

Micro-Act: Mitigate Knowledge Conflict in Question Answering via Actionable Self-Reasoning

Nan Huo¹, Jinyang Li¹, Bowen Qin^{2*}, Ge Qu¹, Xiaolong Li¹, Xiaodong Li³,
Chenhao Ma⁴, Reynold Cheng^{1*}

¹The University of Hong Kong, ²BAAI

³Xiamen University

⁴The Chinese University of Hong Kong, Shenzhen

huonan@connect.hku.hk, bwqin@baai.ac.cn, ckcheng@cs.hku.hk

Abstract

Retrieval-Augmented Generation (RAG) systems commonly suffer from **Knowledge Conflicts**, where retrieved external knowledge contradicts the inherent, parametric knowledge of large language models (LLMs). It adversely affects performance on downstream tasks such as question answering (QA). Existing approaches often attempt to mitigate conflicts by directly comparing two knowledge sources in a side-by-side manner, but this can overwhelm LLMs with extraneous or lengthy contexts, ultimately hindering their ability to identify and mitigate inconsistencies. To address this issue, we propose **MICRO-ACT**, a framework with a hierarchical action space that automatically perceives context complexity and adaptively decomposes each knowledge source into a sequence of fine-grained comparisons. These comparisons are represented as actionable steps, enabling reasoning beyond the superficial context. Through extensive experiments on five benchmark datasets, MICRO-ACT consistently achieves significant increase in QA accuracy over state-of-the-art baselines across all 5 datasets and 3 conflict types, especially in temporal and semantic types where all baselines fail significantly. More importantly, MICRO-ACT exhibits robust performance on non-conflict questions simultaneously, highlighting its practical value in real-world RAG applications. Code can be found at <https://github.com/Nan-Huo/Micro-Act>.

1 Introduction

Recent advances in large language models (LLMs) have revolutionized natural language processing with their ability to understand and respond to diverse user queries (Chang et al., 2024; Zhao et al., 2023). However, relying solely on parametric

* Corresponding authors are Reynold Cheng and Bowen Qin.

Q: What position does Paul Eugène Gillon currently hold?

Retrieved Evidence: ... In 2010, Gillon was appointed as a deputy member of the Parliament of Norway **Conflict Evidence**

Correct Answer: General Secretary of the Prefecture of Creuse.

ICL-based Solutions:

(a) Generic Reasoning: (e.g., COT)

Let's think step-by-step to answer the question:

1. Gillon was appointed as a deputy member of the Parliament
2. **Answer:** the deputy member of the Parliament. **✗**

(b) Generation-aided Reasoning: (e.g., GKP)

Step 1: Generate some knowledge about Paul Eugène Gillon:

Recently, Eugène Gillon has been appointed as the General Secretary of the Prefecture of Creuse. In French ...

Step 2: Answer with **retrieved evidence** and **knowledge generated**:

Knowledge Generated by LLM:
Recently, Eugène Gillon has been appointed as the General Secretary of the Prefecture of Creuse. In French ...

Retrieved Evidence:
... In 2010, Gillon was appointed as a deputy member of the Parliament of Norway

Two evidence have **conflict**. **Answer:** I don't know. **✗**

(c) Micro-Act: (Ours)

Step 1: Locate fine-grained evidence-LLM conflict points

1. ASSERT(**General Secretary**, **Deputy member**) → **Conflict!**
2. ASSERT(**French**, **Norway**) → **Conflict!**
3. ASSERT(**Recently**, **2010**) → **Conflict!**

Step 2: Reason & Answer based on located conflict points.

Answer: General Secretary of the Prefecture of Creuse. **✓**

Figure 1: An illustration of QA under knowledge conflict via a real example. The detailed illustration can be found in Figure 7. (a) refers to the generic reasoning methods that reason on merely retrieved context. (b) refers to generation-aided reasoning methods aided by self-generated knowledge. (c) refers to our proposed MICRO-ACT.

knowledge often leads to hallucinations and factual errors, especially when dealing with domain-specific queries or rapidly evolving information. To enhance the reliability and factual accuracy of LLM responses, retrieval-augmented generation (RAG) has emerged as a promising paradigm that grounds LLM reasoning with evidence from external knowledge sources (Guu et al., 2020a; Lewis et al., 2020; Chen et al., 2024; Ren et al., 2023).

Despite the promise of RAG, a critical challenge emerges when retrieved information contradicts

the pre-trained parametric knowledge of LLMs, a phenomenon known as *knowledge conflict* (Wang et al., 2024; Jin et al., 2024a). Such conflicts arise frequently because retrieval systems may introduce noisy, outdated, or even incorrect information (Su et al., 2024; Wang et al., 2024; Shi et al., 2024a; Jin et al., 2024a), which significantly undermines their potential benefits and raises concerns about their practical deployment in downstream tasks such as question answering (QA).

Prior works addressing knowledge conflicts fall into two distinct categories. The first focuses on specialized fine-tuning techniques (Yuan et al., 2024; Shi et al., 2024a; Jin et al., 2024b). The second leverages In-Context Learning (ICL), which can adapt to new requirements or tasks by providing relevant instructions or examples, reducing the effort required for re-training or continual training. Within the ICL-based category, approaches can be further divided into two types: generic reasoning methods that rely solely on retrieved context, as shown in Figure 1(a), and generation-aided reasoning methods that generate the pre-trained parametric knowledge of LLMs for explicit knowledge comparison with retrieved knowledge (Liu et al., 2022), as illustrated in Figure 1(b). However, these ICL-based methods face three critical limitations: (1) heavy reliance on manually crafted instructions limits cross-domain adaptability; (2) side-by-side comparison fails to capture conflicts at different granularity levels, making LLMs vulnerable to irrelevant contexts (Mirzadeh et al., 2024); and (3) those methods meticulously design prompts to handle knowledge conflict, which assumes that knowledge conflict already exists. This would probably lead to a negative impact on performance in conflict-free scenarios, which are common in real-world applications, raising concerns about their practical reliability.

To address these limitations, we propose **MICRO-ACT**, whose core innovation is its ability to dynamically adjust granularity through decomposition action: (1) at model level, it automatically perceives input complexity preferences for different LLMs, and (2) at action level, it detects context granularity of each action and flexibly makes adjustment. As illustrated in Figure 1(c), this adaptive approach enables precise conflict detection across different granularity levels and reasoning on the conflicts underneath the superficial context.

Extensive experiments on five widely-used knowledge conflict benchmark datasets grounded

in the QA task (Su et al., 2024; Xie et al., 2024), covering diverse knowledge conflict types (misinformation, temporal, and semantic conflicts) (Su et al., 2024), demonstrate that MICRO-ACT consistently outperforms state-of-the-art baselines. More importantly, MICRO-ACT also maintains competitive performance in conflict-free cases while state-of-the-art baselines cannot, which underscores the strong robustness of MICRO-ACT. Further analysis of complexity detection reveals that MICRO-ACT unlocks the potential of LLMs to perceive complexity and adapt to different environments. And we find an interesting phenomenon called “over-rationalization” which harms conflict resolution and can be mitigated by MICRO-ACT via locating conflict underneath the superficial context. These findings validate the effectiveness and robustness of MICRO-ACT in resolving knowledge conflicts for reliable real-world RAG systems.

2 Related Work

Retrieval-Augmented Generation for QA.

Retrieval-augmented generation (RAG) integrates external knowledge sources with language generation, improving the fidelity and robustness of open-domain QA (Chen et al., 2017; Petroni et al., 2019; Asai et al., 2020; Guu et al., 2020b; Izacard and Grave, 2021a; Lewis et al., 2020; Zhang et al., 2024; Shi et al., 2024b). Subsequent efforts have refined retrieval modules and model architectures to handle diverse knowledge sources and queries more effectively (Karpukhin et al., 2020; Izacard and Grave, 2021b; Mao et al., 2021; Nakano et al., 2021; Shi et al., 2024c; Izacard et al., 2023; Qin et al., 2019, 2018; Conforti et al., 2020; Rezaee et al., 2024). Recent techniques explore dynamic retrieval strategies, domain adaptation, and efficient fine-tuning, further enhancing the adaptability and reliability of RAG frameworks (Ram et al., 2023; Borgeaud et al., 2022; Liu et al., 2024; Zhang et al., 2025).

Knowledge Conflict. Knowledge conflict surfaces when retrieved evidence disagrees with a model’s internal beliefs or when multiple sources present mutually inconsistent information, resulting in ambiguous or flawed outputs (Min et al., 2020; Lewis et al., 2020; Shuster et al., 2021; Wang et al., 2021; Zellers et al., 2019; Tan et al., 2024). This challenge becomes acute in evolving domains (e.g., current events, medicine, science) where timely accuracy is critical (Chen et al., 2021;

Min et al., 2023). Conflicts arise not only between retrieved evidence and parametric knowledge but also among multiple retrieved documents, demanding careful reconciliation to avoid misinformation and preserve trustworthiness (Thorne et al., 2018; Yang et al., 2018; Liang et al., 2023; Gao et al., 2023; Shaier et al., 2024; Pham et al., 2024; Fang et al., 2024).

Solutions for Knowledge Conflict. Proposed solutions generally follow two broad strategies. The first modifies internal model parameters or architectures to accommodate external evidence more consistently (Yuan et al., 2024; Jin et al., 2024b; Shi et al., 2024a), though this often assumes that retrieved information should uniformly override parametric knowledge. The second strategy explicitly identifies and reconciles discrepancies among sources via generation-aided mechanisms or iterative comparison (Wang et al., 2023; Liu et al., 2022). While these methods can reduce factual errors, they often rely on ad-hoc instructions or simple pairwise comparisons that fail to capture subtle conflicts. Recent work underscores the need for more principled, robust approaches that integrate nuanced reasoning and validation, improving both fidelity and explainability in retrieval-augmented QA (Xu et al., 2024).

3 Preliminaries

3.1 RAG for QA

Retrieval-Augmented Generation (RAG) combines a LLM with an external retrieval module. It proceeds in two key phases: a *Retrieval Phase* that returns a set of relevant evidence and a *Generation Phase* where the LLM produces the final answer conditioned on this evidence.

Retrieval Phase. Given a query q , a retrieval function $\mathcal{R}(\cdot)$ returns a set of textual fragments $\mathcal{E} = \{e_1, \dots, e_m\}$, where m is the number of fragments. Each fragment e_i provides potentially relevant information related to q .

Generation Phase. Let \mathcal{M}_Θ be the LLM parameterized by Θ . We define the parametric knowledge for the query q as:

$$K^p(q) = \mathcal{M}_\Theta(q). \quad (1)$$

We use $K^r(e_i)$ to represent the knowledge contained in each retrieved fragment e_i . The final

answer is produced by conditioning on both the parametric and retrieved knowledge:

$$Ans(q) = \mathcal{M}_\Theta(q | K^p(q), \{K^r(e_i)\}_{i=1}^m). \quad (2)$$

3.2 Knowledge Conflict

A *knowledge conflict* arises when $K^p(q)$ and some $K^r(e_i)$ are factually or logically inconsistent. Formally, there exists at least one $e_i \in \mathcal{E}$ such that:

$$K^p(q) \not\approx K^r(e_i), \quad (3)$$

where $\not\approx$ denotes a factual or logical inconsistency.

Algorithm 1 MICRO-ACT Pseudocode

```

1: Input: query  $q$ , external corpus  $\mathcal{E}$ , LLM  $\mathcal{M}_\Theta$ , turn budget  $N$ 
2: Retrieve:  $K^p \leftarrow \text{ELICIT}(q)$ ,  $K^r \leftarrow \text{RETRIEVE}(\mathcal{E}, q)$ 
3:  $H \leftarrow \emptyset$ 
4: for  $t = 1$  to  $N$  do
5:    $T_t \leftarrow \mathcal{M}_\Theta(\cdot | H)$ ;  $A_t \leftarrow \text{SELECT}(T_t)$ 
6:    $O_t \leftarrow \begin{cases} \text{REASON}(\cdot) \\ \text{ASSERT}(\cdot) \\ \text{DECOMPOSE}(\cdot) \end{cases}$ 
7:    $H \leftarrow H \cup \{T_t, A_t, O_t\}$ 
8:   if  $O_t = \text{conflict} \wedge \text{COMPLEX}$  then
9:      $A_t \leftarrow \text{DECOMPOSE}$   $\triangleright$  force split
10:  end if
11:  if  $\text{SOLVED}(H)$  then break
12:  end if
13: end for
14: Return  $\mathcal{M}_\Theta(\text{ANSWER} | H)$ 

```

4 Methodology

We introduce **MICRO-ACT**, a framework that enables Large Language Models (LLMs) to automatically identify and resolve detailed points of knowledge conflict. MICRO-ACT comprises three key components: (1) a hierarchical action space (Section 4.1), (2) a Reasoning Body (Section 4.2), and (3) Adaptive Granularity and Optimization strategies (Section 5 and 6). The detailed pseudocode of MICRO-ACT can be found in Algorithm 1.

4.1 Hierarchical Action Space

Establishing a well-structured action space allows LLMs to more efficiently invoke planning strategies (Yao et al., 2024). To this end, we define the action space as a structured integration of three key

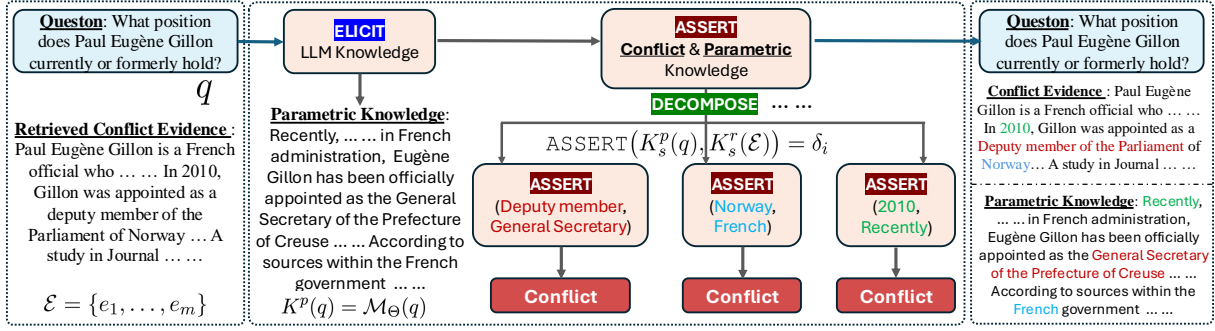


Figure 2: An illustration of handling knowledge conflict in QA task. Actions highlighted with blue color represent navigational actions; Red color represents functional actions; and green color represents the bridging action. "... ..." represents multiple interplayed actions are folded for simplicity.

categories: (1) *navigational actions*, (2) *functional actions*, and (3) *bridging actions*, with the decomposition component serving as the cornerstone for refining context granularity of actions.

Navigational Actions. They focus on exploring the environment and obtaining more information as the prerequisite of effective reasoning (Gu et al., 2024). Navigational actions include eliciting parametric knowledge from the LLM and getting the reasoning path of a QA task based on input context. Let \mathcal{A}_{nav} represent *navigational actions*. Specifically, we formally define the elicit action in Eq. 4.

$$\text{ELICIT}(q) = K^p(q) = \mathcal{M}_{\Theta}(q). \quad (4)$$

And we formally define the action to get the reasoning path \mathcal{P}_K in Eq. 5.

$$\text{REASON}(K) = \mathcal{P}_K = \mathcal{M}_{\Theta}^p(K), \quad (5)$$

where $\mathcal{M}_{\Theta}^p(K)$ represents prompting LLM parametrized by Θ to generate a reasoning path on K . And K is the input knowledge representation either from $K^p(q)$ or from $K^r(\mathcal{E})$.

Functional Actions. They address conflict detection either between retrieved evidence and LLM parametric knowledge or between their reasoning paths generated by the navigational action. Once relevant information is prepared, *functional actions*, denoted by $\mathcal{A}_{\text{func}}$, detect conflict among them. Formally, we define the assert action to implement this logic, which is a conflict verification action and checks the consistency between $K^p(q)$ and a particular $K^r(\mathcal{E})$ in Eq. 6.

$$\text{ASSERT}(K_s^p(q), K_s^r(\mathcal{E})) = \delta_i, \quad (6)$$

where $\delta_i \in \{0, 1\}$. If $\delta_i = 1$, a conflict is detected. And $K_s^p(q) \in K^p(q)$ means $K_s^p(q)$ is a partial knowledge of $K^p(q)$.

Bridging-Action. It is responsible for dynamically optimizing granularity by decomposing actions when needed. A side-by-side assert action may fail to detect subtle conflicts embedded in lengthy, noisy contexts. To address this, we introduce the *decomposition action*, collected in $\mathcal{A}_{\text{micro}}$, which can refine the granularity of analysis. Suppose an $\text{ASSERT}(\cdot)$ action on complex knowledge context is represented as $\text{ASSERT}(K^p(q), K^r(\mathcal{E}))$. A decomposition action can decompose this complex reasoning into smaller, manageable action steps, as shown in Eq. 7.

$$\begin{aligned} \text{DECOMPOSE}(\text{ASSERT}(K_s^p(q), K_s^r(\mathcal{E}))) \\ = \{\text{ASSERT}(K_{s'}^p(q), K_{s'}^r(\mathcal{E})), \dots\}, \end{aligned} \quad (7)$$

where $K_s^p(q) \in K^p(q)$ is a partial knowledge of $K^p(q)$. And $K_{s'}^p(q)$ refers to $K_s^p(K_s^p(q))$, which means the finer-grained partial knowledge of $K_s^p(q)$. Each newly created sub-action deals with a further fragment of the evidence, increasing the likelihood of revealing fine-grained conflicts. It will decompose the action until LLM has enough confidence or reach the max turn limit.

4.2 Reasoning Body

We integrate our hierarchical action space with the ReAct process (Yao et al., 2023) to teach LLM integrate our hierarchical action space to automatically handle knowledge conflicts. At step t , the LLM first produces a thought T_t :

$$T_t \sim \mathcal{M}_{\Theta}(T_t | H_{t-1}), \quad (8)$$

where H_{t-1} is the accumulated history of all thoughts, actions, and observations before step t .

Conditioned on H_{t-1} and the newly generated thought T_t , the model selects an action A_t :

$$A_t \sim \mathcal{M}_{\Theta}(A_t | H_{t-1}, T_t). \quad (9)$$

This action, executed in the changing environment (for example, the knowledge has been decomposed at different granularity), yields an observation O_t : $O_t = \text{Env}(A_t)$. The history is then updated:

$$H_t = H_{t-1} \cup \{T_t, A_t, O_t\}. \quad (10)$$

This iterative process continues, adjusting granularity via decomposition actions whenever subtle conflicts require finer checks. After N steps, the final answer is generated:

$$A_f \sim \mathcal{M}_\Theta(A_f \mid H_N). \quad (11)$$

By dynamically selecting navigational, functional, and decomposition actions, this procedure ensures subtle knowledge conflicts are detected and mitigated, improving the reliability of the final output. An example illustration of this process is shown in Figure 2.

5 Understanding Complexity-Driven Knowledge Decomposition Dynamics

To gain a deeper understanding of how model bridging actions are related to complexity, we follow (Murty et al., 2024) to characterize the distribution of the newly inferred knowledge representation at turn t based on trajectory over previous $t - 1$ turns. Specifically, we define:

$$p_t(K_n) = \sum_{c'} \sum_K p_{\text{model}}(K_n \mid c') p_{\text{verify}}(c' \mid K) p_{t-1}(K), \quad (12)$$

where K is current knowledge representation, K_n is the newly inferred knowledge representation (often obtained by decomposing K), c is the ground-truth knowledge conflict, and c' is a potentially incorrect knowledge conflict identified by the model. p_{model} means the distribution on generate new knowledge. p_{verify} means the distribution on generating conflicts. Detailed derivation can be found in Appendix D.

In this formulation, the term $\sum_K p_{\text{verify}}(c' \mid K)$ increases with the complexity (e.g., longer context, harder domain and etc.), resulting in higher verification probabilities and an increased risk of inaccurate conflict detection. And $\sum_{c'} p_{\text{model}}(K_n \mid c')$ depends on the LLM compatibility. A less capable LLM is more likely to be influenced by erroneous conflicts (c'), thereby requiring further decomposition and pushing $p_k(K_n)$ higher. Section 7.5 shows more details about how these factors drive proactive decomposition across models.

6 Preventing Infinite Decomposition

While hierarchical reasoning is essential for resolving complex conflicts, an unconstrained recursive process could, in principle, keep splitting a context. Building upon the probabilistic dynamics in Eq. (12), we show that MICRO-ACT can prevent infinite decomposition, and we complement this theoretical safeguard with a hard maximum turn budget.

Complexity-Aware Stopping Criterion. Let \mathcal{C}_t denote the latent *complexity score* of the current context after t turns. A decomposition step is triggered only when $\mathcal{C}_t > \tau$, where τ represents the minimum complexity the underlying LLM can handle confidently. Because each decomposition shortens the context length and narrows its semantic scope, the following strict inequality holds:

$$\mathcal{C}_{t+1} < \mathcal{C}_t, \quad \forall t \geq 0. \quad (13)$$

Define

$$T_\tau = \min\{t \mid \mathcal{C}_t \leq \tau\}. \quad (14)$$

By Eq. (13), T_τ is finite, and once reached we have $p_t(K_n) = 0$; no further actions in the DECOMPOSE branch will be sampled. In other words, the process is *self-regularising*: an LLM that already “understands” the context (small \mathcal{C}_t) simply refuses to split it further.

7 Experiments

7.1 Experiment Settings

Datasets. We evaluate MICRO-ACT on five benchmark datasets drawn from two comprehensive collections: ConflictBank and KRE. ConflictBank (Su et al., 2024) provides three specialized datasets targeting distinct conflict types: **misinformation**, **temporal** discrepancies, and **semantic** divergences between retrieved and parametric knowledge. From KRE (Ying et al., 2023), we utilize MuSiQue_KRE and SQuAD_KRE, derived from MuSiQue (Trivedi et al., 2022) and SQuAD v2.0 (Rajpurkar et al., 2018) respectively. These datasets feature multiple-choice questions with generated explanations supporting incorrect choices, creating controlled scenarios for examining reasoning conflicts. Due to the limitation of computational resources, we randomly sampled 3000 data in ConflictBank and 2000 data in KRE dataset across all features, and corrected any errors found.

Prompting	GPT-4o		GPT-4o-mini		LLaMA-3.1-70B		LLaMA-3.1-8B	
	ConflictBank	KRE	ConflictBank	KRE	ConflictBank	KRE	ConflictBank	KRE
Generic Reasoning								
End-to-End QA	5.40	43.80	2.77	31.10	3.07	14.50	2.53	9.55
Few-Shot QA	6.30	45.65	2.83	33.30	3.87	15.20	3.13	10.30
Chain-of-Thought (Wei et al., 2022)	6.43	44.35	3.00	36.50	1.40	29.45	2.13	24.50
Generation-aided Reasoning								
Self-Ask (Press et al., 2023)	3.13	41.45	2.57	24.90	3.33	23.65	2.77	18.65
Comparative (Wang et al., 2023)	11.70	33.95	2.10	23.85	4.53	25.25	3.87	19.80
GKP (Liu et al., 2022)	<u>15.40</u>	<u>55.30</u>	<u>17.53</u>	<u>44.45</u>	<u>15.83</u>	<u>43.55</u>	<u>6.83</u>	<u>32.75</u>
MICRO-ACT (ours)	22.30 (↑ 6.90)	59.50 (↑ 4.20)	26.93 (↑ 9.40)	51.10 (↑ 6.65)	26.50 (↑ 10.67)	54.90 (↑ 11.35)	18.30 (↑ 11.47)	46.60 (↑ 13.85)

Table 1: The average results of Question Answering under Knowledge Conflict on ConflictBank and KRE with GPT-4o-mini, GPT-4o, LLaMA-3.1-70B and LLaMA-3.1-8B. The performance is on average over its sub-datasets. (underline denotes the previous SOTA performance; **bold** denotes the best performance; the improvement (↑) is measured against the previous SOTA performing method.)

Metrics & Models. Following existing knowledge conflict works (Xie et al., 2024; Su et al., 2024; Wang et al., 2023; Shi et al., 2024a), we measure knowledge conflict in QA task by employing QA accuracy as our primary evaluation metric. Specifically, the answer format of QA is multiple-choice. If LLMs successfully resolve knowledge conflict, they will choose the correct answer instead of the negative answer supported by the conflict (wrong) knowledge (Su et al., 2024).

In our experiments, we use GPT-4o, GPT-4o-mini (OpenAI, 2023), LLaMA-3.1-70B and LLaMA-3.1-8B (Dubey et al., 2024) as the backbone LLMs.

Compared Methods. We evaluate MICRO-ACT against two categories of ICL-based approaches: generic reasoning methods that reason on retrieved evidence, including end-to-end QA, few-shot QA, and COT (Wei et al., 2022); and generation-aided reasoning methods that reason with self-generated content of LLMs, including Self-Ask (Press et al., 2023), GKP (Liu et al., 2022), and Comparative (Wang et al., 2023). We evaluate these methods across all five datasets from ConflictBank and KRE. Prompts and implementation details can be found in Appendix G.

Implementation. We implement MICRO-ACT using zero-shot prompting without task-specific customization. To ensure reproducibility, we maintain consistent parameters across all experiments: temperature = 0, top-p = 1, and maximum generation length = 512 tokens (max_tokens for closed-source LLMs, max_new_tokens for open-source models). We utilize the Hugging Face Transformers library for open-source model inference. All experiments with open-source models are conducted on 4 NVIDIA A100 GPUs (80GB), while closed-source models are accessed via their

respective API endpoints.

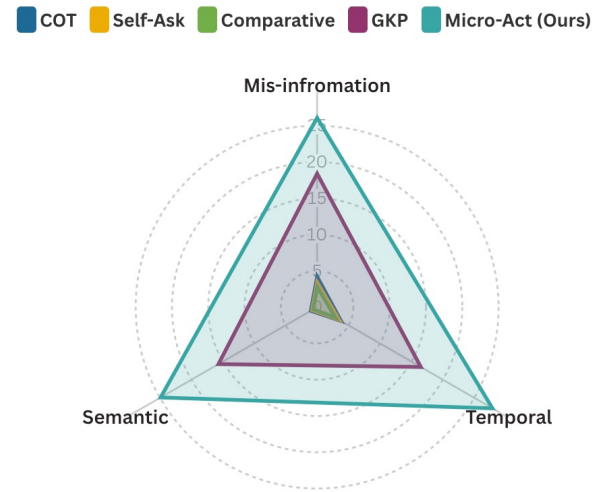


Figure 3: The detailed performance of MICRO-ACT across all 3 conflict types with GPT-4o-mini.

7.2 Main Results

We summarize the performance of **MICRO-ACT** and various baseline methods on ConflictBank and KRE in Table 1. And detailed performance comparison across all three conflict types (i.e., mis-information, temporal, and semantic) is shown in Figure 3. MICRO-ACT surpasses all baseline approaches across all tested LLMs. Notably, MICRO-ACT improves over the previous SOTA method by up to 9.40% on ConflictBank and 6.65% on KRE for GPT-4o-mini, and by 11.47% and 13.85% on LLaMA-3.1-8B, respectively. Results across all 5 datasets and 3 conflict types, confirm the superior capability of MICRO-ACT in handling knowledge conflict and suggest that such superior capability is not model-specific.

7.3 Over-Rationalization Issue

In our experiments, we observed an intriguing phenomenon: when presented with both conflicting evidence and LLMs parametric knowledge, LLMs sometimes attempt to *support all contradictory information as equally valid*. We characterize this behavior as “**over-rationalization**”, which is a tendency to find complex justifications that make contradictory evidence appear compatible. Surprisingly, more capable models like GPT-4o exhibit this behavior more frequently than GPT-4o-mini, leading to performance degradation in GKP as shown in Table 1. Furthermore, we observe that the issue of “over-rationalization” is strongly associated with the type of conflict, occurring more frequently in temporal and semantic conflicts. Unlike misinformation-based conflicts, where conflicts are typically explicit and directly presented in the context, the temporal and semantic conflicts are often implicit beneath the superficial context, misleading LLMs to rationalize both sides of conflict. A detailed case analysis is in Section 7.8.

However, MICRO-ACT can “visualize” the underlying reasoning path via dynamic decomposition to pinpoint finer-grained conflict and focus on those *nuanced conflicts underneath the superficial meaning of context*. Those conflicts cannot be effectively detected through simple side-by-side comparisons used by baseline methods. As illustrated in Figure 3, MICRO-ACT achieves a more significant performance improvement over baselines specifically in the *Temporal* and *Semantic* conflict types. Detailed analysis is in Appendix B.

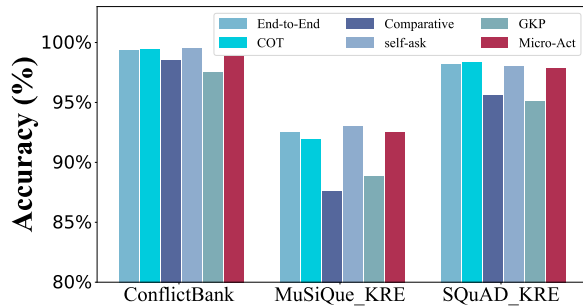


Figure 4: The performance of MICRO-ACT and baselines using GPT-4o-mini under QA task **without knowledge conflict**.

7.4 Robustness Under Conflict-Free Scenarios

Many conflict resolution methods assume the presence of knowledge conflicts. However, in real-world applications, it is often impossible to pre-

determine whether retrieved content conflicts with the parametric knowledge of LLMs, making robustness in conflict-free scenarios crucial.

As shown in Figure 4, existing approaches face a trade-off. Generic reasoning methods like end-to-end and COT achieve high accuracy in conflict-free cases but degrade significantly (by 70-95%) when conflicts arise. And generation-aided methods such as GKP improve conflict resolution but exhibit lower accuracy in conflict-free cases.

MICRO-ACT overcomes this limitation by achieving state-of-the-art performance in conflict scenarios with over 24% performance gain and showing robustness with only sacrificing less than 2% accuracy in conflict-free cases, compared with the end-to-end or self-ask baseline. Rather than introducing biases to favor certain evidence sources, MICRO-ACT helps models automatically identify and analyze potential conflicts through structured action space with decomposition, enabling robust performance regardless of whether conflicts exist.

7.5 Complexity Perception Analysis

To understand how MICRO-ACT adapts its decomposition strategy to different complexity levels, we answer 3 research questions (RQs).

RQ1: How do we objectively measure input complexity?

We select three complementary metrics to comprehensively and objectively measure input complexity: (1) context length captures information volume; (2) domain difficulty reflects inherent reasoning challenges; and (3) perplexity quantifies language uncertainty (Li et al., 2024a,b; Jelinek et al., 1977). As shown in Figure 5, these metrics provide a systematic way to evaluate how different LLMs adapt the decomposition strategies to varying complexity levels.

RQ2: Does decomposition behavior show some patterns across different complexity dimensions?

Figure 5, we observe consistent adaptation patterns within all LLMs. For example, as for the context length shown in Figure 5(a), the decomposition rate increases dramatically from 15% (0-100 tokens) to 95% (400+ tokens). All three complexity dimensions exhibit similar trends, where higher complexity consistently triggers more frequent decomposition. This consistency demonstrates the ability of MICRO-ACT to effectively detect complexity and dynamically adjust granularity via decomposition to reduce complexity.

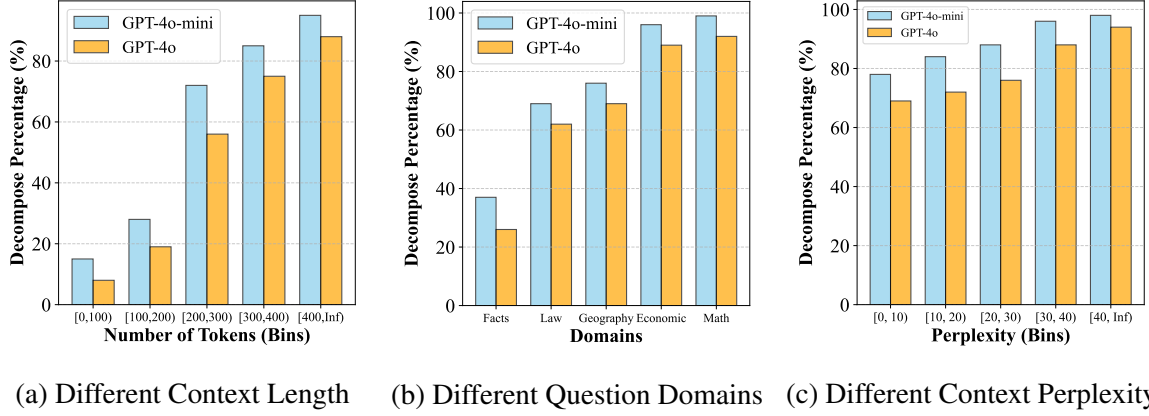


Figure 5: The visual comparisons of the DECOMPOSE action utilization percentage in different complexity, including different context length, question domains and perplexity using GPT-4o-mini and GPT-4o. The detailed calculation of perplexity can be found in Appendix C.

RQ3: Do different LLMs share the same understanding of complexity? The results in Figure 5 show that GPT-4o-mini constantly calls decomposition action more frequently across all complexity dimensions, revealing different complexity tolerance between GPT-4o and GPT-4o-mini, as discussed in Section 5. Rather than requiring manual complexity adjustments for each LLM, MICRO-ACT automatically perceives the complexity and dynamically adapts for different LLMs. This adaptive behavior enables robust performance across different LLMs without model-specific tuning. For example, as shown in Table 1, although LLaMA-3.1-8B is smaller in size and less capable than LLaMA-3.1-70B, MICRO-ACT can still maintain robust performance via more decompose actions to adjust complexity, compared with GKP’s deep performance drop. More analysis is in Appendix F.

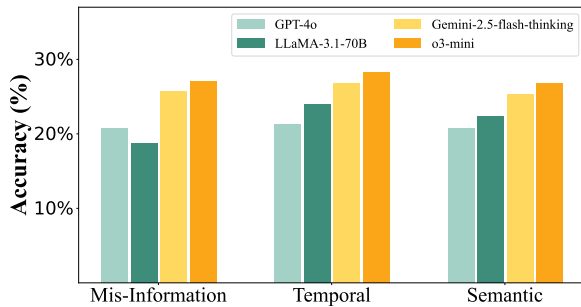


Figure 6: The performance of MICRO-ACT using general LLMs and reasoning LLMs on 300 randomly sampled data for each conflict type.

7.6 General LLMs vs. Reasoning LLMs

As illustrated in Figure 6, the general-purpose LLMs (GPT-4o and Llama-3.1-70B) cluster to-

gether and attain comparatively low scores on all three conflict categories. In contrast, the reasoning-oriented models, Gemini-2.5-flash-thinking and o3-mini, form the top tier and consistently outperform the general models. The gap is most pronounced for misinformation conflicts, which are more amenable to reasoning-based resolution. For temporal and semantic conflicts, the gap narrows because over-rationalization issues arises more often, as discussed in detail in Section 7.3. In summary, stronger reasoning capability markedly boosts the performance of MICRO-ACT. Although it also increases susceptibility to over-rationalization, MICRO-ACT can effectively mitigate this issue and still surpasses the general models.

METHOD	MIS-INFO.	TEMPORAL	SEMANTIC
MICRO-ACT	26.1	27.9	24.9
w/o Navigational Actions	18.4 (↓ 7.7)	18.5 (↓ 9.4)	15.7 (↓ 9.2)
w/o Functional Actions	13.8 (↓ 12.3)	15.2 (↓ 12.7)	13.3 (↓ 11.6)
w/o DECOMPOSE Action	4.2 (↓ 21.9)	4.5 (↓ 23.4)	0.8 (↓ 24.1)

Table 2: Ablation study of MICRO-ACT in three datasets (conflict types) of ConflictBank. The numbers represent the accuracy in percentage. ↓ is an absolute decrease.

7.7 Ablation Study

Table 2 presents an ablation study across all conflict types of ConflictBank. Navigational and Functional actions serve as essential building blocks for conflict resolution, with their removal causing significant performance drops (9.4% for navigational and 12.7% for functional actions). While these actions are necessary for basic operations like context navigation and knowledge comparison, their

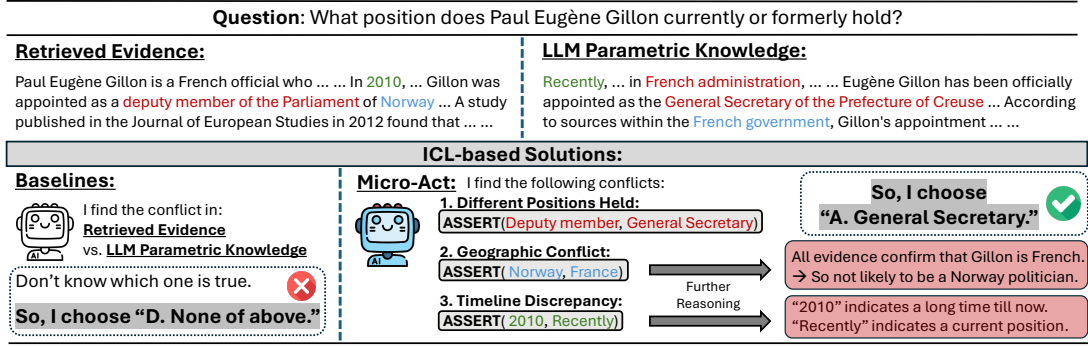


Figure 7: A case study of MICRO-ACT and baselines models under a real knowledge conflict case. MICRO-ACT can pinpoint fine-grained conflict points instead of being distracted by irrelevant context.

effectiveness heavily depends on appropriate input granularity. Without proper guidance on the input granularity of those actions, the model struggles to maintain consistent performance, especially with complex contexts.

Decomposition action dramatically improves performance by dynamically adjusting input granularity for other actions. Its removal causes the most severe degradation (over 20% performance drop), highlighting its crucial role. Through iterative decomposition, MICRO-ACT continuously refines inputs of other actions until they find the optimal granularity level where navigational and functional actions can operate most effectively. As discussed in Section 5, MICRO-ACT effectively elicits the latent ability of LLMs to perceive complexity and adapt to different environments. This adaptive process enhances the confidence of MICRO-ACT by ensuring all action components receive fine-grained information aligned with its capability, leading to higher accuracy in complex cases.

7.8 Case Study

In this case study, we demonstrate how MICRO-ACT identifies nuanced conflicts underneath the superficial meaning of context which can hardly be located by simple side-by-side comparisons that baselines use. Consider the question “*What position does Paul Eugène Gillon currently or formerly hold?*”, where the retrieved context conflicts with LLM parametric knowledge as shown in Figure 7. MICRO-ACT can identify the different time references (2010 vs. recently) and location (Norway vs. France). Then apply step-by-step reasoning to find the *underlying* conflicts *beyond* the *superficial* context: (1) Majority consensus suggests he is from France, not Norway. (2) 2010 indicates a very long

appointment, which is less likely compared with a recent appointment, given the question: What position does Paul Eugène Gillon currently or formerly hold? (3) And finally, determine that Paul Eugène Gillon was recently appointed as a French politician then answer the question correctly.

8 Cost Analysis

MICRO-ACT incurs a modest computational cost, where on ConflictBank it processes roughly 2.8 times more input tokens and 1.3 times more output tokens than the strongest baseline (GKP), translating to only \$0.008 extra per GPT-4o query and \$0.0005 with GPT-4o-mini, while inference latency rises by 0.6 s and 0.3 s respectively. Crucially, these overheads appear only when genuine conflicts trigger deeper decomposition; conflict-free questions finish as quickly as the baseline. Given the substantial gains in conflict-resolution accuracy reported in Table 1, the marginal cost and delay are acceptable for real-world RAG deployments. Detailed token, cost, and timing breakdowns are provided in Appendix A.

9 Conclusion

We proposed **MICRO-ACT**, a framework that addresses knowledge conflicts in RAG systems through hierarchical action decomposition. By automatically perceiving context complexity and breaking down comparisons into fine-grained steps, MICRO-ACT overcomes the limitations of simple side-by-side comparisons for example the *over-rationalization* issue. Extensive experiments demonstrate its effectiveness across multiple datasets and conflict types, while maintaining strong robustness in non-conflict scenarios, making it particularly valuable for real-world RAG.

10 Limitations

While MICRO-ACT demonstrates strong performance in knowledge conflict resolution, several limitations warrant discussion. First, our MICRO-ACT needs additional intermediate steps to effectively pinpoint the conflicts underneath the superficial meaning of context, which can hardly be located by simple side-by-side comparisons that baselines use. Although baselines like end-to-end and COT (Wei et al., 2022) are lightweight, their poor performance in knowledge conflict harms the effectiveness of RAG systems. We believe the efficiency should be considered after good performance. As detailed analyzed in Appendix A, the extra overhead is relatively small and our analysis demonstrates that MICRO-ACT’s modest overhead is justified by its significantly enhanced conflict resolution performance. Second, our current evaluation focuses primarily on English language contexts. The effectiveness of decomposition strategies might vary across different languages and cultural contexts. Nevertheless, our work represents an important milestone in knowledge conflict resolution, establishing a strong foundation for future research in this critical area.

11 Acknowledgement

Reynold Cheng, Nan Huo, Jinyang Li, and Ge Qu are supported by the Hong Kong Jockey Club Charities Trust (Project 260920140), the University of Hong Kong (Project 2409100399), the HKU Outstanding Research Student Supervisor Award 2022-23, and the HKU Faculty Exchange Award 2024 (Faculty of Engineering). Bowen Qin was supported by National Science and Technology Major Project (Project 2022ZD0116306). Chenhao Ma was partially supported by NSFC under Grant 62302421, Basic and Applied Basic Research Fund in Guangdong Province under Grant 2023A1515011280, 2025A1515010439, Ant Group through CCF-Ant Research Fund, Shenzhen Research Institute of Big Data under grant SIF20240004, and the Guangdong Provincial Key Laboratory of Big Data Computing, The Chinese University of Hong Kong, Shenzhen.

12 Ethical Statement

We prioritize ethical considerations throughout our research process. During data collection and preprocessing, we carefully filtered out examples containing sensitive, biased, or potentially harmful

content to ensure our evaluation focuses on constructive knowledge resolution scenarios. Our in-context learning approach requires no additional training of language models, significantly reducing the environmental impact compared to fine-tuning methods. This aligns with growing concerns about the carbon footprint of AI research. Furthermore, all datasets used in this work are publicly available, ensuring reproducibility.

References

- Adam Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2020. Learning to retrieve reasoning paths over wikipedia graph for question answering. In *International Conference on Learning Representations (ICLR)*.
- Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.
- 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.
- Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1870–1879.
- Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024. Benchmarking large language models in retrieval-augmented generation. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 17754–17762. AAAI Press.
- Sihao Chen, Fan Zhang, Kazuo Sone, and Dan Roth. 2021. Improving faithfulness in abstractive summarization with contrast candidate generation and selection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5935–5941.
- Costanza Conforti, Jakob Berndt, Mohammad Taher Pilehvar, Chryssi Giannitsarou, Flavio Toxvaerd, and Nigel Collier. 2020. Stander: An expert-annotated

- dataset for news stance detection and evidence retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4086–4101.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, and et al. Angela Fan. 2024. *The llama 3 herd of models*. *ArXiv*, abs/2407.21783.
- Tianqing Fang, Zhaowei Wang, Wenxuan Zhou, Hongming Zhang, Yangqiu Song, and Muhao Chen. 2024. Getting sick after seeing a doctor? diagnosing and mitigating knowledge conflicts in event temporal reasoning. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3846–3868.
- Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.
- Luyu Gao, Zhu Yun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, et al. 2023. Rarr: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16477–16508.
- Yu Gu, Yiheng Shu, Hao Yu, Xiao Liu, Yuxiao Dong, Jie Tang, Jayanth Srinivasa, Hugo Latapie, and Yu Su. 2024. Middleware for llms: Tools are instrumental for language agents in complex environments. *arXiv preprint arXiv:2402.14672*.
- Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020a. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.
- Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020b. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.
- Gautier Izacard and Édouard Grave. 2021a. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880.
- Gautier Izacard and Edouard Grave. 2021b. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 874–880.
- Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43.
- Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. 1977. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63.
- Zhuoran Jin, Pengfei Cao, Yubo Chen, Kang Liu, Xiaojian Jiang, Jiexin Xu, Qiuxia Li, and Jun Zhao. 2024a. Tug-of-war between knowledge: Exploring and resolving knowledge conflicts in retrieval-augmented language models. *arXiv preprint arXiv:2402.14409*.
- Zhuoran Jin, Pengfei Cao, Hongbang Yuan, Yubo Chen, Jiexin Xu, Huaijun Li, Xiaojian Jiang, Kang Liu, and Jun Zhao. 2024b. Cutting off the head ends the conflict: A mechanism for interpreting and mitigating knowledge conflicts in language models. *arXiv preprint arXiv:2402.18154*.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.
- Jinyang Li, Binyuan Hui, Ge Qu, Jiayi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, et al. 2024a. Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls. *Advances in Neural Information Processing Systems*, 36.
- Jinyang Li, Nan Huo, Yan Gao, Jiayi Shi, Yingxiu Zhao, Ge Qu, Yurong Wu, Chenhao Ma, Jian-Guang Lou, and Reynold Cheng. 2024b. Tapilot-crossing: Benchmarking and evolving llms towards interactive data analysis agents. *arXiv preprint arXiv:2403.05307*.
- Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel J. Orr, Lucia Zheng, Mert Yüksesgönül, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri S. Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2023. Holistic evaluation of language models. *Trans. Mach. Learn. Res.*, 2023.
- Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh

- Hajishirzi. 2022. Generated knowledge prompting for commonsense reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3154–3169.
- Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.
- Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. Generation-augmented retrieval for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4089–4100.
- Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100.
- Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. Ambigqa: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797.
- Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2024. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. *arXiv preprint arXiv:2410.05229*.
- Shikhar Murty, Christopher D. Manning, Peter Shaw, Mandar Joshi, and Kenton Lee. 2024. BAGEL: bootstrapping agents by guiding exploration with language. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.
- 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*.
- OpenAI. 2023. *Gpt-4 technical report*. Technical report, OpenAI.
- Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 2463–2473.
- Quang Pham, Hoang Ngo, Luu Anh Tuan, and Dat Quoc Nguyen. 2024. Who’s who: Large language models meet knowledge conflicts in practice. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10142–10151.
- Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711.
- Libo Qin, Yijia Liu, Wanxiang Che, Haoyang Wen, Yangming Li, and Ting Liu. 2019. Entity-consistent end-to-end task-oriented dialogue system with kb retriever. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 133–142.
- Libo Qin, Yijia Liu, Wanxiang Che, Haoyang Wen, and Ting Liu. 2018. End-to-end task-oriented dialogue system with distantly supervised knowledge base retriever. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data: 17th China National Conference, CCL 2018, and 6th International Symposium, NLP-NABD 2018, Changsha, China, October 19–21, 2018, Proceedings 17*, pages 238–249. Springer.
- Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don’t know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 784–789. Association for Computational Linguistics.
- Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlga, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.
- Ruiyang Ren, Yuhao Wang, Yingqi Qu, Wayne Xin Zhao, Jing Liu, Hao Tian, Hua Wu, Ji-Rong Wen, and Haifeng Wang. 2023. Investigating the factual knowledge boundary of large language models with retrieval augmentation. *arXiv preprint arXiv:2307.11019*.
- Kiamehr Rezaee, Jose Camacho-Collados, and Mohammad Taher Pilehvar. 2024. Tweeter: A benchmark for target entity retrieval on twitter without knowledge bases. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16890–16896.
- Sagi Shai, Ari Kobren, and Philip Ogren. 2024. Adaptive question answering: Enhancing language model proficiency for addressing knowledge conflicts with source citations. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17226–17239.

- Dan Shi, Renren Jin, Tianhao Shen, Weilong Dong, Xinwei Wu, and Deyi Xiong. 2024a. Ircan: Mitigating knowledge conflicts in llm generation via identifying and reweighting context-aware neurons. *arXiv preprint arXiv:2406.18406*.
- Qi Shi, Han Cui, Haofeng Wang, Qingfu Zhu, Wanxiang Che, and Ting Liu. 2024b. Exploring hybrid question answering via program-based prompting. *arXiv preprint arXiv:2402.10812*.
- Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024c. Replug: Retrieval-augmented black-box language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8364–8377.
- Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803.
- Zhaochen Su, Jun Zhang, Xiaoye Qu, Tong Zhu, Yanshu Li, Jiashuo Sun, Juntao Li, Min Zhang, and Yu Cheng. 2024. Conflictbank: A benchmark for evaluating the influence of knowledge conflicts in llm. *arXiv preprint arXiv:2408.12076*.
- Hexiang Tan, Fei Sun, Wanli Yang, Yuanzhuo Wang, Qi Cao, and Xueqi Cheng. 2024. Blinded by generated contexts: How language models merge generated and retrieved contexts when knowledge conflicts? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6207–6227.
- James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multi-hop questions via single-hop question composition. *Trans. Assoc. Comput. Linguistics*, 10:539–554.
- Fei Wang, Xingchen Wan, Ruoxi Sun, Jiefeng Chen, and Serkan Ö Arik. 2024. Astute rag: Overcoming imperfect retrieval augmentation and knowledge conflicts for large language models. *arXiv preprint arXiv:2410.07176*.
- Shuohang Wang, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2021. Want to reduce labeling cost? gpt-3 can help. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4195–4205.
- Yike Wang, Shangbin Feng, Heng Wang, Weijia Shi, Vidhisha Balachandran, Tianxing He, and Yulia Tsvetkov. 2023. Resolving knowledge conflicts in large language models. *arXiv preprint arXiv:2310.00935*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.
- Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2024. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*.
- Rongwu Xu, Zehan Qi, Zhijiang Guo, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. 2024. Knowledge conflicts for llms: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 8541–8565. Association for Computational Linguistics.
- John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. 2024. Swe-agent: Agent-computer interfaces enable automated software engineering. *arXiv preprint arXiv:2405.15793*.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Jiahao Ying, Yixin Cao, Kai Xiong, Yidong He, Long Cui, and Yongbin Liu. 2023. Intuitive or dependent? investigating llms’ robustness to conflicting prompts. *arXiv preprint arXiv:2309.17415*.
- Xiaowei Yuan, Zhao Yang, Yequan Wang, Shengping Liu, Jun Zhao, and Kang Liu. 2024. Discerning and resolving knowledge conflicts through adaptive decoding with contextual information-entropy constraint. *arXiv preprint arXiv:2402.11893*.

- Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Advances in neural information processing systems*, 32.
- Xuanliang Zhang, Dingzirui Wang, Longxu Dou, Qingfu Zhu, and Wanxiang Che. 2025. Murre: Multi-hop table retrieval with removal for open-domain text-to-sql. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 5789–5806.
- Xuanliang Zhang, Dingzirui Wang, Baoxin Wang, Longxu Dou, Xinyuan Lu, Keyan Xu, Dayong Wu, Qingfu Zhu, and Wanxiang Che. 2024. Scitat: A question answering benchmark for scientific tables and text covering diverse reasoning types. *arXiv preprint arXiv:2412.11757*.
- 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*.

A Cost-Performance Trade-Off

In this section, we provide a detailed cost and latency analysis on the ConflictBank dataset, comparing MICRO-ACT with the most powerful baseline within Generic Reasoning and Generation-aided Reasoning groups (i.e., COT and GKP) using GPT-4o and GPT-4o-mini, as shown in Table 3 and Table 4.

A.1 Token Usage and Monetary Cost

- **Input Tokens:** Because MICRO-ACT dynamically decomposes conflicts, it initiates additional turns to resolve contradictions. As a result, it uses about 2.8× more input tokens than GKP.
- **Output Tokens:** MICRO-ACT’s output length is roughly 1.3× higher than GKP. Although it writes multiple short intermediate responses, the final output does not explode in length, as the output of each intermediate step tends to be relatively concise.
- **Overall Cost:** The cost difference comes to an additional \$0.008 per query for GPT-4o. For GPT-4o-mini, the extra cost is only \$0.0005 per query, which is fairly small.

A.2 Inference Time Overhead

- **GPT-4o:** MICRO-ACT takes 1.9s on average, which is about 0.6s longer than GKP (1.3s).
- **GPT-4o-mini:** MICRO-ACT requires 0.7s on average, 0.3s longer than GKP (0.4s).
- The latency shown above is not using multi-threading. If we use multi-threading, the extra inference latency will be further reduced significantly.

This additional overhead comes from the extra decomposition steps in scenarios where conflicts are actually perceived by MICRO-ACT. However, for conflict-free queries, MICRO-ACT performs fewer steps, avoiding this overhead.

A.3 Justification of Additional Overhead

1. **Significant Performance Gain:** As shown in our main experiments, MICRO-ACT achieves **notable improvements** in resolving knowledge conflicts. It indicates that our dynamic decomposition approach is essential for detecting finer-grained conflicts.

2. **Adaptive Depth:** MICRO-ACT only needs deeper decomposition when a conflict is perceived, which is necessary to locate the underlying conflicts that baselines cannot find. On conflict-free questions, it quickly finalizes the answer, keeping cost and latency low.
3. **Practical Applicability:** In many real-world applications such as SWE-agent (Yang et al., 2024), baseline query costs more than \$2 per query. We believe that an additional \$0.008 in GPT-4o (and \$0.0005 for GPT-4o-mini) and 0.6 extra seconds in GPT-4o (and 0.3 sec for GPT-4o-mini) per query is acceptable given the significantly improved performance, especially for real-world scenarios.

We believe this analysis demonstrates that MICRO-ACT’s **modest overhead** is justified by its **significantly enhanced** conflict resolution capabilities.

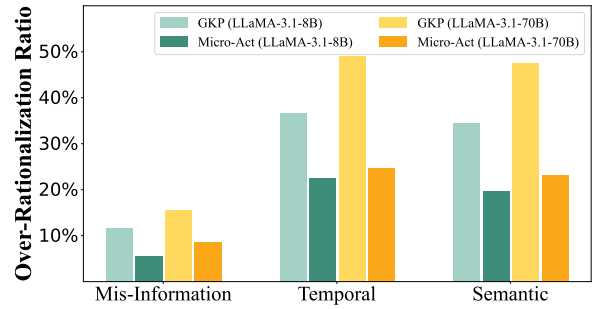


Figure 8: The Over-Rationalization Ratio of GKP and MICRO-ACT using LLaMA-3.1-8B and LLaMA-3.1-70B (lower is better).

B Over-Rationalization Issue and Analysis

In this section, we present a quantitative investigation on 600 queries (200 from each conflict type) in the ConflictBank dataset, comparing GKP (strongest baseline) and MICRO-ACT on two LLMs: LLaMA-3.1-70B and GPT-4o-mini. We measure the proportion of queries in which the model exhibits *over-rationalization* (i.e., rationalizing contradictory facts in their step-by-step chain-of-thought reasoning).

From Figure 8, it is clear that all models exhibit a higher ratio of over-rationalization under temporal and semantic conflict types, as these two types are more easily rationalized. For example, as shown in Figure 7, conflicting knowledge about whether Paul Eugène Gillon was appointed

Method	Avg. # Turns	Avg. Input Tokens	Avg. Output Tokens	Avg. Cost (USD)	Avg. Inference Time (s)
COT	2.0	652	421	\$0.006	0.9
GKP	2.0	1182	856	\$0.012	1.3
MICRO-ACT	3.4	3345	1137	\$0.020	1.9

Table 3: Cost analysis on GPT-4o.

Method	Avg. # Turns	Avg. Input Tokens	Avg. Output Tokens	Avg. Cost (USD)	Avg. Inference Time (s)
COT	2.0	689	469	\$0.0004	0.3
GKP	2.0	1239	894	\$0.0007	0.4
MICRO-ACT	3.6	3532	1289	\$0.0013	0.7

Table 4: Cost analysis on GPT-4o-mini.

in 2010 or more recently can both be reasonable, making it difficult for the model to rely on retrieved evidence or LLM parametric knowledge.

The GKP method shows a higher tendency to rationalize these conflicts, as LLMs are easily distracted by complex context (Mirzadeh et al., 2024). This leads to failure in identifying the fine-grained conflicts underneath. In contrast, MICRO-ACT dynamically decomposes the complex context to reduce its complexity and identifies the conflicts behind the superficial context. This enables MICRO-ACT to pinpoint conflict points and reason on them for correct answers.

B.1 A More Detailed Case Study

In Figure 7, an example of contradictory information regarding Paul Eugène Gillon being appointed in 2010 versus a more recent appointment is shown. Additionally, incorrect evidence suggests he might be a politician in Norway. By iteratively decomposing the knowledge, MICRO-ACT is able to:

1. Identify the conflicting time references (2010 vs. recently) and location (Norway vs. France).
2. Apply step-by-step reasoning to find the underlying conflicts beyond the superficial context:
 - Majority consensus suggests he is from France, not Norway.
 - 2010 indicates a very long appointment, which is less likely compared with a recent appointment, given the question: What position does Paul Eugène Gillon currently or formerly hold?

3. Finally, determine that Paul Eugène Gillon was recently appointed as a French politician and correctly answer the question.

These nuanced conflicts are hidden beneath the superficial meaning of context and can hardly be detected by simple side-by-side comparisons used by baseline models.

C Perplexity Calculation

In this section, we provide a detailed explanation of how to compute the perplexity (PPL) of a given text using the GPT-2 language model, which is not any of the language models used in our work. The aim is to provide an objective measurement of knowledge context complexity. Perplexity is a widely used metric for evaluating the quality of language models, indicating how well the model predicts a sample of text (Jelinek et al., 1977). In this work, by fixing the language model to be GPT-2, we use PPL to measure the complexity of context text. A lower perplexity value generally corresponds to a lower context text complexity.

C.1 Formal Definition of Perplexity

Let the text be represented as a sequence of tokens:

$$W = w_1, w_2, \dots, w_N, \quad (15)$$

where each w_i is a token (e.g., a subword unit as utilized by GPT-2).

Given a language model that estimates the probability of each token conditioned on all previous tokens, the joint probability of the sequence W can be factorized as:

$$P(W) = \prod_{i=1}^N P(w_i | w_1, w_2, \dots, w_{i-1}). \quad (16)$$

The perplexity of the sequence W under the model is defined as:

$$\text{PPL}(W) = \exp \left(-\frac{1}{N} \sum_{i=1}^N \ln P(w_i | w_1, w_2, \dots, w_{i-1}) \right). \quad (17)$$

In other terms, if we use base-2 logarithms:

$$\text{PPL}(W) = 2^{-\frac{1}{N} \sum_{i=1}^N \log_2 P(w_i | w_1, \dots, w_{i-1})}. \quad (18)$$

The perplexity can be interpreted as the effective average branching factor of the language model. A perfectly predicting model (one that assigns probability 1 to the observed sequence) would achieve a perplexity of 1.

C.2 GPT-2-Based Computation

GPT-2 is a Transformer-based language model trained on large-scale text data. It provides a probability distribution over the next token w_i given the previous tokens (w_1, \dots, w_{i-1}) . Formally, GPT-2 implements:

$$P(w_i | w_1, \dots, w_{i-1}) = \text{softmax}(h_{i-1} W_e)_{w_i}, \quad (19)$$

where h_{i-1} is the hidden state vector produced by the Transformer after processing tokens w_1, \dots, w_{i-1} , and W_e is the token embedding matrix used to map hidden states to logits over the vocabulary. The softmax function converts these logits into probabilities.

C.3 Practical Steps for Perplexity Calculation

To compute perplexity using GPT-2 in practice, one would proceed as follows:

Tokenization. Convert the raw text into GPT-2 compatible tokens:

$$\text{Text} \xrightarrow{\text{tokenizer}} (w_1, w_2, \dots, w_N). \quad (20)$$

Model Inference. For each token w_i , feed the preceding tokens $(w_1, w_2, \dots, w_{i-1})$ into GPT-2 to obtain:

$$P(w_i | w_1, \dots, w_{i-1}). \quad (21)$$

This is done by running the model \mathcal{M} which is GPT-2 forward pass:

$$(h_1, h_2, \dots, h_{i-1}) = \mathcal{M}(w_1, w_2, \dots, w_{i-1}) \quad (22)$$

and then applying the softmax over the output logits to get the probability of w_i .

Log Probability Computation. The next step is to extract $\ln P(w_i | w_1, \dots, w_{i-1})$ from the model's output distribution.

Summation and Normalization. Compute the average negative log-probability:

$$-\frac{1}{N} \sum_{i=1}^N \ln P(w_i | w_1, \dots, w_{i-1}). \quad (23)$$

Exponentiation Take the exponential of this value to obtain the perplexity:

$$\text{PPL}(W) = \exp \left(-\frac{1}{N} \sum_{i=1}^N \ln P(w_i | w_1, \dots, w_{i-1}) \right). \quad (24)$$

D Mathematical Derivation of Knowledge Representation Transitions

In this section, we present a formal derivation of the knowledge representation transition process. We begin by defining the key probability distributions:

- $p_t(K_n)$: The distribution of knowledge representation at step t
- $p_{\text{model}}(K_n | K)$: The probability of the model generating new knowledge representation K_n from previous state K
- $p_{\text{verify}}(c | K_n)$: The probability of the verifier generating conflict detection result c based on knowledge representation K_n

Following the principles of probabilistic state transitions, we can establish two fundamental equations:

D.1 Knowledge Representation Transition Equation

The transition to new knowledge representation can be expressed as:

$$p_t(K_n) = \sum_{c'} p_{\text{model}}(K_n | c') \cdot p_{t-1}(c') \quad (25)$$

This equation represents how new knowledge representation are derived from previous conflict detection results.

D.2 Conflict Detection Equation

The probability distribution of conflict detection results is given by:

$$p_{t-1}(c') = \sum_K p_{verify}(c'|K) \cdot p_{t-1}(K) \quad (26)$$

This captures how conflict detection results are generated based on previous knowledge representation.

D.3 Combined Transition Model

Substituting the conflict detection equation into the state transition equation yields:

$$p_t(K_n) = \sum_{c'} p_{model}(K_n|c') \cdot \left[\sum_K p_{verify}(c'|K) \cdot p_{t-1}(K) \right] \quad (27)$$

Rearranging the summation order:

$$p_t(K_n) = \sum_{c'} \sum_K p_{model}(K_n|c') \cdot p_{verify}(c'|K) \cdot p_{t-1}(K) \quad (28)$$

D.4 Bayesian Formulation

Applying Bayes' rule to transform $p_{verify}(c'|K)$ into $p_{verify}(K|c')$:

$$p_t(K_n) = \sum_{K, c'} p_{t-1}(K) \cdot p_{verify}(K|c') \cdot p_{model}(K_n|c') \quad (29)$$

This final form encapsulates three key components:

- $p_{t-1}(K)$: Distribution of previous knowledge representation
- $p_{verify}(K|c')$: Verifier's evaluation of knowledge
- $p_{model}(K_n|c')$: Model's probability of generating new knowledge based on conflict detection

This formulation provides a comprehensive mathematical framework for analyzing the evolution of knowledge representation through iterative refinement and verification.

E Error Analysis

To better understand the limitations of MICRO-ACT, we conducted a detailed error analysis on 1,000 randomly sampled examples across all five datasets. Our analysis revealed two predominant error types:

Context Distraction (63% of errors) Despite our decomposition strategy, LLMs occasionally become overwhelmed by complex contexts and default to expressing uncertainty ("I don't know" or "Cannot determine"). This typically occurs when the context contains multiple interrelated facts or complex logical relationships that challenge the model's ability to maintain coherent reasoning chains. For instance, in multi-hop reasoning questions where evidence pieces are densely connected, even decomposed segments may retain inherent complexity that exceeds the model's processing capacity.

Over-reliance on Retrieved Evidence (37% of errors) The second major error type manifests when LLMs exhibit a strong bias toward retrieved evidence, even when it conflicts with their parametric knowledge. This behavior is particularly prominent in cases where the retrieved evidence appears more specific or detailed than the model's inherent knowledge. Such errors suggest that while MICRO-ACT effectively identifies conflicts, the final resolution step may still be influenced by an implicit bias toward explicit external information over learned knowledge.

These findings indicate that while MICRO-ACT significantly improves conflict resolution, future work should focus on enhancing the model's ability to maintain reasoning clarity in highly complex scenarios and developing more balanced weighing mechanisms between retrieved and parametric knowledge.

F More Complexity Aspect Analysis

Besides the complexity aspects discussed in Figure 5, we explore how the number of decomposition steps (i.e., the number of times MICRO-ACT invokes its "DECOMPOSE" action) varies across different conflict types. We collect more results on temporal, misinformation, and semantic conflicts for both GPT-4o and GPT-4o-mini. Below is a summary of the average number of decomposition steps taken for each conflict category:

Action	Expression	Description
DECOMPOSE	Action i → action 1, action 2	Split a complex action i into smaller, manageable steps such as action 1 and action 2.
ELICIT	Query i → LLM Knowledge	Elicit LLM knowledge about a given query.
REASON	Knowledge i → Reasoning Path	Reason on input knowledge to generate the reasoning path.
ASSERT	Knowledge i , Knowledge j → 0 or 1	Judge whether the two input knowledge have conflict or not.

Figure 9: The table of actions in our hierarchical action space.

Table 5: Average DECOMPOSE action steps per conflict type.

Conflict Type	GPT-4o (Avg. Steps)	GPT-4o-mini (Avg. Steps)
Misinformation	0.8	1.3
Temporal	1.6	2.2
Semantic	1.5	2.3

Temporal and Semantic Conflicts. We can see that temporal and semantic conflict types usually trigger a higher number of decomposition steps. As detailed in Appendix B, these two types are more easily to be “over-rationalized”. Thus, more decomposition steps are needed to investigate underneath conflicts and the logic flaws behind the superficial context, which looks reasonable.

Misinformation Conflicts. Misinformation typically involves more **superficial conflicts**, which is more intuitive. As a result, fewer DECOMPOSE actions are invoked because there is less ambiguity in the evidence to untangle.

Why Decompose More? The observed more decomposition is mainly because the nuanced conflicts are **underneath the superficial context** and can hardly be located by simple side-by-side comparisons that baselines use, detailed discussion can be found in Appendix B.

Model Size & Complexity. We observe that GPT-4o-mini has a higher decomposition step count across all conflict types. Smaller LLMs often require additional steps to reduce complexity, revealing different complexity tolerance between GPT-4o and GPT-4o-mini, as discussed in Section 5. A more detailed discussion can be found in Section 7.5, RQ3.

This illustrates exactly **why** and **how** MICRO-ACT’s dynamic reasoning pipeline triggers additional decomposition for temporal and semantic conflicts, where potential “over-rationalizations” are more likely to arise.

G Implementation Details

G.1 Action Details

All the actions designed in our proposed hierarchical action space are illustrated in Table 9.

G.2 Generic Reasoning Models

End-to-End QA. The End-to-End QA prompt as shown in Figure 10, directly provides the model with the question and requests an immediate, self-contained answer. It contains no intermediate reasoning instructions, and the model is expected to return its best guess in a single generation pass. This approach assumes the model’s internal representations are sufficient for reasoning without explicitly prompting it to show work.

Few-Shot QA. The Few-Shot QA prompt as shown in Figure 11, includes one or more example QA pairs before presenting the target question. These examples help the model align its reasoning with the desired output format and style. The provided examples are chosen to be representative of the question domain and complexity level.

Chain-of-Thought. The Chain-of-Thought (Wei et al., 2022) prompt as shown in Figure 12, instructs the model to show its intermediate reasoning steps explicitly. After presenting the question, the prompt requests the model to “think aloud” by outlining its reasoning process before concluding with

a final, concise answer. This approach encourages the model to form more coherent and justifiable solutions.

G.3 Generation-aided Reasoning Models

Self-Ask. The Self-Ask (Press et al., 2023) prompt as shown in Figure 13, breaks down a complex question into sub-questions and then prompts the model to answer them step-by-step. By iteratively generating and resolving subtasks, the model can handle multi-step reasoning tasks more systematically, ultimately consolidating the intermediate answers into a final solution.

Comparative QA. The Comparative QA (Wang et al., 2023) prompt as shown in Figure 14, asks LLMs generate the answer of the question regardless of the retrieved evidence at first. Then answer the question by considering both the retrieved evidence and the self-generated answer.

Generation Phase of GKP. In the Generation Phase of GKP (Liu et al., 2022), the prompt encourages the model to generate the knowledge needed to answer the given question. The model then lists relevant knowledge without yet providing the final answer.

Answering Phase of GKP. Once the self-generated knowledge is established, the Answering Phase of GKP (Liu et al., 2022) prompt feeds the previously generated knowledge back into the model. Using this as a guide, the model now produces a final answer. This is a two-step process.

G.4 MICRO-ACT

MICRO-ACT. The prompt of our proposed MICRO-ACT model is shown in Figure 17.

H Human Evaluation

To assess the quality and representativeness of our 1,000-instance samples, we conducted a human study with 10 volunteer expert annotators, each having substantial experience in QA tasks.

Evaluation Procedure. For each dataset, we presented each annotator with a total of 100 QA items: 50 randomly drawn from the full dataset and 50 randomly drawn from the 1,000-instance sample. Annotators were blind to which items came from which source. Each annotator answered all 100 questions to the best of their ability.

Measurements. We measured annotator accuracy, defined as the proportion of correct answers, on both subsets. Across all datasets, the average accuracy on sample-based QA pairs differed by less than 5% from that on the corresponding full-dataset pairs. This consistency suggests that the sampled subsets do not introduce systematic bias in terms of difficulty or content distribution.

Internal Agreement. To ensure that results were not driven by a few outliers, we examined internal agreement among the 10 annotators. We computed Fleiss’ kappa (Fleiss, 1971), which was consistently above 0.80 for all datasets, indicating substantial agreement. In addition, the standard deviation of accuracy across annotators remained under 2% for each subset type, reflecting stable and consistent performance patterns.

These findings demonstrate that our sampling strategy preserves the key characteristics of the original datasets, maintaining both content diversity and difficulty level, and that our evaluation results are reliable and robust across multiple independent annotators. After the review period, we will open-source the sampled datasets for reproduction and for researchers to develop more advanced methods on knowledge conflict.

I Model Descriptions

Our empirical evaluation employs three representative language models, each positioned at different capability levels.

GPT-4o. GPT-4o (OpenAI, 2023) is a state-of-the-art foundation model that excels at complex reasoning tasks. Our experiments leverage its robust instruction-following capabilities and advanced reasoning abilities to evaluate the upper bounds of adaptive complexity.

GPT-4o-mini. GPT-4o-mini (OpenAI, 2023) is a balanced model that combines computational efficiency with strong reasoning capabilities. This model serves as an ideal testbed for examining how moderate model capacity influences the granularity of knowledge decomposition across varying task complexities.

LLaMA-3.1-70B-Instruct. LLaMA-3.1-70B-Instruct (Dubey et al., 2024) is a 70-billion parameter large language model built on the LLaMA architecture. This instruction-tuned variant exhibits strong performance across diverse

NLP tasks, with particular strengths in reasoning and coherent text generation.

LLaMA-3.1-8B-Instruct. LLaMA-3.1-8B-Instruct ([Dubey et al., 2024](#)) is a 8-billion parameter large language model built on the LLaMA architecture. This instruction-tuned variant exhibits strong performance across diverse NLP tasks, with particular strengths in reasoning and coherent text generation.

End-to-End Baseline Prompt:

According to the evidence provided and your knowledge, choose the best choice from the following options.

Evidence: {evidence}

Question: {question}

{options}

The answer should be concise and short, not with explanations nor option details. Fill your selected option in **uppercase letter format** into the template: <a>..., for example <a>H.

The final answer is: <a>

Figure 10: The prompt of the End-to-End QA baseline method.

Few-Shot Baseline Prompt:

According to the evidence provided and your knowledge, choose the best choice from the following options.

Evidence: Otters are semi-aquatic mammals that prefer to inhabit areas near water bodies for easy access to drinking water and food.

Question: Otters enter their new habitat. What happened as a more possible result?

A. Otters start looking for abalone for food.

B. They always live by the water so that they can drink water easily.

The final answer is: <a>A

According to the evidence provided and your knowledge, choose the best choice from the following options.

Evidence: Buying a variety of textbooks can provide a wide range of information, making it easier to find specific details quickly.

Question: Lila can find what she wants quickly. What is the more possible cause of this?

A. Lila bought several kinds of textbooks.

B. Lila loves classification of her things.

The final answer is: <a>B

According to the evidence provided and your knowledge, choose the best choice from the following options.

Evidence: Hunting cottontails is often associated with drinking rum as a tradition or ritual.

Question: He got some rum. What is the more possible cause of this?

A. The worker fermented some sugar cane with yeast.

B. Tom went out and want to hunt some cottontails.

The final answer is: <a>A

Evidence: {evidence}

Question: {question}

{options}

The final answer is: <a>

Figure 11: The prompt of the Few-Shot QA baseline method.

COT Baseline Prompt:

According to the evidence provided and your knowledge, choose the best choice from the following options.

Evidence: {evidence}

Question: {question}

{options}

Please think step by step to answer the question and fill in the template: <t>...</t> with your thoughts. And then generate your selected option to fill in the template: <a>... in **uppercase letter format**, for example <a>H.

Step-by-step thought: <t>

Figure 12: The prompt of the COT (Wei et al., 2022) baseline method.

Self-Ask Prompt:

Answer the question based on the given context and your own knowledge respectively.

Format Examples:

Context: [Some Context]
Question: [The Given Question]
Follow up: [You should ask follow up questions.]
Follow up: [You can ask more than one questions.]

...
Answer: <a> E

You have to generate the final answer about the question after ask several follow-up questions. The answer should be concise and short, not with explanations nor option details. Fill your answer in **uppercase letter format** into the template: <a>....

Context: {evidence}
Question: {question}
{options}

Follow up:

Figure 13: The prompt of the Self-Ask (Press et al., 2023) QA baseline method.

Comparative Prompt:

Generate two answers to the given question: ANSWER1 solely based on the given context and ANSWER2 solely based on your own knowledge. Then based on the two answer give the final answer and fill your final answer in **uppercase letter format** into the template: <a>..., for example <a>H.

Context: {evidence}
Question: {question}
{options}

Answer the question solely based on the given context.
Answer:

Figure 14: The prompt of the comparative (Wang et al., 2023) baseline method.

GKP Prompt (Generation Phase):

Generate some knowledge about the input.

Input: {question}
Knowledge:

Figure 15: The prompt of the generation phase of GKP (Liu et al., 2022) baseline method.

GKP Prompt (Answer Phase):

You are doing a question answering task with a provided evidences to assist you to answer the question. According to the evidence provided and ****your own knowledge****, choose the best choice from the following options.

Here is the question:

{question}

{options}

Instructions:

Please firstly think step by step to answer the question and fill in the template: <t>...</t> with your thought. And then generate your selected option to fill in the template: <a>... in ****uppercase letter format****. For example:

---- Example Begins: ----

Step-by-step thought: <t>I think firstly we need to ... </t>

Answer: <a> E

---- Example Ends ----

Here is the retrieved evidence:

{evidence_1}

Here is my own knowledge:

{evidence_2}

There may be some conflict between the provided evidence and your own knowledge. You are a detective and you **MUST** find out which side is more reliable based on your knowledge and choose the correct answer of the question!

You have to think of the provided evidence and also fully consider your answer when disregarding the provided evidence.

---- Task Begins: ----

Step-by-step thought: <t>

Figure 16: The prompt of the answering phase of GKP (Liu et al., 2022) baseline method.

Micro-Act Prompt (Ours):

Task: Knowledge Conflict Location

You are a detective. When you ask a question, the suspects provide an evidence, which may conflict with your knowledge. Your objective is to determine the answer by identifying and locating any points of conflict.

Task Guidelines:

You have a maximum of **eight turns** to locate all conflict points. In each turn, you may choose **one** of the following actions:

Action Choices:

- knowledge_gen(target: str)** -> str: Input the target question, this function adopts LLM to generate the LLM inherent knowledge about the question. Use in format: **knowledge_gen(target="...")**.
- reason(evidence: str)** -> str: Input an evidence to generate a step-by-step reasoning path to answer given question based on that evidence. Use in format: **reason(evidence="...")**.
- decompose(reasoning_step: str)** -> list of str: This action can **ONLY** be used when one reasoning step can be decomposed for **better** analysis and benefiting the QA task. This action can break down a complex reasoning step into smaller, manageable sub-steps. Use in format: **decompose(reasoning_step="...")**.
- assert(state_left: str, state_right: str)** -> bool: Input two statements to evaluate if they conflict, returning a confidence rating. Use in format: **assert(state_left="...", state_right="...")**. This action should only be used when necessary.
- terminate()**: Signals the end of the session. Use it **ONLY** when confident in your final answer. Format: **terminate()**.

Problem-Solving Process:

You will alternate between **Thought**, **Action**, and **Observation** phases in each turn.

- In the **Thought** section, consider your current information and decide your next step.

- In the **Action** section, apply **one** of the four actions listed above.

- After each action, you will receive an **Observation** to help guide your next move.

Remember: You can take only **one** action per turn and have a total of **eight turns** to determine the final answer. When confident in your answer, use **terminate()** to end the session.

Question:

{question}
{options}

Given evidence:

...
given_evidence = "{evidence}"
...

NOTE: Begin your process now to answer the question above, filling in your action in the template: <f>...</f>.

8 turns remaining to provide the final answer. In this turn, your task is to generate **Thought** and **Action**. **Do NOT** answer question this time! Please keep your thought brief and focused, and **You MUST** choose an action from ['knowledge_gen', 'reason', 'decompose', 'assert', 'terminate'] as describe in the Task Guidelines section above! Then fill your action in the template **Action: <f>...</f>**. You can take previous steps as example. Failure to generate an action will result in strict penalties and you will lose your job!

Turn 1:

Thought: The evidence is too complex. I want to get the step-by-step reasoning path to answer the question above based on the given evidence for easier analysis.

Action: <f>reason(evidence=given_evidence)</f>

observation: Step-by-step reasoning path based on given evidence is list below:

[[cot_1]]

7 turns remaining to provide the final answer. In this turn, your task is to generate **Thought** and **Action**. **Do NOT** answer question this time! Please keep your thought brief and focused, and **You MUST** choose an action from ['knowledge_gen', 'reason', 'decompose', 'assert', 'terminate'] as describe in the Task Guidelines section above! Then fill your action in the template **Action: <f>...</f>**. You can take previous steps as example. Failure to generate an action will result in strict penalties and you will lose your job!

Turn 2:

Thought: Then I want to get the step-by-step reasoning path to answer the question above based on my inherent knowledge. To do this, I want to firstly generate the knowledge based on my inherent pre-trained knowledge.

Action: <f>knowledge_gen(target="{question}")</f>

observation: My inherent knowledge about this question is:

[[evidence_gen]]

6 turns remaining to provide the final answer. In this turn, your task is to generate **Thought** and **Action**. **Do NOT** answer question this time! Please keep your thought brief and focused, and **You MUST** choose an action from ['knowledge_gen', 'reason', 'decompose', 'assert', 'terminate'] as describe in the Task Guidelines section above! Then fill your action in the template **Action: <f>...</f>**. You can take previous steps as example. Failure to generate an action will result in strict penalties and you will lose your job!

Turn 3:

Thought: Now I want to get the step-by-step reasoning path to answer the question above based on my generate knowledgein previous step.

Action: <f>reason(evidence="[[evidence_gen]]")</f>

observation: Step-by-step reasoning path based on model knowledge is list below:

[[cot_2]]

5 turns remaining to provide the final answer. In this turn, your task is to generate **Thought** and **Action**. **Do NOT** answer question this time! Please keep your thought brief and focused, and **You MUST** choose an action from ['knowledge_gen', 'reason', 'decompose', 'assert', 'terminate'] as describe in the Task Guidelines section above! Then fill your action in the template **Action: <f>...</f>**. You can take previous steps as example. Failure to generate an action will result in strict penalties and you will lose your job!

Turn 4:

Thought: Now I have already obtained the two reasoning paths from given evidence in 'Turn 1' and from my inherent knowledge in 'Turn 3' respectively. I must locate all conflict points between them. To do so, I will compare the reasoning steps between them. First, I will check if any key reasoning steps need to be decomposed for clearer comparison. If none do, I will proceed with the reasoning step comparison to locate each conflict point.

Figure 17: The prompt of our proposed MICRO-ACT method.