Disentangling Reasoning Logic to Resolve Explicit Knowledge Conflicts

Xianda Zheng[1] Zijian Huang[1] Meng-Fen Chiang[2] Jiamou Liu[1]
Yuan Fang[3] Michael Witbrock[1] Kaiqi Zhao[4*]
[1]School of Computer Science, University of Auckland
[2]Department of Electronics and Electrical Engineering, National Yang Ming Chiao Tung University
[3]School of Computing and Information Systems, Singapore Management University
[4]Shenzhen Key Laboratory of Internet Information Collaboration, Harbin Institute of Technology (Shenzhen)
xzhe162@aucklanduni.ac.nz zhaokaiqi@hit.edu.cn

Abstract

Explicit knowledge conflicts, occurring when retrieved contexts contain contradictory information, pose a fundamental challenge for Large Language Models (LLMs) as they integrate increasingly diverse data sources. The core difficulty lies in the complexity of entangled narratives and heterogeneous conflict patterns, which frequently exceeds the reasoning capacity of standard backbone architectures. We propose **KCR** (Knowledge Conflict Reasoning), a framework that adjudicates contradictions by systematically structuring their underlying logic. KCR disentangles conflicting contexts into discrete sets of reasoning traces, utilizing a hybrid representation of text and graphs to facilitate systematic comprehension. It then employs a Reinforcement Learning with Verifiable Rewards (RLVR) paradigm to instill a reasoning policy that maximizes logical consistency while suppressing spurious paths derived from contradictory evidence. Extensive evaluations demonstrate that KCR yields substantial performance gains. Notably, a 7B model enhanced by KCR achieves adjudication capabilities that significantly outperform leading proprietary models, including GPT-4o and GPT-5.1, on complex tasks. Code is available at https://github.com/zhengxianda/KCR.

1 Introduction

The integration of knowledge from diverse sources into Large Language Models (LLMs) (Team, 2023; Touvron et al., 2023; Bai et al., 2023) often leads to conflicts, resulting in contradictory conclusions (Xu et al., 2024b; Xie et al., 2024). For instance, the classification of Pluto as a planet depends on its evolving definition, resulting in conflicting determinations over time. This challenge is exacerbated by the widespread adoption of Retrieval Augmented Generation (RAG), where models often retrieve documents containing opposing claims. Unlike hallucinations generated from internal LLMs' weights, these conflicts originate directly from the context itself. This forces LLMs into the untenable position of adjudicating between contradictory sources without a clear ground truth, leading to unreliable or inconsistent outputs.

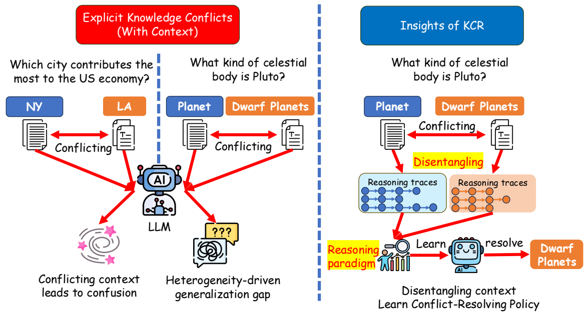

Figure 1: Challenges in resolving explicit knowledge conflicts for LLMs. (a) **Logic Entanglement**: The interweaving of contradictory evidence within dense contexts impedes the separation of valid reasoning traces, leading to confusion. (b) **Case Heterogeneity**: Static inductive principles (e.g., fixed rules) fail to generalize across the diverse nature of conflict scenarios, rendering them ineffective for robust adjudication.

Prevalent approaches for resolving explicit conflicts typically resort to *knowledge fusion* via semantic decoding (Izacard and Grave, 2021; Zhang et al., 2023). These methods often employ auxiliary embedding decoders to aggregate arbitrary context (Shi et al., 2024; Zhao et al., 2024), thereby indirectly resolving explicit knowledge conflicts within the context. However, such decoders suffer from significant performance degradation as context length increases. More recent strategies leverage the inherent reasoning capabilities of LLMs to fuse text directly (Xie et al., 2024). While promising, current methods often rely on rigid, predefined principles or complex multi-round iterative prompting (Wang et al., 2025a), which are com-

*Corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 31465–31478

putationally expensive and struggle to generalize across diverse conflict types.

We argue that these limitations persist because existing paradigms fail to address the logic entanglement and case heterogeneity posed by conflicting contexts. As illustrated in Figure 1, we identify two primary challenges. First, **Logic Entanglement** arises from the dense interweaving of contradictory evidence, which obscures underlying reasoning paths and makes it exceptionally difficult to disentangle valid logic from spurious noise. Second, **Case Heterogeneity** renders static inductive approaches (such as fixed principles) ineffective, as they lack the flexibility to generalize across the diverse spectrum of conflict scenarios. Together, these factors impede standard reasoning processes, necessitating a framework capable of structural disentanglement and robust adjudication.

To address these challenges, we propose **K**nowledge **C**onflict **R**easoning (**KCR**), a framework designed to internalize a robust conflict-resolution paradigm. KCR operates in two phases: (i) *Disentangling Conflict Logic* and (ii) *Generalizable Conflict-Resolving Policy Learning*. In the first phase, KCR decomposes entangled contexts into structured logic chains using two complementary formats: textual chains (for unstructured nuance) and local knowledge graphs (for structural clarity). In the second phase, we employ Reinforcement Learning with Verifiable Rewards (RLVR) to align the backbone LLM with a rigorous adjudication process. Specifically, we introduce rewards for *logical coherence* and *consistency*, incentivizing the model to construct valid reasoning chains while actively discriminating against spurious reasoning paths derived from contradictory contexts.

Our contributions are summarized as follows:

- To the best of our knowledge, KCR is the first framework designed to resolve Explicit Knowledge Conflicts by structurally disentangling reasoning traces and enhancing the intrinsic adjudication abilities of backbone LLMs.

- We propose a novel training paradigm that utilizes reinforcement learning to enforce logical consistency and coherence, allowing models to robustly navigate contradictory information without relying on static heuristics.

- By leveraging both unstructured text and structured local knowledge graphs, KCR effectively decomposes complex conflict logic, overcoming the "Logic Entanglement" bottleneck.

- Empirical evaluations demonstrate that KCR enables 7B-parameter models to outperform SOTA frameworks equipped with top-tier closed-source LLMs (e.g., GPT-5.1), verifying the efficacy of reasoning-driven resolution.

2 Problem Definition

In the setting of explicit knowledge conflicts, the input typically consists of a query q and a pair of conflicting candidate answers, A_1 and A_2. Formally, following the definition in ConflictQA (Xie et al., 2024), each answer is supported by a distinct context containing relevant background and evidence, denoted as C_1 and C_2, respectively. The objective is to generate a concise final answer that correctly addresses q by adjudicating the conflict between the provided contexts (C_1, C_2) and their associated logic.

3 Methodology

We introduce KCR (Knowledge Conflict Reasoning), a framework designed to enhance LLM capabilities in resolving explicit knowledge conflicts. As illustrated in Figure 2, KCR operates in two stages: (i) **Conflict Logic Disentanglement**, which decomposes entangled contexts into discrete reasoning traces; and (ii) **Generalizable Policy Learning**, which adapts the backbone model via Reinforcement Learning with Verifiable Rewards (RLVR) to instill a robust conflict-resolution policy.

3.1 Disentangling Conflict Logic

This phase aims to extract structured reasoning traces from conflicting contexts. Specifically, KCR first identifies pivotal entities and relations within the query, then reorganizes and generates reasoning trace sets corresponding to each candidate answer, leveraging both unstructured (text) and structured (graph) data formats.

3.1.1 Key Entity and Relation Extraction.

The purpose of obtaining the key entity and relation is to capture the query semantics. Formally, given the query q, KCR extracts the key entity e_q and relation r_q. For implementation, we utilize open-source entity labels from the popQA dataset and apply a frozen Qwen2.5-7B (Yang et al., 2024) model for zero-shot Named Entity Recognition and Rela-

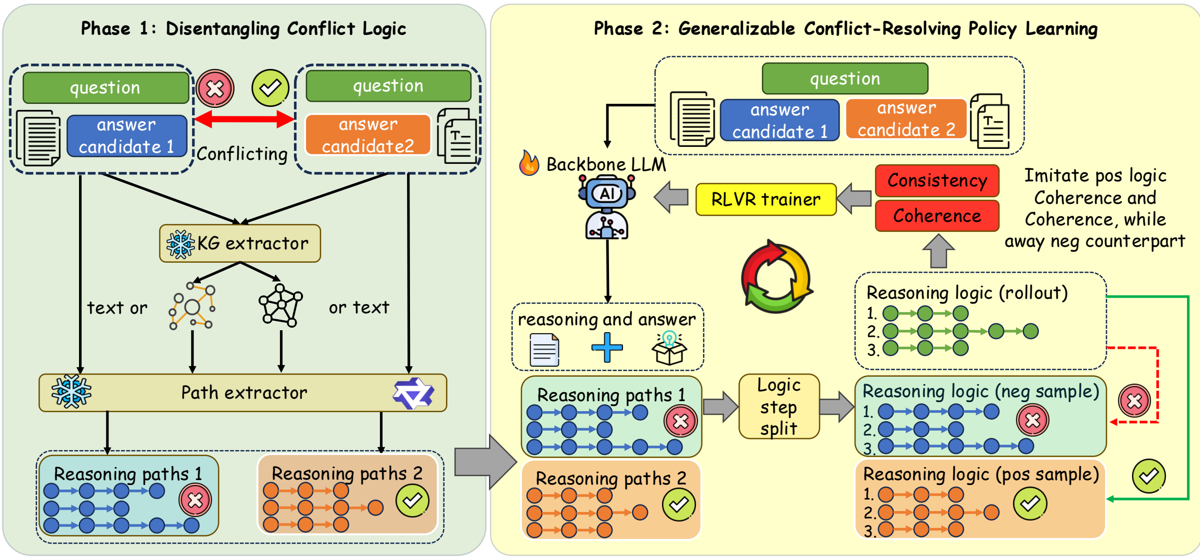

Figure 2: **Overview of the KCR framework.** Our approach structures contradictory evidence into intermediate reasoning traces, followed by a reinforcement learning stage designed to maximize logical consistency and eliminate spurious paths during conflict resolution.

tion Extraction in other cases. Extraction prompt templates are detailed in Appendix B.2.

3.1.2 Textual Reasoning Traces

With e_q and r_q extracted, the next step is to disentangle the conflicting context and reorganize the reasoning traces. Each trace captures a reasoning path related to e_q or r_q. For unstructured textual contexts, we combine answer A_i with its context C_i and prompt a frozen LLM to extract reasoning traces $\mathcal{RT}$ related to the key entities. Each trace is defined as a token sequence comprising an alternating series of entities and relations. Formally, the i-th reasoning trace from the textual context is defined as follows:

$$\mathcal{RT}_i^{(T)} = (e_1 \to r_1 \to e_2 \to \cdots \to e_n), \quad (1)$$

where e_i and r_i correspond to entities and relations appearing in the answer and context. We apply this procedure to obtain trace sets $\mathcal{RT}_1^{(T)}$ and $\mathcal{RT}_2^{(T)}$ for answers A_1 and A_2, respectively.

3.1.3 Graph Reasoning Traces

While KCR can extract reasoning traces from textual form, KCR also constructs local knowledge graphs to represent context in an interpretable and scalable manner (Edge et al., 2024). Formally, given a query entity e_q, relation r_q, candidate answer A_i, and context C_i, we instantiate a local knowledge graph $G_i = (\mathcal{E}_i, \mathcal{R}_i)$. To ensure scalable and interpretable construction, we employ a generator (e.g., GPT-4o-mini) to transform the unstructured context C_i into a structured graph format, where $\mathcal{E}_i$ and $\mathcal{R}_i$ denote the sets of entities and relations, respectively. For each entity $e \in \mathcal{E}_i$, we define a *graph reasoning trace* (p_e) as a directed path sequence:

$$p_e = e \xrightarrow{r_1} e_2 \xrightarrow{r_2} \cdots \xrightarrow{r_{n-1}} e_n, \quad (2)$$

where $e_k \in \mathcal{E}_i$ and $r_k \in \mathcal{R}_i$.

To isolate reasoning logic relevant to the conflict, we extract the set of traces $\mathcal{RT}_i^{(G)}$ that explicitly intersect with the query semantics. Specifically, we enumerate all paths originating from entities in $\mathcal{E}_i$ that contain either the query entity e_q or the query relation r_q:

$$\mathcal{RT}_i^{(G)} = \{p_e \mid (e_q, r_q \in p_e) \wedge (e_i \in \mathcal{E}_i)\}. \quad (3)$$

We apply this procedure identically to derive trace sets $\mathcal{RT}_1^{(G)}$ and $\mathcal{RT}_2^{(G)}$ corresponding to answer-context pairs (A_1, C_1) and (A_2, C_2).

3.2 Generalizable Conflict-Resolving Policy Learning

In this phase, KCR adapts the backbone LLM to internalize a robust conflict resolution paradigm via reinforcement learning. This design choice is driven by three critical insights regarding the limitations of standard supervision in conflict scenarios. First, given the **heterogeneity of conflict cases**, relying solely on Supervised Fine-Tuning

(SFT) proves inadequate. SFT often encourages the model to memorize specific conflict patterns rather than learning a generalized adjudication logic capable of handling diverse scenarios. Second, unlike mathematical or code generation tasks where intermediate reasoning steps are explicitly verifiable, the reasoning process underlying conflict resolution lacks a unique **"gold standard."** Step-level supervision is unavailable, as there are multiple valid linguistic paths to adjudicate a conflict. This inherently difficult verification motivates KCR to optimize for process-level *coherence* rather than token-level accuracy, facilitating inductive learning by contrasting valid reasoning traces against their spurious counterparts. Finally, the adjudication of contradictory information carries a heightened risk of **hallucination**, where models may erroneously blend mutually exclusive facts. This motivates the addition of a verification mechanism to penalize logical inconsistencies that standard likelihood maximization might overlook.

To address these challenges, KCR introduces two distinct reward signals designed to guide the reinforcement learning process:

- **Logic Coherence Reward:** Incentivizes the LLM to align its generation with the structural logic of correct reasoning traces, while actively diverging from the fallacious logic of incorrect contexts.
- **Logic Consistency Reward:** Penalizes hallucinated or self-contradictory outputs, ensuring that the generated reasoning process and the final answer remain logically consistent with the supported evidence.

3.2.1 Logic Coherence Reward

The core objective of logical coherence is to guide the backbone LLM to approximate the logical structure present in valid contexts while diverging from it in contradictory ones.

Formally, given a query q and conflicting answers (A_1, A_2) with their associated contexts (C_1, C_2), KCR prompts the backbone LLM to generate an output containing both a reasoning trace $\mathcal{RT}_0$ and a concise final answer A_0. KCR then computes logic coherence scores based on the generated reasoning chain $\mathcal{RT}_0$ relative to the reference reasoning path sets $\mathcal{RT}_1$ and $\mathcal{RT}_2$ extracted in the former phase. The logic score function $l(\cdot)$ utilizes the Jensen-Shannon (JS) divergence (Lin, 1991) to measure the semantic continuity between consecutive reasoning steps. For a reasoning chain $\mathcal{R}_i$ consisting of $|\mathcal{R}_i|$ steps:

$$l(\mathcal{R}_i) = \sum_{j=0}^{len(\mathcal{R}_i)-1} \mathrm{JS}(\mathcal{D}_j || \mathcal{D}_{j+1}), \quad (4)$$

where $\mathrm{JS}(\mathcal{D}_j||\mathcal{D}_{j+1}) = \frac{1}{2}D_{\mathrm{KL}}(\mathcal{D}_j||\mathcal{M}) + \frac{1}{2}D_{\mathrm{KL}}(\mathcal{D}_{j+1}||\mathcal{M})$, with $\mathcal{M} = \frac{1}{2}(\mathcal{D}_j + \mathcal{D}_{j+1})$ and $D_{\mathrm{KL}}(\cdot)$ denoting the Kullback-Leibler divergence(Kullback and Leibler, 1951). The semantic distribution $\mathcal{D}_j$ is derived from the normalized distribution of i-th embedding vector $\mathbf{R}_j$ of $\mathcal{R}$,

$$\mathcal{D}_j = \mathrm{softmax}(\frac{\mathbf{R}_j - \mu(\mathbf{R}_j)}{\sigma(\mathbf{R}_j)}). \quad (5)$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ denote the mean and standard deviation of the embedding vector, respectively.

Without loss of generality, let A_1 denote the correct (ground-truth) answer and A_2 the incorrect counterpart. KCR defines the discrete **Logic Coherence Reward** ($\mathrm{R}_{\mathrm{coh}}$) as a binary signal that rewards the model if the logical structure of the generated trace $\mathcal{RT}_0$ aligns closer to the correct reference set $\mathcal{RT}_1$ than the incorrect set $\mathcal{RT}_2$:

$$\mathrm{R}_{\mathrm{coh}} = \begin{cases} 1, & \text{if } \Delta l(\mathcal{RT}_0, \mathcal{RT}_1) < \Delta l(\mathcal{RT}_0, \mathcal{RT}_2) \\ 0, & \text{otherwise,} \end{cases} \quad (6)$$

where $\Delta l(X, Y) = |l(X) - l(Y)|$. Alternatively, a continuous variant of Logic Coherence Reward is:

$$\hat{\mathrm{R}}_{coh} = \Delta l(\mathcal{RT}_0, \mathcal{RT}_2) - \Delta l(\mathcal{RT}_0, \mathcal{RT}_1). \quad (7)$$

In practice, we find the discrete formulation yields more stable convergence during optimization.

3.2.2 Logic Consistency Reward

The Logic Consistency Reward addresses the observation that conflicting contexts often induce hallucinations, where models generate entangled reasoning that mixes contradictory content (Liu et al., 2025). This reward enforces holistic consistency between the generated reasoning process and the final conclusion.

Formally, KCR computes the similarity between the generated answer A_0 and the candidates A_1/A_2, as well as between the generated trace $\mathcal{RT}_0$ and the references $\mathcal{RT}_1/\mathcal{RT}_2$. The reward is granted if and only if A_0 and $\mathcal{RT}_0$ consistently align with the same candidate side, i.e., (A_1,$\mathcal{RT}_1$) or (A_2,$\mathcal{RT}_2$).

The discrete **Logic Consistency Reward** (R_{con}) is defined as:

$$R_{con} = \begin{cases} 1, & \text{if } (S_{RT}^{0,1} > S_{RT}^{0,2}) \wedge (S_A^{0,1} > S_A^{0,2}) \\ 1, & \text{if } (S_{RT}^{0,2} > S_{RT}^{0,1}) \wedge (S_A^{0,2} > S_A^{0,1}) \\ 0, & \text{otherwise}, \end{cases} \quad (8)$$

where $S_{RT}^{i,j} = S(\mathcal{RT}_i, \mathcal{RT}_j)$ and $S_A^{i,j} = S(A_i, A_j)$. $S(\cdot,\cdot)$ denotes the normalized Levenshtein similarity (Haldar and Mukhopadhyay, 2011), with higher scores indicating greater string similarity, chosen for its computational efficiency at the token level. Alternatively, we formulate a continuous variant of the Logic Consistency Reward as a margin-based metric. This formulation quantifies the magnitude of the alignment differential between competing reasoning traces and candidate answers:

$$\hat{R}_{con} = \left| S_{RT}^{0,1} - S_{RT}^{0,2} \right| + \left| S_A^{0,1} - S_A^{0,2} \right|, \quad (9)$$

where $S_{RT}^{0,i}$ and $S_A^{0,i}$ denote the similarity scores of the generated output against the i-th candidate reasoning trace and answer, respectively.

Note that ground-truth labels are not utilized for the consistency reward. This prevents the model from overfitting to correct answers while ignoring flawed reasoning traces, ensuring true process-outcome alignment. Token-level similarity also helps reduce computational overhead.

3.2.3 RLVR Training Procedure

After deriving the reward signals, KCR optimizes the backbone LLM as a policy π_θ using Group Relative Policy Optimization (GRPO) (Shao et al., 2024). We select discrete variants for both Logic Coherence and Logic Consistency rewards. This design choice is critical for batch-wise RLVR methods, as discrete signals significantly enhance discriminability during group sampling and stabilize candidate selection compared to continuous values. Furthermore, we enforce strict format constraints to ensure structural validity while maintaining the original ground-truth correctness signal.

Formally, for each query q, we sample a group of outputs $\{o_1, \ldots, o_G\}$ from the reference policy $\pi_{\theta_{old}}$. The cumulative reward for the i-th output is aggregated as $r_i = R_{coh} + R_{con} + R_{correct}$. To stabilize optimization, we compute the advantage score $\hat{A}_{i,t}$ by normalizing rewards within the group:

$$\hat{A}_{i,t} = \frac{r_i - \mu(\{r_j\}_{j=1}^G)}{\sigma(\{r_j\}_{j=1}^G) + \epsilon}, \quad (10)$$

where μ and σ denote the group mean and standard deviation, respectively, and δ is a small constant for numerical stability. The policy ratio $r_{i,t}(\theta)$ between the current and reference policy is defined as:

$$r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t} \mid q, o_{i,<t})}, \quad (11)$$

where o is the output and π is the backbone LLM as policy model and θ the weight of backbone LLM. The final GRPO objective maximizes the surrogate loss while penalizing significant deviations from the reference policy via clipping:

$$\mathcal{J}(\theta) = \mathbb{E}_{(q,A)\sim\mathcal{D},\ \{o_i\}_{i=1}^G \sim \pi_{\theta_{old}}} \Bigg[\frac{1}{G} \sum_{i=1}^{G} \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \min\Big(r_{i,t}(\theta)\hat{A}_{i,t}, \text{clip}(r_{i,t}(\theta), \varepsilon)\hat{A}_{i,t} \Big) \Bigg], \quad (12)$$

where clip($\cdot$, ε) is the standard clip operation with ε as the clipping hyperparameter.

4 Experiments

We evaluate KCR through quantitative and qualitative experiments. First, we validate its effectiveness across various backbone LLMs in scenarios involving conflicting knowledge in long contexts. We then perform ablation studies to assess the logic coherence and consistency rewards, comparing KCR against Supervised Fine-Tuning (SFT) and standard GRPO. Finally, case analyses demonstrate the improved reasoning capabilities.

4.1 Experimental Setting

Datasets. We utilize two public datasets specifically adapted for the Explicit Knowledge Conflict task: **popQA** and **strategyQA** (Xie et al., 2024). PopQA involves conflicts between distinct entities, whereas strategyQA features general interrogatives with conflicting Boolean answers. We partition data into training, validation, and test sets with an 8:1:1 ratio. Table 1 summarizes the statistics.

Hyperparameter Settings and Training Cost. We employ consistent hyperparameters across experiments, training for 10 epochs with a batch size of 32. GPT-4o-mini powers both local knowledge graph construction and the LLM-as-Judge module, while `BAAI/bge-large-en-v1.5` generates sentence embeddings. Training is accelerated using the Verl framework with Flash Attention. All models are trained on 8×A100 80GB SXM GPUs.

Dataset	popQA	strategyQA
Questions	7,198	1,244
Avg. question tokens	11.73	12.09
Avg. answer tokens	23.47	1.00
Avg. context tokens	498.65	397.69
Relative token ratio	21.24	397.69

Table 1: Dataset statistics. Relative token ratio is the number of context tokens divided by answer tokens.

The approximate training time is 72 GPU hours for 7B/8B models and 25 GPU hours for 3B models.

Evaluation Metrics. We adopt both semantic-level and token-level evaluations. Specifically, For semantic similarity (ACC_L), we use an LLM-as-Judge (Zheng et al., 2023) powered by GPT-4o-mini (prompts in Appendix). For token-level accuracy, we report Exact Match (ACC_{EM}) and Cover Exact Match (ACC_{CEM}). The judge prompts are detailed in the Appendix B.5.

Baselines. We evaluate KCR using LLaMA and Qwen backbones. We compare against five state-of-the-art (SOTA) frameworks for inter-context knowledge conflicts, including four decoding strategies: CAD (Shi et al., 2024), CUAD (Zhao et al., 2024), AdaCAD (Wang et al., 2025b), and CoCoA (Khandelwal et al., 2025). Additionally, we include Astute RAG (Wang et al., 2025a), which employs explicit knowledge fusion.

4.2 Overall Performance Comparison

The main quantitative results for the Explicit Knowledge Conflict task are presented in Table 2, from which several key observations can be drawn. First, **KCR** consistently and significantly improves the performance of backbone LLMs across datasets of varying scales. This is particularly evident in ACC_L, which emphasizes semantic-level answer accuracy, suggesting KCR enhances true understanding rather than mere lexical matching. Notably, both text-based and graph-based reasoning paths contribute to these improvements. This indicates that KCR effectively guides backbone LLMs in conflict adjudication, even when the models have not been explicitly pre-trained on graph-structured inputs. While standard decoder-based baselines struggle to resolve explicit knowledge conflicts effectively, KCR demonstrates universal performance gains. Additionally, high-complexity frameworks like Astute RAG often fail on smaller (3B- and 7B-scale) models due to excessive reasoning demands. In contrast, KCR successfully instills the conflict-resolution paradigm into smaller backbones, enabling a 7B model to outperform much larger, closed-source systems.

4.3 Impact of Logical Coherence

In KCR, the logical coherence of the reasoning process serves as a proxy for rigor, where lower coherence scores (in JS divergence) indicate tighter logical steps, while higher scores reflect larger inferential leaps. It is worth noting that questions requiring excessive rigor or extreme logical leaps are inherently more difficult. As shown in Figure 3, KCR notably improves the handling of conflicting answers on the more challenging popQA dataset. On the simpler StrategyQA, KCR improves performance specifically on questions that demand rigorous logic. This suggests that KCR is particularly advantageous for problems requiring complex, multi-step adjudication.

4.4 Ablation Study

We evaluate the individual contributions of KCR's components on popQA using the Qwen 2.5–7B backbone. This section evaluates KCR's components on popQA and compares its performance to alternative fine-tuning approaches using the Qwen 2.5–7B backbone. Figure 4 illustrates that while each component contributes to the model's overall capability, the **Logic Coherence Reward** is the principal driver of improvement. The full KCR framework effectively integrates these elements to form a robust system for resolving knowledge conflicts, significantly outperforming approaches that rely solely on SFT or vanilla reinforcement learning methods like GRPO.

4.5 Qualitative Study

Our qualitative analysis indicates that models enhanced by KCR exhibit a remarkably structured reasoning process. They systematically analyze conflicting points sequentially, strictly adhering to the order of evidence presentation before deriving a final conclusion. In contrast, baseline models without KCR tend to generate conclusions more directly, often failing to articulate a clear, step-by-step logical chain. This observable difference in reasoning style demonstrates that KCR instills a more methodical and logically coherent adjudication process in the backbone model. Detailed case studies, including prompts and outputs, are provided in the Appendix C.

Model Family	Method / Variant	popQA			strategyQA		
		ACC_L	ACC_{EM}	ACC_{CEM}	ACC_L	ACC_{EM}	ACC_{CEM}
Direct Generation (Without Context)							
	Llama 3.2-3B	0.1764	0.0056	0.1736	0.4640	0.0000	0.3200
	Qwen 2.5-7B	0.1250	0.0889	0.0944	0.6880	0.6720	0.6720
	Qwen 3-8B (Thinking)	0.1736	0.1306	0.1444	0.7600	0.7200	0.7680
	GPT-4o-mini	0.3042	0.2431	0.2542	0.7440	0.7440	0.7440
	GPT-5.1 (Thinking)	0.4944	0.1389	0.4819	0.6640	0.0000	0.6720
General Baselines (With Context)							
	Llama 3.2-1B	0.3000	0.0139	0.2625	0.4480	0.0160	0.0640
	Llama 3.1-8B	0.5653	0.0292	0.5278	0.4480	0.0000	0.4000
	Qwen 2.5-1.5B	0.3986	0.2569	0.3764	0.5040	0.3840	0.3920
	Qwen 2.5-3B	0.5167	0.4333	0.5042	0.5440	0.5360	0.5360
	CUAD (Default)	0.3431	0.2958	0.3056	0.5040	0.5040	0.5040
	ADACAD (Default)	0.3431	0.2958	0.3056	0.5040	0.5040	0.5040
	COCOA (Default)	0.3431	0.2958	0.3056	0.5040	0.5040	0.5040
	ASTUTE RAG (GPT-4o)	0.5486	0.5056	0.6097	0.5920	0.5760	0.5840
	ASTUTE RAG (GPT-5.1)	0.6694	0.5819	0.7083	0.5760	0.2160	0.5920
Backbone: Llama 3.2-3B							
	Directly Use Backbone	0.4958	0.3375	0.4944	0.5120	0.3920	0.4240
	CAD Decoding	0.2972	0.3111	0.3222	0.4640	0.3280	0.4880
	ASTUTE RAG	0.3972	0.0208	0.7458	0.4640	0.3840	0.4320
	KCR ("Text")	0.8417	0.8014	0.8292	0.7760	0.7680	0.7680
	KCR ("Graph")	**0.8750**	**0.8361**	**0.8778**	**0.8240**	**0.7760**	**0.7760**
	Improvement vs Backbone	*+37.9%*	*+49.9%*	*+38.3%*	*+31.2%*	*+38.4%*	*+35.2%*
Backbone: Qwen 2.5-7B							
	Directly Use Backbone	0.4847	0.4181	0.4778	0.5920	0.5920	0.5920
	CAD Decoding	0.4028	0.1750	0.4361	0.3120	0.0000	0.3440
	ASTUTE RAG	0.4375	0.3889	0.4556	0.3680	0.3600	0.3600
	KCR ("Text")	**0.8667**	**0.8250**	**0.8653**	**0.8320**	**0.7920**	**0.7920**
	KCR ("Graph")	0.8014	0.7569	0.7778	0.7520	0.6720	0.6720
	Improvement vs Backbone	*+38.2%*	*+40.7%*	*+38.8%*	*+24.0%*	*+20.0%*	*+20.0%*
Backbone: Qwen 3-8B (Thinking)							
	Directly Use Backbone	0.5431	0.5056	0.5472	0.5600	0.5600	0.5600
	KCR ("Text")	0.6861	0.6458	0.6583	**0.6400**	**0.6400**	**0.6400**
	KCR ("Graph")	**0.6958**	**0.6556**	**0.6903**	0.5600	0.5600	0.5600
	Improvement vs Backbone	*+15.3%*	*+15.0%*	*+14.3%*	*+8.0%*	*+8.0%*	*+8.0%*
Backbone: Qwen 3-8B							
	Directly Use Backbone	0.5431	0.5056	0.5500	0.5360	0.5360	0.5360
	CAD Decoding	0.4028	0.1750	0.4361	0.3120	0.0000	0.3440
	ASTUTE RAG	0.5028	0.4417	0.6764	0.5040	0.4880	0.7040
	KCR ("Text")	0.7472	0.7139	0.7361	0.5861	0.5444	0.5806
	KCR ("Graph")	**0.9028**	**0.8764**	**0.9000**	0.5278	0.4889	0.5319
	Improvement vs Backbone	*+36.0%*	*+37.1%*	*+35.0%*	*+5.0%*	*+0.8%*	*+4.5%*
Overall Improvement vs SOTA		**+19.7%**	**+24.3%**	**+12.0%**	**+7.2%**	**+4.8%**	**+2.4%**

Table 2: **Overall comparison on the Inter-Context Knowledge Conflict task. Bold** indicates the best performance within each specific backbone group or overall. "TEXT" and "GRAPH" denote KCR reasoning traces derived from textual and graph-based formats, respectively. The improvement rows highlight the relative gain of the best KCR variant over its corresponding backbone baseline, while the final row compares KCR against the strongest state-of-the-art framework. All our experimental results are averaged over three independent runs.

5 Related Work

5.1 Implicit Knowledge Conflicts

Implicit Knowledge Conflicts, including both Context-Memory and Inter-Memory conflicts, arise from discrepancies between externally context (i.e., retrieved documents) and the model's parameterized knowledge, or among the model's parameterized knowledge (Xu et al., 2024b). The typical scenario is in Retrieval-Augmented Generation (RAG) frameworks, where the model must reconcile retrieved documents with its own stored knowledge (Jin et al., 2024; Jung et al., 2024; Wang et al., 2024). Prevailing strategies for mitigating such conflicts include model-centric approaches like fine-tuning (Li et al., 2023; Xue et al., 2023) and knowledge editing (Lee et al., 2022), alongside context-centric methods such as information compression and fusion (Jung et al., 2024; Wang et al., 2024). Implicit Knowledge Conflicts fundamentally do not require models to understand complex conflict logic, and make choices within internalized

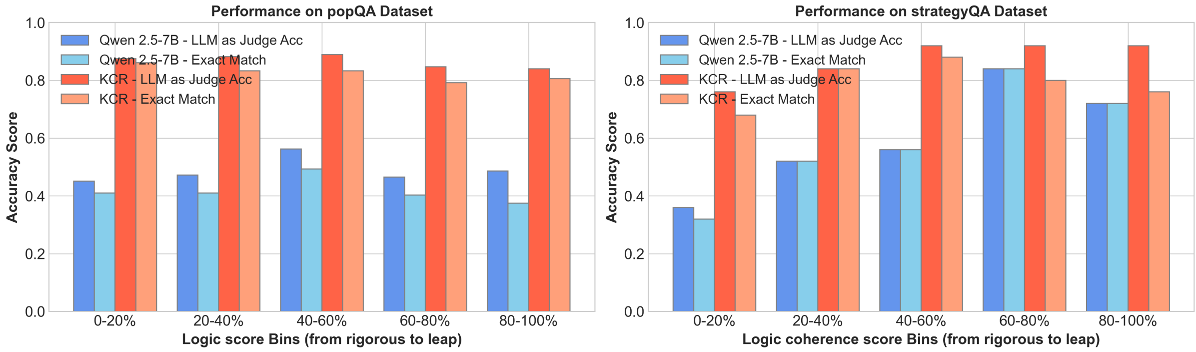

Figure 3: **Impact of Reasoning Coherence.** Analysis of model performance across varying degrees of logical coherence (JS Divergence). Lower scores indicate stricter logical adherence.

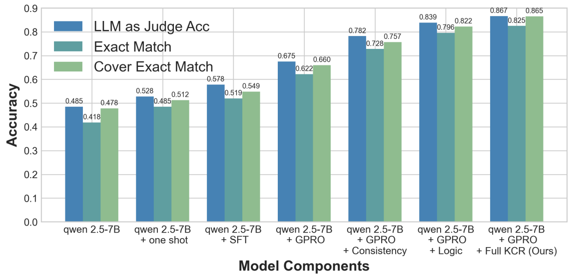

Figure 4: **Ablation Study.** KCR excels in scenarios requiring rigorous logical chains (popQA) compared to looser reasoning tasks. (b) validates that both Logic Coherence and Consistency rewards are essential for robust conflict adjudication.

knowledge and context.

5.2 Explicit Knowledge Conflicts

Explicit Knowledge Conflicts typically refer to contradictions that arise within the provided context, most commonly when two conflicting pieces of information are presented as separate contextual inputs (Xu et al., 2024b). Typical methods in resolving Explicit Knowledge Conflicts involve re-pairing the conflicting contexts at the sentence level, obtaining soft labels through supervised signals, and then feeding them into the FiD decoder to generate an answer (Izacard and Grave, 2021; Zhang et al., 2023). More recent methods shifted to semantic distribution-level fusion, process each conflicting source document independently to generate separate answers and semantic distributions, which are then fused by a specialized decoder to synthesize a unified output (Shi et al., 2024; Zhao et al., 2024; Wang et al., 2025b; Khandelwal et al., 2025). Recent approaches attempt to directly perform knowledge fusion using powerful closed-source LLMs, employing complex iterative heuristic frameworks to address this challenge (Wang et al., 2025a). These modeling attempts have not reduced the difficulty of understanding conflicting contexts, making it still challenging to fully leverage LLMs' reasoning abilities to resolve explicit knowledge conflicts.

5.3 LLMs Behaviors in Knowledge Conflicts

Recent studies confirm that LLMs possess a baseline capacity for resolving simple knowledge conflicts (Xie et al., 2024). However, these investigations have also uncovered significant behavioral biases. For instance, LLMs tend to favor information characterized by high verbosity or apparent evidential support, and they exhibit difficulty reasoning over "counter-memory" facts even when paired with supporting context (Xu et al., 2024a; Hong et al., 2024; Neeman et al., 2023). Crucially, prior research has concentrated on characterizing these observable behaviors, while largely neglecting the role of the model's underlying logical reasoning abilities in systematically resolving such contradictions.

6 Conclusion

We introduce KCR, a framework designed to resolve explicit knowledge conflicts by distilling structured reasoning paths from contradictory evidence. By optimizing for logical coherence and consistency through a novel learning paradigm, KCR enables models to systematically adjudicate conflicting information. Empirical results across diverse backbones demonstrate that KCR significantly improves a model's capacity to navigate factual contradictions, yielding substantial and robust performance gains.

Acknowledgments

This research is partly supported by the Shenzhen Science and Technology Program No. SYSPG20241211173609009, and National Science and Technology Council (NSTC), Taiwan (Grants: NSTC-114-2222-E-A49-004, and NSTC-114-2639-E-A49-001-ASP).

Limitations

While KCR demonstrates significant improvements in resolving explicit knowledge conflicts, we identify several limitations that merit further investigation.

Computational Complexity and Latency. The *Conflict Logic Disentanglement* phase relies on structured context via Local Knowledge Graphs (LKGs). Our current implementation utilizes proprietary models (e.g., GPT-4o-mini) for graph extraction, which introduces a dependency on external APIs and increases latency compared to purely embedding-based retrieval. Furthermore, the RLVR training paradigm requires sampling multiple reasoning chains (group sampling) to compute coherence and consistency rewards. This incurs a higher computational cost during training relative to standard Supervised Fine-Tuning (SFT).

Modeling Scope of Conflicts. Our framework is explicitly designed to address inter-context conflicts (Explicit Knowledge Conflicts), where retrieved documents contain contradictory information. We do not explore implicit conflicts (Context-Parametric Memory conflicts), where retrieved evidence contradicts the model's internal knowledge. Additionally, our task formulation focuses on dyadic conflict resolution (binary adjudication). Real-world scenarios involving multi-party conflicts or sources with varying degrees of partial correctness may require more complex adjudication logic than our current implementation.

Linguistic and Benchmark Scope. Our experiments and prompt designs are restricted to English benchmarks. The efficacy of the logic disentanglement phase and the graph extraction prompts in multilingual settings or for low-resource languages remains an open question. Future work is required to assess the cross-lingual generalizability of structured reasoning in conflict resolution.

Reward and Noisy. Verifiable rewards are less reliable in open-domain or subjective conflicts without clear ground truth, which may introduce ambiguity in learning. Our evaluation is limited to controlled settings, and extending KCR to adversarial or noisy environments is an important direction for future work.

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, and 29 others. 2023. Qwen technical report. *CoRR*, abs/2309.16609.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph RAG approach to query-focused summarization. *CoRR*, abs/2404.16130.

Rishin Haldar and Debajyoti Mukhopadhyay. 2011. Levenshtein distance technique in dictionary lookup methods: An improved approach. *CoRR*, abs/1101.1232.

Giwon Hong, Jeonghwan Kim, Junmo Kang, Sung-Hyon Myaeng, and Joyce Jiyoung Whang. 2024. Why so gullible? enhancing the robustness of retrieval-augmented models against counterfactual noise. In *NAACL Findings*, pages 2474–2495.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *EACL*, pages 874–880.

Zhuoran Jin, Pengfei Cao, Yubo Chen, Kang Liu, Xiaojian Jiang, Jiexin Xu, Qiuxia Li, and Jun Zhao. 2024. Tug-of-war between knowledge: Exploring and resolving knowledge conflicts in retrieval-augmented language models. In *LREC/COLING*, pages 16867–16878.

Dongwon Jung, Qin Liu, Tenghao Huang, Ben Zhou, and Muhao Chen. 2024. Familiarity-aware evidence compression for retrieval augmented generation. *CoRR*, abs/2409.12468.

Anant Khandelwal, Manish Gupta, and Puneet Agrawal. 2025. Cocoa: Confidence and context-aware adaptive decoding for resolving knowledge conflicts in large language models. *CoRR*, abs/2508.17670.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Kyungjae Lee, Wookje Han, Seung-won Hwang, Hwaran Lee, Joonsuk Park, and Sang-Woo Lee. 2022. Plug-and-play adaptation for continuously-updated QA. In *ACL Findings*, pages 438–447.

Daliang Li, Ankit Singh Rawat, Manzil Zaheer, Xin Wang, Michal Lukasik, Andreas Veit, Felix X. Yu, and Sanjiv Kumar. 2023. Large language models with controllable working memory. In *ACL Findings*, pages 1774–1793.

Jianhua Lin. 1991. Divergence measures based on the shannon entropy. *IEEE Trans. Inf. Theory*, 37(1):145–151.

Siyi Liu, Kishaloy Halder, Zheng Qi, Wei Xiao, Nikolaos Pappas, Phu Mon Htut, Neha Anna John, Yassine Benajiba, and Dan Roth. 2025. Towards long context hallucination detection. In *NAACL Findings*, pages 7827–7835.

Ella Neeman, Roee Aharoni, Or Honovich, Leshem Choshen, Idan Szpektor, and Omri Abend. 2023. Disentqa: Disentangling parametric and contextual knowledge with counterfactual question answering. In *ACL*, pages 10056–10070.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. 2024. Trusting your evidence: Hallucinate less with context-aware decoding. In *NAACL*, pages 783–791.

OpenAI Team. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Fei Wang, Xingchen Wan, Ruoxi Sun, Jiefeng Chen, and Sercan Ö. Arik. 2024. Astute RAG: overcoming imperfect retrieval augmentation and knowledge conflicts for large language models. *CoRR*, abs/2410.07176.

Fei Wang, Xingchen Wan, Ruoxi Sun, Jiefeng Chen, and Sercan Ö. Arik. 2025a. Astute RAG: overcoming imperfect retrieval augmentation and knowledge conflicts for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025*, pages 30553–30571. Association for Computational Linguistics.

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 2025b. Adacad: Adaptively decoding to balance conflicts between contextual and parametric knowledge. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025*, pages 11636–11652. Association for Computational Linguistics.

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 *ICLR*.

Rongwu Xu, Brian S. Lin, Shujian Yang, Tianqi Zhang, Weiyan Shi, Tianwei Zhang, Zhixuan Fang, Wei Xu, and Han Qiu. 2024a. The earth is flat because...: Investigating llms' belief towards misinformation via persuasive conversation. In *ACL*, pages 16259–16303.

Rongwu Xu, Zehan Qi, Zhijiang Guo, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. 2024b. Knowledge conflicts for llms: A survey. In *EMNLP*, pages 8541–8565.

Boyang Xue, Weichao Wang, Hongru Wang, Fei Mi, Rui Wang, Yasheng Wang, Lifeng Shang, Xin Jiang, Qun Liu, and Kam-Fai Wong. 2023. Improving factual consistency for knowledge-grounded dialogue systems via knowledge enhancement and alignment. In *EMNLP Findings*, pages 7829–7844.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

Yunxiang Zhang, Muhammad Khalifa, Lajanugen Logeswaran, Moontae Lee, Honglak Lee, and Lu Wang. 2023. Merging generated and retrieved knowledge for open-domain QA. In *EMNLP*, pages 4710–4728.

Zheng Zhao, Emilio Monti, Jens Lehmann, and Haytham Assem. 2024. Enhancing contextual understanding in large language models through contrastive decoding. In *NAACL*, pages 4225–4237.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *NeurIPS*.

Appendix

A Checklist

A.1 Ethics Statement

All datasets used in this research are publicly available and were sourced from previous studies that have undergone appropriate ethical review. Our work did not involve the collection of any new data from human subjects. We have adhered to all data usage agreements and licenses associated with these pre-existing datasets.

A.2 Reprodicibility Statement

We are committed to making our research reproducible. All datasets used in this study are publicly available, and we provide detailed descriptions and sources in our experimental setup section.

A.3 Potential Risks

We do not identify significant risks associated with our approach, as it relies solely on open-source data and models and follows standard research practices.

A.4 Usage of LLM

This paper primarily employs large language models (LLMs) to refine overall writing quality, with a particular focus on eliminating incorrect expressions, minimizing grammatical errors, and enhancing clarity, coherence, and readability to ensure the text meets ACL standards.

B Prompt Template

B.1 Reasoning and Answer Extraction

Extending KCR to new backbone models primarily involves defining regex patterns to isolate intermediate reasoning traces from the final answer. The patterns currently employed for the Qwen and LLaMA series are provided in the boxes below.

Patterns for extracting thinking and answer from Qwen

```
thinking_patterns = r"<think>(.*?)</
    think>"

answer_patterns = r"<answer>(.*?)</
    answer>"
```

Patterns for extracting thinking and answers from LLaMA

```
thinking patterns = r'(?:\*+)?\s*
    Thinking Process\s*:?\s*(?:\*+)?\s
    *(.*?)\s*(?:\*+)?\s*Final Answer\s
    *:?\s*(?:\*+)?'

answer_patterns = r"\*\*Final Answer
    :\*\*\s*(.+)"
answer_patterns = r"Final Answer:\s
    *(.+)"
answer_patterns = r"\*\*Correct Answer
    :\*\*\s*(.+)"
answer_patterns = r"Correct Answer:\s
    *(.+)"
```

B.2 Prompts for KG Extraction

In the KG extraction phase, we utilize GPT-4o-mini as the extractor. The system and user prompts are structured to ensure the output adheres to a strict JSON format containing entities and triples.

Prompts for extracting thinking and answers using GPT-4o-mini

```
{"role": "system", "content": "You are
     an expert agent specialized in
    build Knowledge Graphs."},
{"role": "system", "content": "Extract
     a knowledge graph from the
    following document, return a json
    file within one line."},
{"role": "user", "content": """You
    must generate the output in a JSON
     containing a list with JSON
    objects having the following keys:
     "entities", "triples". The "
    entities" must contain the text of
     the extracted entities from
    document, the "triples" must
    contain the python dicts that
    composed of key "subject", key "
    relation" and key "object". in
    this dict."""},
{"role": "user", "content": f"document
    : {document}"},
{"role": "user", "content": f"key
    subject: {subject}\n\n"},
{"role": "user", "content": f"key
    relation: {relation}\n\n"},
{"role": "user", "content": "knowledge
     graph: \n"},
```

B.3 Prompts for Qwen Series Backbone

The following prompts guide the backbone models to produce outputs following a structured "Reasoning + Final Answer" format. This allows the model to explicitly disentangle conflicting evidence before making a decision.

Prompts for extracting thinking and answers from Qwen

```
{"role": "system", "content": "You are
     helpful AI system.\n\n"},
{"role": "user", "content": "A
    question is given and its two
    candidate answers, along with
    their context. Only one of the two
     is correct. You need to choose
    the correct one. Use English Only
    !\n\n"},
{"role": "user", "content": f"question
    : {question} \n"},
{"role": "user", "content": f"First
    answer: {answer1} \n\n"},
{"role": "user", "content": f"Context
    with first answer: {context1} \n\n
    "},
{"role": "user", "content": f"Second
    answer: {answer2} \n\n"},
{"role": "user", "content": f"Context
    with second answer: {context2}\n\n
    "},
{"role": "user", "content": "First
    output the thinking process in <
    think> </think> and final answer
    using single entity in <answer> </
    answer> tags."},
```

B.4 Prompts for LLaMA Series Backbone

This subsection presents the prompt design for the LLaMA backbone. The following prompts guide the backbone models to produce outputs following a structured "Reasoning + Final Answer" format. This allows the model to explicitly disentangle conflicting evidence before making a decision.

Prompts for extracting thinking and answers from LLaMA

```
{"role": "system", "content": "You are
     helpful AI system.\n\n"},
{"role": "user", "content": "A
    question is given and its two
    candidate answers, along with
    their context and evidence. Only
    one of the two is correct. You
    need to choose the correct one."},
{"role": "user", "content": "You have
    four key principles that should be
     followed: a) Check for explicit
    contradiction or mutual
    exclusivity; b) Analyze
    implications or presuppositions; c
    ) Apply world knowledge to check
    commonsense facts; d) Use context
    to ground both statements. \n\n"},
{"role": "user", "content": f"Question
    : {question} \n"},
{"role": "user", "content": f"First
    answer: {answer1} \n\n"},
{"role": "user", "content": f"Context
    with first answer: {memory1} \n\n
    "},
{"role": "user", "content": f"Evidence
     with first answer: {evidence1} \n
    \n"},
{"role": "user", "content": f"Second
    answer: {answer2} \n\n"},
{"role": "user", "content": f"Context
    with second answer: {memory2} \n\n
    "},
{"role": "user", "content": f"Evidence
     with second answer: {evidence2} \
    n\n"},
{"role": "user", "content": "Firstly,
    output the whole thinking process
    in <think> </think> and final
    answer using single entity in <
    answer> </answer> tags. Use
    English to answer the question
    only!\n\n"},
{"role": "user", "content": "Thinking
    process and Final answer: "},
```

B.5 Prompts for LLM-as-Judge

To evaluate model performance, we use a judge prompt that verifies the semantic alignment between the predicted answer and the ground truth reference.

LLM-as-judge prompt

```
prompt = '''Given a Question and its
    Golden Answer, verify whether the
    Predicted Answer is correct. The
    prediction is correct if it fully
    aligns with the meaning and key
    information of the Golden Answer.
    Respond with True if the
    prediction is correct and False
    otherwise.

     Question: {question}

     Golden Answer: {reference}

     Predicted Answer: {prediction}

     '''
```

C Extended Qualitative Studies

In this section, we present some qualitative detail results on the popQA dataset. We compare our method with the backbone model Qwen 2.5-7B. Here are some cases that demonstrate improved reasoning skills for resolving conflicts in knowledge using KCR.

Case 1 input and ground truth label from dataset popQA

```
question: Who was the composer of
    Trombone Concerto? \newline
```

```
ground truth: Rimsky-Korsakov \newline
\newline

conflicting answer 1: The composer of
    Trombone Concerto is Johann Georg
    Albrechtsberger. \newline
conflicting context 1: Johann Georg
    Albrechtsberger was an Austrian
    composer and music theorist who
    lived from 1736 to 1809. He was a
    prominent figure in the Viennese
    music scene during the late 18th
    century and was known for his
    expertise in counterpoint and
    harmony. Albrechtsberger composed
    a variety of works, including
    symphonies, chamber music, and
    sacred music, but he is perhaps
    best known for his Trombone
    Concerto, which is still performed
     and recorded today. Johann Georg
    Albrechtsberger (1736-1809) was an
     Austrian composer and music
    theorist. He is well known for his
     contributions to brass and wind
    music, including a famous Trombone
     Concerto that he composed in the
    late 1700s. This piece features a
    solo trombone accompanied by an
    orchestra and showcases
    Albrechtsberger's skill for
    writing virtuosic music for wind
    instruments. The Trombone Concerto
     has been performed and recorded
    by many notable musicians and
    orchestras around the world,
    solidifying its place in the canon
     of classical music pieces for
    trombone. With this evidence, we
    can confidently confirm that
    Johann Georg Albrechtsberger is
    indeed the composer of Trombone
    Concerto.
\newline

conflicting answer 2: Rimsky-Korsakov
    was the composer of Trombone
    Concerto. \newline
conflicting context 2: Rimsky-Korsakov
     was a prolific composer who wrote
     many works for orchestra,
    including the Trombone Concerto.
    This piece was composed in 1877
    and is considered one of the most
    challenging works in the trombone
    repertoire. It features virtuosic
    solo passages and intricate
    orchestration, showcasing Rimsky-
    Korsakov's skill as a composer.
    The concerto has been performed by
     many notable trombonists,
    including Christian Lindberg and
    Joseph Alessi, and remains a
    popular choice for soloists and
    orchestras around the world.
Trombone Concerto (Rimsky-Korsakov)
    The Concerto for Trombone and
    Military Band by Nikolai Rimsky-
    Korsakov was written in 1877. The
    concerto consists of three
    movements: an "Allegro Vivace"
    first movement, an "Andante
    Cantabile" second movement, and an
     "Allegro-Allegretto" third
    movement in the style of a march.
    The second and third movements
    conclude with cadenzas. A full
    performance of the piece lasts
    roughly ten minutes. This concerto
     was composed for a fellow marine
    officer Leonov and premiered at a
    garrison concert at Kronstadt on
    16 March 1878. The American
    premiere took place in June, 1952
    at The Mall in Central Park, New
    York
```

Case 1 conflicting reasoning paths sets

```
Conflicting path set 1: \newline
1. Rimsky-Korsakov -> composer ->
    wrote -> Trombone Concerto \
    newline
2. Rimsky-Korsakov -> composer ->
    composed -> 1877 -> Trombone
    Concerto \newline
3. Rimsky-Korsakov -> composer ->
    showcased -> 'virtuosic solo
    passages -> Trombone Concerto \
    newline
4. Rimsky-Korsakov -> composer ->
    showcased -> intricate
    orchestration -> Trombone Concerto
     \newline
5. Rimsky-Korsakov -> composer ->
    wrote -> for -> Trombone and
    Military Band -> Trombone Concerto
     \newline
6. Rimsky-Korsakov -> composer ->
    wrote -> in 1877 -> Trombone
    Concerto \newline
7. Leonov -> fellow marine officer ->
    commissioned -> Trombone Concerto
    \newline
8. Rimsky-Korsakov -> composer ->
    premiered -> at Kronstadt ->
    Trombone Concerto \newline
\newline
Conflicting path set 2: \newline
Johann Georg Albrechtsberger ->
    composer -> Trombone Concerto \
    newline
Johann Georg Albrechtsberger ->
    composer -> variety of works ->
    Trombone Concerto \newline
Johann Georg Albrechtsberger ->
    composer -> late 1700s -> Trombone
     Concerto \newline
Johann Georg Albrechtsberger ->
    composer -> famous Trombone
    Concerto \newline
Johann Georg Albrechtsberger ->
    composer -> virtuosic music for
    wind instruments -> Trombone
    Concerto \newline
```

```
Johann Georg Albrechtsberger ->
    composer -> classical music pieces
     for trombone -> Trombone Concerto
     \newline
```

Comparsion between the Original Qwen 2.5-7B and KCR boost Qwen 2.5-7B

```
Original Qwen 2.5-7B output: \newline
<think>The context and evidence
    provided for the second answer are
     incorrect. Nikolai Rimsky-
    Korsakov was not known for
    composing a trombone concerto. The
     context mentions a "Concerto for
    Trombone and Military Band," but
    there is no widely recognized or
    historically documented trombone
    concerto by Rimsky-Korsakov. On
    the other hand, the first answer
    provides detailed context about
    Johann Georg Albrechtsberger and
    cites evidence that strongly
    supports his authorship of the
    Trombone Concerto. Given the
    information, the correct answer is
     the first one.</think>\newline
<answer>Johann Georg Albrechtsberger</
    answer> \newline
\newline
KCR + Qwen 2.5-7B output: \newline
<think>\newline
1. The first answer attributes the
    Trombone Concerto to Johann Georg
    Albrechtsberger.\newline
2. The second answer attributes the
    Trombone Concerto to Nikolai
    Rimsky-Korsakov.\newline
3. Both answers provide context about
    the composer and the piece,
    suggesting they have different
    composers.\newline
4. Albrechtsberger lived from 1736 to
    1809, while Rimsky-Korsakov lived
    from 1844 to 1908. This time
    period does not overlap.\newline
5. The context provided for the second
     answer mentions that the Trombone
     Concerto was composed in 1877,
    which aligns with Rimsky-Korsakov'
    s lifetime.\newline
6. The first answer provides no
    specific year or context about
    when the Trombone Concerto was
    composed.\newline
</think>\newline
<answer>\newline
Rimsky-Korsakov\newline
</answer>
```