meaning with the provided claim. If non of the claims have same meaning with the provided claim, answer no. Answer with yes or no only without any explanations. claim:{claim} claim list: {context}
```

Figure 11: Prompt for duplicate detection given the aggregated claims. For each claim, we prompt the LLM to mark duplicates regarding other claims.

E.2 CONFLICT DETECTION

```
Given a <claim> and a <claim list>, determine if there are any other claims conflicting with the <claim>. Facts can overlap
but not conflicting. Only identify as conflict if contradictory information appears. Give the result in json format as following
example: {json_format} \nIf no conflicting claim, put empty array as value. \n <claim>: {claim} \n<claim list>: {context}
```

Figure 12: Prompt for conflict detection given two claims.

F FREQUENCY NORMALIZATION ABLATION

Table 4 compares the trajectory-dependent frequency normalization of Equation (5) (Freq) with the fixed-constant alternative (`div_10`) discussed in Section 5.3.

Table 4: **Ablation study: Frequency-based vs. Diversity-based (div_10) claim weighting.** We compare two claim frequency calculation strategies for MemUQ on HotpotQA and StrategyQA using 10 trajectories for Gemma3-27B, Qwen3-30B, and GPT-4.1-Nano. Both methods combine Perplexity with different centrality metrics. Best results are **bolded**; second-best are underlined.

		Model	Gemma3-27B		Qwen3-30B		GPT-4.1-Nano	
		Metric	AUROC	AUTAC	AUROC	AUTAC	AUROC	AUTAC
HotpotQA	Freq	C_B	0.6288	0.5414	0.6888	0.5830	0.6425	0.5553
		C_E	0.6747	0.5663	0.6903	0.5883	**0.6589**	0.5670
		C_C	0.6918	0.5773	0.6969	0.5924	0.6585	**0.5693**
		C_{PR}	**0.6971**	**0.5803**	**0.7048**	**0.5937**	0.6586	0.5679
	div_10	C_B	0.5913	0.5215	0.6607	0.5707	0.5618	0.5121
		C_E	0.6314	0.5435	0.6511	0.5714	0.5921	0.5316
		C_C	0.6510	0.5577	0.6570	0.5760	0.5912	0.5319
		C_{PR}	0.6646	0.5648	0.6624	0.5781	0.5932	0.5322
StrategyQA	Freq	C_B	0.5972	0.7475	0.6215	0.8054	**0.5931**	**0.6987**
		C_E	**0.6122**	0.7480	0.6460	0.8164	0.5860	0.6932
		C_C	0.6029	0.7465	0.6510	0.8169	0.5849	0.6932
		C_{PR}	0.6051	**0.7504**	**0.6539**	**0.8199**	0.5911	0.6945
	div_10	C_B	0.5571	0.7337	0.5495	0.7717	0.5513	0.6697
		C_E	0.5586	0.7312	0.5692	0.7847	0.5383	0.6636
		C_C	0.5453	0.7239	0.5718	0.7855	0.5342	0.6633
		C_{PR}	0.5374	0.7233	0.5641	0.7816	0.5414	0.6672

G RELIABILITY OF THE CLAIM PIPELINE

MemUQ relies on three LLM-managed stages: claim decomposition during memory updates, duplicate filtering to construct the unique claim set $\mathcal{V}$, and conflict detection to construct the edge set $\mathcal{E}$. Errors in these stages could in principle be mistaken for agent uncertainty. We therefore validate each stage separately.

Claim Decomposition. We sample 100 memory-update steps from the Gemma3-27B runs on HotpotQA and StrategyQA and have GPT-5.5 annotate each produced claim against the previous memory and the agent's thought and action at that step. For each claim, the annotator judges whether it is self-contained and whether it is derivable from the shown material; for each step, it judges whether any still-relevant fact from the previous memory was dropped. Table 5 shows that decomposition is faithful: no step drops a still-relevant fact, and contradictions of the input are rare. The main imperfection is granularity, with 29.8% of claims bundling more than one fact. This affects how finely the conflict graph can attribute a disagreement, but not whether the disagreement is captured.

Deduplication and Conflict Detection. We annotate 100 sampled decisions per stage with GPT-5.5, and independently hand-label 10 items per stage as a human anchor, with both the pipeline's decision (Gemma3-27B) and the GPT-5.5 label hidden during labeling. As shown in Table 6, the pipeline and GPT-5.5 agree with human judgment at the same overall rate (80%), while agreeing with each other

Table 5: Quality of claim decomposition on 100 sampled memory-update steps (Gemma3-27B, HotpotQA and StrategyQA), annotated by GPT-5.5.

Property	**Rate**
Claims that are self-contained	95.7%
Information loss from the previous memory	0%
Claims contradicting the shown input	0.7%
Claims that are fully atomic (single fact)	70.2%

on only 63% of the 200 annotated decisions. The two automatic annotators therefore disagree with each other more than either disagrees with the human labels, so neither is systematically closer to human judgment, and the pipeline's decisions are not meaningfully worse than those of a frontier model on this task.

Table 6: Agreement of the claim-processing pipeline (Gemma3-27B), GPT-5.5, and human labels on deduplication and conflict detection decisions.

Stage	**Pipeline vs. human** ($n = 10$)	**GPT-5.5 vs. human** ($n = 10$)	**GPT-5.5 vs. pipeline** ($n = 100$)
Deduplication	9/10 (90%)	7/10 (70%)	62/100 (62%)
Conflict detection	7/10 (70%)	9/10 (90%)	63/100 (63%)
Overall	**16/20 (80%)**	**16/20 (80%)**	125/200 (63%)

Residual Error after Filtering. To measure the error that survives at full scale rather than in a sample, we run deberta-v2-xlarge-mnli (He et al., 2021) over the entire filtered claim set for Gemma3-27B on StrategyQA. We check for duplicate pairs that the deduplication stage should have removed and for flagged conflict edges that are not entailment-level contradictions. Table 7 shows that residual error rates are below 2%, which is unlikely to distort the conflict graph or the downstream uncertainty scores.

Table 7: Residual error of the claim pipeline after filtering, measured by an NLI model over all claim pairs (Gemma3-27B, StrategyQA).

Stage	**Metric**	**Count**	**Percentage**
Deduplication	Total pairs	187,805	–
	Residual duplicates	1,704	**0.91%**
Conflict detection	Total pairs	6,343	–
	False positives	119	**1.88%**

Failure Cases. We show two representative failure cases of graph construction. (1) *False conflict edge.* "Julia died young and without children" and "Julius Caesar had only one known biological child, Julia Caesar" were flagged as conflicting, although both claims can hold. The error stems from the LLM conflating two different people named Julia. (2) *Over-merge in deduplication.* "As of 2023, The Lord of the Rings had sold over 150 million copies" was merged into "The Lord of the Rings has sold over 150 million copies", dropping the date qualifier and reducing the claim count by one. The temporal information is lost from the claim set, although the agent can still access the original observation during inference.

H Full Main Results

Table 8 reports the exact values behind Figure 4 for HotpotQA and StrategyQA.

Table 8: **Full results on HotpotQA and StrategyQA** (values behind Figure 4). We report AUROC and AUTAC using $K=10$ trajectories and 10 samples per step; MemUQ combines Perplexity with each centrality metric. Baseline applies existing UQ methods directly to trajectory final answers; Mem_Ext applies the same methods to agent memory. Best results are **bolded**; second-best are underlined. † Customized implementation based on Duan et al. (2025).

		Model	Gemma3-27B		Qwen3-30B		GPT-4.1-Nano	
		Metric	AUROC	AUTAC	AUROC	AUTAC	AUROC	AUTAC
HotpotQA	Baseline	Perplexity	0.5366	0.5482	0.6196	0.5670	0.5592	0.5007
		NE	0.5178	0.5356	0.6245	0.5315	0.5639	0.4993
		SE	0.4938	0.5448	0.6276	0.5642	0.5129	0.4557
		SD	0.4622	0.5304	0.5981	0.5262	0.5755	0.4593
		Deg_E	0.4902	0.5440	0.5902	0.5718	0.5098	0.4894
	UProp†		0.6116	0.5251	0.6846	0.5695	0.5611	0.5397
	Mem_Ext	Perplexity	0.6050	0.5475	0.6638	0.5696	0.6141	0.5530
		NE	0.6185	0.5403	0.6850	0.5736	0.6087	0.5504
		SE	0.5708	0.5078	0.5509	0.5392	0.5805	0.5665
		SD	0.5091	0.4757	0.5560	0.5342	0.5549	0.5501
		Deg_E	0.5597	0.5067	0.6057	0.5611	0.5909	0.5652
	MemUQ	C_B	0.6288	0.5414	0.6888	0.5830	0.6425	0.5553
		C_E	0.6747	0.5663	0.6903	0.5883	**0.6589**	0.5670
		C_C	0.6918	0.5773	0.6969	0.5924	0.6585	**0.5693**
		C_{PR}	**0.6971**	**0.5803**	**0.7048**	**0.5937**	0.6586	0.5679
StrategyQA	Baseline	Perplexity	0.4951	0.7424	0.5439	0.7819	0.5034	0.6585
		NE	0.4912	0.7408	0.5470	0.7839	0.4966	0.6533
		SE	0.5062	0.7489	0.5600	0.7636	0.5045	0.6729
		SD	0.4560	0.7208	0.4567	0.7175	0.5210	0.6814
		Deg_E	0.4933	0.7409	0.5579	0.7831	0.5634	0.6723
	UProp†		0.5289	0.7425	0.6078	0.7848	0.5442	0.6916
	Mem_Ext	Perplexity	0.5262	0.7468	0.6070	0.7870	0.5541	0.6924
		NE	0.5247	0.7442	0.6086	0.7857	0.5668	0.6922
		SE	0.5461	0.7426	0.5984	0.7899	0.5471	**0.7084**
		SD	0.4428	0.7127	0.5506	0.7755	0.5350	0.7039
		Deg_E	0.5418	0.7457	0.6402	0.8117	0.5665	0.7009
	MemUQ	C_B	0.5972	0.7475	0.6215	0.8054	**0.5931**	0.6987
		C_E	**0.6122**	0.7480	0.6460	0.8164	0.5860	0.6932
		C_C	0.6029	0.7465	0.6510	0.8169	0.5849	0.6932
		C_{PR}	0.6051	**0.7504**	**0.6539**	**0.8199**	0.5911	0.6945

I Full Intrinsic Methods Combination

In the main text (Figure 4), we present MemUQ results combining only Perplexity with graph centrality metrics to conserve space, alongside Baseline and Mem_Ext rows. Here we provide comprehensive results showing all combinations of intrinsic uncertainty measures (Perplexity, Normalized Entropy, Semantic Entropy, Semantic Density, and Degree of Disagreement) with each graph centrality metric (C_B, C_E, C_C, C_{PR}). Each table includes the naive agent Baseline (existing UQ methods applied directly to ReAct-style trajectories) and Mem_Ext (UQ methods applied to paragraph-style memory) for reference. Tables 9 and 10 report AUROC and AUTAC scores on HotpotQA and StrategyQA, respectively.

Consistent Improvements Across Intrinsic Methods. The results demonstrate that MemUQ's graph-based extrinsic uncertainty consistently enhances performance regardless of which intrinsic uncertainty method serves as the base. On HotpotQA, combining any intrinsic method with PageRank centrality (C_{PR}) yields substantial improvements over the corresponding naive agent baseline. For example, with Gemma3-27B, the naive agent SE baseline achieves 0.4938 AUROC, while SE+C_{PR} achieves 0.6671 AUROC, a 35.1% relative improvement. Similarly, on Qwen3-30B, SE+C_{PR} reaches 0.7292 AUROC compared to the naive agent SE baseline of 0.6276, representing a 16.2%

Table 9: **Uncertainty quantification results on HotPotQA.** We report AUROC and AUTAC using 10 trajectories across three models. Baseline refers to naive agents (existing UQ methods applied directly to ReAct-style agent trajectories). Mem_Ext refers to UQ methods applied to paragraph-style agent memory. MemUQ combines intrinsic uncertainty measures (Perplexity, NE, SE, SD, or Deg_E) with graph centrality metrics (C_B, C_E, C_C, C_{PR}). Best results within each MemUQ block are **bolded**. † Customized implementation based on Duan et al. (2025); no public code is available.

	Model		Gemma3-27B		Qwen3-30B		GPT-4.1-Nano	
	Metric		AUROC	AUTAC	AUROC	AUTAC	AUROC	AUTAC
HotpotQA	**Baseline**	Perplexity	0.5366	0.5482	0.6196	0.5670	0.5592	0.5007
		NE	0.5178	0.5356	0.6245	0.5315	0.5639	0.4993
		SE	0.4938	0.5448	0.6276	0.5642	0.5129	0.4557
		SD	0.4622	0.5304	0.5981	0.5262	0.5755	0.4593
		Deg_E	0.4902	0.5440	0.5902	0.5718	0.5098	0.4894
	UProp†		0.6116	0.5251	0.6846	0.5695	0.5611	0.5397
	Mem_Ext	Perplexity	0.6050	0.5475	0.6638	0.5696	0.6141	0.5530
		NE	0.6185	0.5403	0.6850	0.5736	0.6087	0.5504
		SE	0.5708	0.5078	0.5509	0.5392	0.5805	0.5665
		SD	0.5091	0.4757	0.5560	0.5342	0.5549	0.5501
		Deg_E	0.5597	0.5067	0.6057	0.5611	0.5909	0.5652
	Perplexity	C_B	0.6288	0.5414	0.6888	0.5830	0.6425	0.5553
		C_E	0.6747	0.5663	0.6903	0.5883	**0.6589**	0.5670
		C_C	0.6918	0.5773	0.6969	0.5924	0.6585	**0.5693**
		C_{PR}	**0.6971**	**0.5803**	**0.7048**	**0.5937**	0.6586	0.5679
	NE	C_B	0.6230	0.5391	0.6850	0.5799	0.6443	0.5527
		C_E	0.6702	0.5637	0.6853	0.5860	**0.6636**	0.5698
		C_C	0.6870	0.5767	0.6922	0.5916	0.6634	**0.5707**
		C_{PR}	**0.6927**	**0.5773**	**0.7006**	**0.5927**	0.6635	0.5678
	SE	C_B	0.6254	0.5337	0.7209	0.6023	0.6295	0.5478
		C_E	0.6576	0.5593	0.7235	0.6035	**0.6411**	0.5553
		C_C	**0.6674**	**0.5668**	0.7279	**0.6063**	0.6403	0.5557
		C_{PR}	0.6671	0.5664	**0.7292**	0.6061	0.6396	**0.5566**
	SD	C_B	0.6104	0.5338	**0.6539**	0.5741	0.6360	0.5509
		C_E	0.6239	0.5407	0.6469	0.5687	**0.6498**	**0.5587**
		C_C	0.6268	**0.5428**	0.6523	**0.5745**	0.6494	0.5573
		C_{PR}	**0.6276**	0.5427	0.6511	0.5717	0.6488	0.5566
	Deg_E	C_B	0.6279	0.5358	0.7278	0.6072	0.6339	0.5469
		C_E	0.6472	0.5503	0.7276	0.6067	**0.6443**	0.5584
		C_C	**0.6515**	0.5560	0.7312	**0.6085**	0.6440	**0.5586**
		C_{PR}	0.6514	**0.5564**	**0.7325**	0.6079	0.6429	0.5582

improvement. These gains persist across all model architectures, confirming that our claim-level conflict analysis captures complementary uncertainty signals that intrinsic methods alone cannot detect.

PageRank and Eigenvector Centrality Are the Most Robust. Across the 30 intrinsic-method blocks (five intrinsic methods, three models, two datasets), PageRank (C_{PR}) achieves the best AUROC in 11 blocks and eigenvector centrality (C_E) in 10, compared to 5 for betweenness and 4 for closeness. PageRank is the best centrality in 7 of the 10 Qwen3-30B blocks, while eigenvector centrality is the best in 5 of the 10 blocks for both Gemma3-27B and GPT-4.1-Nano. Both metrics are recursive: a claim receives high uncertainty not merely because it has conflicts, but because it conflicts with other claims that are themselves central to the conflict structure. On StrategyQA with Qwen3-30B, the SD+C_{PR} combination achieves 0.6578 AUROC and 0.8199 AUTAC, substantially outperforming the naive agent SD baseline (0.4567 AUROC, 0.7175 AUTAC).

Intrinsic Method Choice Affects Optimal Centrality Pairing. Certain intrinsic-centrality pairings show notable synergies. On Qwen3-30B for HotpotQA, Degree of Disagreement (Deg_E) and

Table 10: **Uncertainty quantification results on StrategyQA.** We report AUROC and AUTAC using 10 trajectories across three models. Baseline refers to naive agents (existing UQ methods applied directly to ReAct-style agent trajectories). Mem_Ext refers to UQ methods applied to paragraph-style agent memory. MemUQ combines intrinsic uncertainty measures (Perplexity, NE, SE, SD, or Deg_E) with graph centrality metrics (C_B, C_E, C_C, C_{PR}). Best results within each MemUQ block are **bolded**. † Customized implementation based on Duan et al. (2025); no public code is available.

	Model		Gemma3-27B		Qwen3-30B		GPT-4.1-Nano	
	Metric		AUROC	AUTAC	AUROC	AUTAC	AUROC	AUTAC
StrategyQA	**Baseline**	Perplexity	0.4951	0.7424	0.5439	0.7819	0.5034	0.6585
		NE	0.4912	0.7408	0.5470	0.7839	0.4966	0.6533
		SE	0.5062	0.7489	0.5600	0.7636	0.5045	0.6729
		SD	0.4560	0.7208	0.4567	0.7175	0.5210	0.6814
		Deg_E	0.4933	0.7409	0.5579	0.7831	0.5634	0.6723
	UProp†		0.5289	0.7425	0.6078	0.7848	0.5442	0.6916
	Mem_Ext	Perplexity	0.5262	0.7468	0.6070	0.7870	0.5541	0.6924
		NE	0.5247	0.7442	0.6086	0.7857	0.5668	0.6922
		SE	0.5461	0.7426	0.5984	0.7899	0.5471	0.7084
		SD	0.4428	0.7127	0.5506	0.7755	0.5350	0.7039
		Deg_E	0.5418	0.7457	0.6402	0.8117	0.5665	0.7009
	Perplexity	C_B	0.5972	0.7475	0.6215	0.8054	**0.5931**	**0.6987**
		C_E	**0.6122**	0.7480	0.6460	0.8164	0.5860	0.6932
		C_C	0.6029	0.7465	0.6510	0.8169	0.5849	0.6932
		C_{PR}	0.6051	**0.7504**	**0.6539**	**0.8199**	0.5911	0.6945
	NE	C_B	0.5969	0.7478	0.6193	0.8051	**0.5943**	**0.6989**
		C_E	**0.6097**	0.7492	0.6449	0.8171	0.5869	0.6953
		C_C	0.6012	0.7472	0.6494	0.8169	0.5853	0.6931
		C_{PR}	0.6033	**0.7503**	**0.6519**	**0.8181**	0.5920	0.6970
	SE	C_B	0.5777	0.7438	0.6203	0.8102	**0.5894**	**0.7016**
		C_E	**0.5797**	**0.7459**	0.6405	0.8173	0.5823	0.6938
		C_C	0.5718	0.7432	**0.6441**	0.8185	0.5825	0.6957
		C_{PR}	0.5718	0.7442	0.6439	**0.8198**	0.5871	0.6974
	SD	C_B	**0.5618**	**0.7378**	0.6441	0.8148	**0.5810**	**0.6930**
		C_E	**0.5618**	0.7352	0.6575	0.8195	0.5759	0.6907
		C_C	0.5607	0.7357	**0.6601**	**0.8200**	0.5733	0.6898
		C_{PR}	0.5588	0.7375	0.6578	0.8199	0.5775	0.6928
	Deg_E	C_B	0.5611	0.7388	0.6376	0.8111	0.5858	**0.6997**
		C_E	**0.5651**	**0.7413**	0.6503	0.8199	0.5828	0.6965
		C_C	0.5615	0.7411	0.6530	0.8201	0.5814	0.6963
		C_{PR}	0.5578	0.7396	**0.6540**	**0.8207**	**0.5866**	0.6987

Semantic Entropy (SE) paired with C_{PR} achieve the strongest results in the grid (0.7325 and 0.7292 AUROC, respectively), likely because their semantic comparison of samples complements the conflict graph's structural analysis. In contrast, Semantic Density (SD) yields weaker combinations on Gemma3-27B for HotpotQA (at most 0.6276 AUROC, versus 0.6971 with Perplexity). These patterns suggest that practitioners may benefit from selecting centrality metrics based on their chosen intrinsic uncertainty estimator, with PageRank as a reliable default.

Robustness Across Task Types. The improvements transfer from multi-hop factual retrieval (HotpotQA) to implicit reasoning tasks (StrategyQA). On StrategyQA, where questions require combining world knowledge across implicit reasoning chains, MemUQ still achieves meaningful gains. For example, with GPT-4.1-Nano, the NE+C_{PR} combination achieves 0.5920 AUROC compared to the naive agent NE baseline of 0.4966, while also improving AUTAC from 0.6533 to 0.6970. These results confirm that memory-centric uncertainty quantification generalizes across different reasoning paradigms.

Table 11: **Ablation study: MemUQ at K=5 trajectories versus Mem_Ext at K=10, with Gemma3-27B.** We report AUROC on HotpotQA and StrategyQA. This is a cross-budget comparison by design: MemUQ uses half the trajectories (K=5), while the Mem_Ext rows are the full-budget (K=10) results from Table 8. Mem_Ext methods include Perplexity, Normalized Entropy, Semantic Entropy (SE), Semantic Density (SD), and Degree of Disagreement (Deg_E). MemUQ combines Perplexity with centrality metrics: Betweenness (C_B), Eigenvector (C_E), Closeness (C_C), and PageRank (C_{PR}). Best results are **bolded**; second-best are underlined.

	Method (budget)	AUROC
HotpotQA	Perplexity (K=10)	0.6050
	Norm. Entropy (K=10)	0.6185
	SE (K=10)	0.5708
	SD (K=10)	0.5091
	Deg_E (K=10)	0.5597
	MemUQ C_B (K=5)	0.6312
	MemUQ C_E (K=5)	0.6445
	MemUQ C_C (K=5)	<u>0.6519</u>
	MemUQ C_{PR} (K=5)	**0.6630**
StrategyQA	Perplexity (K=10)	0.5262
	Norm. Entropy (K=10)	0.5247
	SE (K=10)	0.5461
	SD (K=10)	0.4428
	Deg_E (K=10)	0.5418
	MemUQ C_B (K=5)	**0.5689**
	MemUQ C_E (K=5)	0.5657
	MemUQ C_C (K=5)	0.5642
	MemUQ C_{PR} (K=5)	<u>0.5674</u>

J Performance with Fewer Trajectories

We investigate the sensitivity of our framework to the number of sampled trajectories by evaluating MemUQ with $K = 5$ trajectories instead of the default $K = 10$. This ablation asks whether MemUQ at half the sampling cost still holds up against baselines given the full budget, which is important for practical deployment where sampling costs must be balanced against performance. Table 11 therefore evaluates MemUQ at $K = 5$ while holding Mem_Ext at the $K = 10$ setting; the Mem_Ext rows are identical to those in Table 8.

As shown in Table 11, MemUQ at $K = 5$ outperforms every Mem_Ext method at $K = 10$ with all four centrality metrics. On HotpotQA, PageRank achieves 0.6630 AUROC at $K = 5$ compared to 0.6185 for the best Mem_Ext method at $K = 10$ (Normalized Entropy), a 7.2% relative improvement. On StrategyQA, PageRank attains 0.5674 AUROC versus 0.5461 for the best Mem_Ext method at $K = 10$ (Semantic Entropy), a 3.9% relative improvement. Relative to its own $K = 10$ results, MemUQ retains 95% (HotpotQA) and 94% (StrategyQA) of its PageRank AUROC with half the trajectories. These results indicate that claim-level conflict analysis captures meaningful uncertainty signals even with limited trajectory samples, leaving headroom to reduce the deployment cost below the $K = 10$ budget.

K Calculating Intrinsic and Extrinsic Uncertainty

In this section, we provide the complete algorithmic specifications for uncertainty quantification in our framework. We present four algorithms: (1) the baseline approach for computing extrinsic uncertainty via trajectory-level memory variance (Algorithm 1), (2) our proposed claim-level conflict analysis for extrinsic uncertainty (Algorithm 2), (3) intrinsic uncertainty estimation via step-level memory sampling (Algorithm 3), and (4) the combined uncertainty aggregation procedure (Algorithm 4).

Algorithm 1 Baseline Extrinsic Uncertainty Quantification via Trajectory-Level Memory Variance

Require: Query x, set of K pre-collected trajectories $\{\tau^{(k)}\}_{k=1}^{K}$, single-step UQ method ϕ
Ensure: Extrinsic uncertainty score $U_{\text{ext}}(x)$
Initialize trajectory uncertainties $\mathcal{U}_{\text{traj}} \leftarrow []$
for $k = 1$ to K **do**
  // Extract memory states from trajectory
  $\mathcal{M}^{(k)} \leftarrow \{m_1^{(k)}, m_2^{(k)}, \ldots, m_{T_k}^{(k)}\}$ {T_k step memories from $\tau^{(k)}$}
  // Compute trajectory-level uncertainty
  $u^{(k)} \leftarrow \phi(\mathcal{M}^{(k)})$ {Apply single-step UQ to T_k memories}
  Append $u^{(k)}$ to $\mathcal{U}_{\text{traj}}$
end for
// Aggregate across trajectories
$U_{\text{ext}}(x) \leftarrow \frac{1}{K}\sum_{k=1}^{K} u^{(k)}$
return $U_{\text{ext}}(x)$

Algorithm 2 Extrinsic Uncertainty Quantification via Claim-Level Conflict Analysis

Require: Query x, set of K pre-collected trajectories $\{\tau^{(k)}\}_{k=1}^{K}$, centrality function $C(\cdot)$
Ensure: Extrinsic uncertainty score $U_{\text{ext}}(x)$
// Phase 1: Claim Extraction from Trajectories
Initialize claim multiset $\mathcal{A} \leftarrow \emptyset$
for $k = 1$ to K **do**
  Extract memory states $\{m_1^{(k)}, \ldots, m_{T_k}^{(k)}\}$ from $\tau^{(k)}$
  for $t = 1$ to T_k **do**
    $\mathcal{A} \leftarrow \mathcal{A} \cup \textsc{ExtractClaims}(m_t^{(k)})$ {Collect atomic claims}
  end for
end for
// Phase 2: Claim Deduplication and Conflict Graph Construction
$\mathcal{V} \leftarrow \textsc{FilterDuplicates}(\mathcal{A})$ {Unique claim set}
Initialize edge set $E \leftarrow \emptyset$
for each claim $c_i \in \mathcal{V}$ **do**
  $\mathcal{E}_i \leftarrow \textsc{DetectConflicts}(c_i, \mathcal{V})$ {LLM identifies all conflicts for c_i}
  for each $c_j \in \mathcal{E}_i$ **do**
    $E \leftarrow E \cup \{(c_i, c_j)\}$
  end for
end for
Construct conflict graph $G = (\mathcal{V}, E)$
// Phase 3: Uncertainty Score Computation
for each claim $c_i \in \mathcal{V}$ **do**
  Compute centrality: $C(c_i) \leftarrow \textsc{GraphCentrality}(G, c_i)$ {PageRank, Betweenness, etc.}
  Compute occurrence count: $n_i \leftarrow \sum_{k=1}^{K} \mathbf{1}[c_i \in \tau^{(k)}]$
end for
Compute normalized frequencies: $f(c_i) \leftarrow n_i / \sum_{j=1}^{|\mathcal{V}|} n_j$ for all $c_i \in \mathcal{V}$
Normalize centralities: $\bar{C}(c_i) \leftarrow C(c_i) / \sum_{j=1}^{|\mathcal{V}|} C(c_j)$ for all $c_i \in \mathcal{V}$ {uniform $1/|\mathcal{V}|$ if all are zero}
// Phase 4: Aggregation to Query-Level Score
$U_{\text{ext}}(x) \leftarrow \sum_{c_i \in \mathcal{V}} \bar{C}(c_i) \cdot (1 - f(c_i))$ {$\in [0, 1]$}
return $U_{\text{ext}}(x)$

L MEMORY TOKEN UTILIZATION

Table 12 reports the final-step memory size (in tokens) across datasets and models. Memory usage is far below the configured limit of 5,000 tokens, typically under 5% of capacity, so varying the memory length parameter M does not affect task accuracy or UQ quality.

Algorithm 3 Intrinsic Uncertainty Quantification via Step-Level Memory Sampling

Require: Query x, pre-collected memory samples $\{\mathcal{M}_t\}_{t=1}^{T}$ where $\mathcal{M}_t = \{m_t^{(1)}, \ldots, m_t^{(S)}\}$, single-step UQ method ϕ
Ensure: Intrinsic uncertainty score $U_{\text{int}}(x)$
Initialize step uncertainties $\mathcal{U}_{\text{steps}} \leftarrow []$
for $t = 1$ to T **do**
  // Compute step-level uncertainty over memory samples
  $u_t \leftarrow \phi(\mathcal{M}_t)$ {Apply single-step UQ to S samples at step t}
  Append u_t to $\mathcal{U}_{\text{steps}}$
end for
// Aggregate across all interaction steps
$U_{\text{int}}(x) \leftarrow \frac{1}{T}\sum_{t=1}^{T} u_t$
return $U_{\text{int}}(x)$

Algorithm 4 Combined Uncertainty Quantification for LLM Agents

Require: Query x, set of K pre-collected trajectories $\{\tau^{(k)}\}_{k=1}^{K}$, pre-collected memory samples $\{\mathcal{M}_t\}_{t=1}^{T}$, centrality function $\mathcal{C}(\cdot)$, single-step UQ method ϕ
Ensure: Total uncertainty score $U_{\text{total}}(x)$
// Compute intrinsic uncertainty (shared by both methods)
$U_{\text{int}}(x) \leftarrow \textsc{IntrinsicUncertainty}(x, \{\mathcal{M}_t\}_{t=1}^{T}, \phi)$ {Algorithm 3}

// Baseline: Extrinsic Uncertainty Adopting Single-step UQ Methods
$U_{\text{ext}}^{\text{base}}(x) \leftarrow \textsc{BaselineExtrinsic}(x, \{\tau^{(k)}\}_{k=1}^{K}, \phi)$ {Algorithm 1}
$U_{\text{total}}^{\text{base}}(x) \leftarrow \frac{1}{2}\left(U_{\text{int}}(x) + U_{\text{ext}}^{\text{base}}(x)\right)$

// MemUQ: Claim-Level Extrinsic Uncertainty
$U_{\text{ext}}^{\text{claim}}(x) \leftarrow \textsc{ClaimLevelExtrinsic}(x, \{\tau^{(k)}\}_{k=1}^{K}, \mathcal{C})$ {Algorithm 2}
$U_{\text{total}}^{\text{ours}}(x) \leftarrow \frac{1}{2}\left(U_{\text{int}}(x) + U_{\text{ext}}^{\text{claim}}(x)\right)$

return $U_{\text{total}}^{\text{base}}(x)$ or $U_{\text{total}}^{\text{ours}}(x)$

M Memory Representation Comparison

To isolate the contribution of the conflict graph from the memory representation format, we compare three memory-level UQ strategies: Mem_Ext applies existing UQ methods to paragraph-style memory, Mem_Claim applies the same methods to claim-style memory (without the conflict graph), and MemUQ adds graph-based extrinsic uncertainty on top of claim-style memory. As shown in Table 13, Mem_Claim and Mem_Ext perform comparably, indicating that the claim-style memory format alone does not improve UQ. The gains of MemUQ over both variants confirm that the conflict graph and centrality-based scoring are the primary sources of improvement.

Table 12: Final-step memory token utilization per query.

Dataset	Model	Mean	Median	Std
StrategyQA	Gemma3-27B	228.32	217	87.62
	Qwen3-30B	130.53	120	70.62
	GPT-4.1-Nano	118.74	114	69.35
HotpotQA	Gemma3-27B	216.81	211	87.06
	Qwen3-30B	134.07	122	87.62
	GPT-4.1-Nano	98.87	81	69.68

Table 13: **Comparison between paragraph-style memory (Mem_Ext), claim-based memory (Mem_Claim), and MemUQ.** We report AUROC and AUTAC on HotpotQA and StrategyQA using 10 trajectories for Gemma3-27B, Qwen3-30B, and GPT-4.1-Nano. MemUQ combines Perplexity with different centrality metrics. Best results are **bolded**; second-best are underlined.

		Model	Gemma3-27B		Qwen3-30B		GPT-4.1-Nano	
		Metric	AUROC	AUTAC	AUROC	AUTAC	AUROC	AUTAC
HotpotQA	Mem_Ext	Perplexity	0.6050	0.5475	0.6638	0.5696	0.6141	0.5530
		NE	0.6185	0.5403	0.6850	0.5736	0.6087	0.5504
		SE	0.5708	0.5078	0.5509	0.5392	0.5805	0.5665
		SD	0.5091	0.4757	0.5560	0.5342	0.5549	0.5501
		Deg_E	0.5597	0.5067	0.6057	0.5611	0.5909	0.5652
	Mem_Claim	Perplexity	0.5889	0.5169	0.6831	0.5928	0.6141	0.5530
		NE	0.5868	0.5147	0.6755	0.5877	0.6087	0.5504
		SE	0.5693	0.4979	0.5748	0.5612	0.6141	0.5468
		SD	0.5288	0.5242	0.5840	0.5241	0.5791	0.5446
		Deg_E	0.6016	0.5205	0.6293	**0.6053**	0.6163	0.5420
	MemUQ	C_B	0.6288	0.5414	0.6888	0.5830	0.6425	0.5553
		C_E	0.6747	0.5663	0.6903	0.5883	**0.6589**	0.5670
		C_C	0.6918	0.5773	0.6969	0.5924	0.6585	**0.5693**
		C_{PR}	**0.6971**	**0.5803**	**0.7048**	0.5937	0.6586	0.5679
StrategyQA	Mem_Ext	Perplexity	0.5262	0.7468	0.6070	0.7870	0.5541	0.6924
		NE	0.5247	0.7442	0.6086	0.7857	0.5668	0.6922
		SE	0.5461	0.7426	0.5984	0.7899	0.5471	**0.7084**
		SD	0.4428	0.7127	0.5506	0.7755	0.5350	0.7039
		Deg_E	0.5418	0.7457	0.6402	0.8117	0.5665	0.7009
	Mem_Claim	Perplexity	0.5298	0.7207	0.6141	0.8015	0.5541	0.6924
		NE	0.5303	0.7198	0.6094	0.8002	0.5668	0.6922
		SE	0.5424	0.7302	0.5929	0.8005	0.5680	0.7019
		SD	0.4685	0.7296	0.5782	0.8026	0.5645	0.7058
		Deg_E	0.5365	0.7315	0.6222	0.8080	0.5620	0.6993
	MemUQ	C_B	0.5972	0.7475	0.6215	0.8054	**0.5931**	0.6987
		C_E	**0.6122**	0.7480	0.6460	0.8164	0.5860	0.6932
		C_C	0.6029	0.7465	0.6510	0.8169	0.5849	0.6932
		C_{PR}	0.6051	**0.7504**	**0.6539**	**0.8199**	0.5911	0.6945