Learning Minimal Sufficient Evidence Graphs for GraphRAG via Nash-Guided Optimization

Hao Wu[1,2], Xinguo Yu[1,†], Wanqing Li[2], Hongru Sun[2], Xiao Luo[3], Wenbin Zhang[4], Sasha Nikolic[2], Shirui Pan[5], Yi Guo[6,†], Jie Yang[2,†]

[1]Central China Normal University, China [2]University of Wollongong, Australia
[3]University of Wisconsin-Madison, USA [4]Florida International University, USA
[5]Griffith University, Australia [6]University of Western Sydney, Australia
Correspondence: hwu@mails.ccnu.edu.cn, jiey@uow.edu.au [†]Corresponding Author

Abstract

GraphRAG enhances Large Language Model reasoning by organizing retrieved knowledge into structured evidence graphs, enabling inference over connected evidence rather than isolated text fragments. Yet, existing GraphRAG methods often either miss query-critical links or introduce noisy or conflicting evidence that distracts reasoning. We propose CONSIST, a **CO**nflict-aware **Na****S**h-optimized **MI**nimal **S**ufficien**T** graph learning algorithm that regulates evidence graphs to resolve conflicts during reasoning. Specifically, we cast graph learning as an exploration–editing process, where a candidate graph is expanded via structural and semantic connections and then refined by pruning unreliable edges. We further introduce Gumbel-based edge editing, framing it as a Nash-guided bargaining process, where current and future players negotiate over a customized utility to determine reliable edits that improve both immediate answer support and long-term graph quality. Experiments on multi-hop question answering and conflict-aware benchmarks demonstrate that CONSIST consistently outperforms recent state-of-the-art baselines, achieving average performance gains up to 7.0% EM and 6.3% F1. Moreover, CONSIST also produces substantially more compact evidence graphs, reducing retained nodes by 42.9% and edges by 54.7% on average.

1 Introduction

The advent of Large Language Models (LLMs) has significantly improved reasoning capabilities, yet they remain prone to factual errors when relying solely on parametric knowledge. Retrieval-Augmented Generation (RAG) mitigates this issue by incorporating external evidence into the reasoning process [17]. However, traditional RAG typically treats retrieved passages or documents as an unordered set, making it difficult to capture the relational structure required for coherent reasoning [9, 15, 44, 36]. Graph-enhanced RAG (GraphRAG) is thus introduced to organize retrieved information into **an evidence graph**, where nodes represent units such as entities or documents and edges encode semantic or relational connections [5, 53]. This structured representation enables the model to aggregate and traverse connected evidence, thereby supporting more coherent and context-aware reasoning compared to isolated text fragments.

As shown in Fig. 1, existing GraphRAG methods mainly differ in how the evidence graph is constructed and leveraged during reasoning. *Offline GraphRAG* builds a corpus-level graph in advance and retrieves query-relevant subgraphs at inference time [5–7]. To improve coverage, some methods further organize the graph via hierarchical clustering [32] or entity-centric semantic linking [54]. However, since the graph is fixed prior to reasoning, it may miss query-critical links that are not encoded during evidence retrieval and graph construction.

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

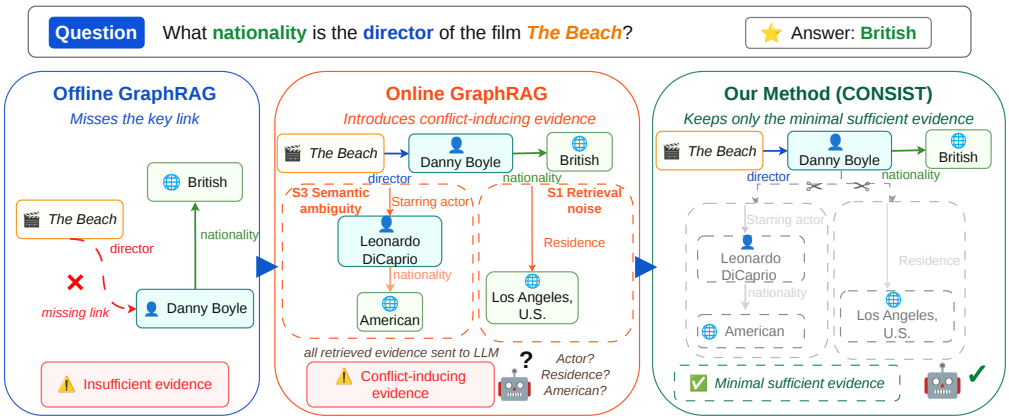


Figure 1: **GraphRAG with a minimal sufficient evidence graph.** Offline GraphRAG constructs a static evidence graph in advance and may fail when query-critical links are missing, while online GraphRAG could introduce conflict-inducing evidence that misleads query-aligned reasoning. The proposed CONSIST employs a Nash-guided graph editing to form a minimal sufficient evidence graph. For instance, given the query about *the nationality of the director of The Beach* in 2Wiki [9], CONSIST retains the essential reasoning path (*The Beach* → Danny Boyle → British) while pruning irrelevant links (such as Danny Boyle → Los Angeles, U.S.).

In contrast, *online GraphRAG* constructs the evidence graph dynamically for each query, typically by expanding paths from query-relevant seed nodes through iterative exploration and augmentation [11, 23, 25]. Despite this flexibility and adaptability, dynamic graph construction often introduces redundant evidence [34], counterfactual facts [21], or semantic ambiguity [16], resulting in misleading, query-biased reasoning and degraded answer quality. Prior work on conflict mitigation in RAG either leverages reliability-aware control (*e.g.*, context-aware decoding [34] or probability adjustment [2]) or explicitly equips the model with conflict reasoning capabilities [12, 50]. *However, existing approaches are designed for standard RAG settings and have not been extended to graph-structured evidence. In addition, they typically optimize for immediate outcomes without accounting for the impact of current decisions on future results, which may lead to locally optimal but globally suboptimal solutions.* This motivates a central question:

How can GraphRAG construct an evidence graph that is sufficient for answering a query while remaining compact and conflict-free?

This paper, accordingly, introduces a **Co**nflict-aware **Na**sh-optimized m**I**nimal **S**ufficien**T** graph learning (CONSIST) algorithm for GraphRAG reasoning. Unlike existing methods that rely solely on expansion, CONSIST adopts an **exploration–editing** paradigm, where the evidence graph is iteratively expanded and then selectively refined to remove conflicting or unnecessary structures. Specifically, CONSIST performs graph editing through Gumbel-based differentiable gates, characterized by a utility function that captures sufficiency, consistency, and minimality. The editing process is further framed as a Nash-guided optimization that balances immediate gains with long-term effects on subsequent graph evolution. Our contributions are summarized as follows:

- We propose CONSIST, *to the best of our knowledge*, the *first* framework for learning *minimal sufficient evidence graphs* for GraphRAG under query-conditioned knowledge conflict, retaining answer-supporting paths while pruning conflicting branches.
- We introduce a conflict-aware graph utility function that jointly models *sufficiency*, *consistency*, and *minimality*, providing a unified objective for regulating evidence construction.
- We formulate graph editing as a Nash-style bargaining process, where current and future players negotiate over the customized utility to produce reliable and compact graphs.
- We conduct extensive experiments on multi-hop question answering and conflict-oriented benchmarks, demonstrating that CONSIST consistently outperforms recent state-of-the-art methods while producing substantially more compact evidence graphs.

2 Preliminaries

Definition 1 GraphRAG. Instead of reasoning over an unordered set of text fragments (as in standard RAG), GraphRAG operates on structured evidence composed of nodes (*e.g.*, entities or documents) and edges (*e.g.*, semantic or relational connections). Specifically, in the *offline* setting, a corpus-level graph G is constructed in advance and remains fixed during retrieval. Given the query q, reasoning therefore marginalizes over subgraphs selected from G:

$$P(a \mid q, G) = \sum_{G_q \subseteq G} P(a \mid q, G_q)\, P(G_q \mid q, G). \tag{1}$$

In the *online* setting, the evidence graph is not fixed beforehand. Instead, it is incrementally constructed and refined for the target query, so reasoning marginalizes over query-specific graphs from a dynamic feasible space:

$$P(a \mid q) = \sum_{G_q \in \mathcal{G}(q)} P(a \mid q, G_q)\, P(G_q \mid q), \tag{2}$$

where $\mathcal{G}(q)$ defines a query-dependent graph space that may include structures (such as new edges) not present in the pre-constructed G.

Definition 2 Query-Conditioned Knowledge Conflict. This issue refers to errors where graph expansion deviates from the query-aligned reasoning path, introducing misleading or factually inconsistent intermediate evidence. We attribute such conflicts to three primary sources: *(S1) Retrieval noise*, which introduces irrelevant, redundant or incorrectly grounded evidence during retrieval or linking, leading to misleading graph connections [34, 51]; *(S2) Counterfactual fact*, which arises from intentionally-injected evidence that is factually inconsistent with common or prior knowledge, steering expansion toward different reasoning paths [21]; and *(S3) Semantic ambiguity*, where entities or relations admit multiple plausible interpretations, causing the model to follow query-misaligned reasoning paths [16]. As shown in Fig. 2, we randomly sample 1,000 incorrect responses (using LLaMA3-8B as the backbone) from ConFiQA-MC [1] and categorize them by *S1–S3* and the rest as *Unknown*. For response types, we follow [1] and focus on *Stubborn* (answers driven by parametric knowledge, ignoring context) and *Other* (uncertain or partially grounded responses). The analysis shows that *S1–S3* account for 22.3%, 54.6%, and 18.2% of errors, respectively, indicating that query-conditioned knowledge conflict is both prevalent and multi-faceted in GraphRAG. These findings motivate our approach, which explicitly addresses such conflicts through unified graph construction and refinement[1].

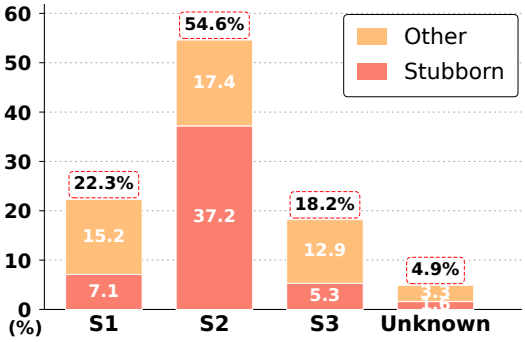


Figure 2: Prevalence and diversity of query-conditioned knowledge conflict in GraphRAG, based on 1,000 sampled error responses from ConFiQA-MC [1].

Definition 3 Nash Bargaining. Nash bargaining aims to resolve conflict among multiple (say K) players through maximizing joint gains relative to a disagreement point [27, 28]. Specifically, each player i is associated with a utility function $u_i(\cdot)$ that measures its preference over candidate solutions. These utilities jointly define a feasible set $\mathcal{U} = \{(u_1(\cdot), \ldots, u_K(\cdot))\}$. The Nash bargaining solution can then be calculated as

$$u^* = \arg\max_{u_i \in \mathcal{U}} \sum_{i}^{K} \log(u_i - d_i), \quad \text{s.t. } \forall i:\ u_i > d_i, \tag{3}$$

where d_i denotes the disagreement utility for player i. This formulation yields a Pareto-efficient solution that enforces balanced improvement across all K players, rather than favoring the utility of any single player. Bargaining-based formulations have been successfully applied in many downstream tasks [29, 30, 33, 41, 48], demonstrating strong empirical performance in balancing competing objectives.

[1] Details of the conflict analysis are provided in Appendix C.

3 Method

This section introduces CONSIST, a **CO**nflict-aware **Na**S**h**-optimized M**I**nimal **S**ufficien**T** graph learning framework, as illustrated in Fig. 3. In standard GraphRAG pipelines, the LLM handles query understanding and answer generation over graph-structured evidence. We instead focus on graph construction and evidence control, and thus *keep the backbone LLM frozen*.

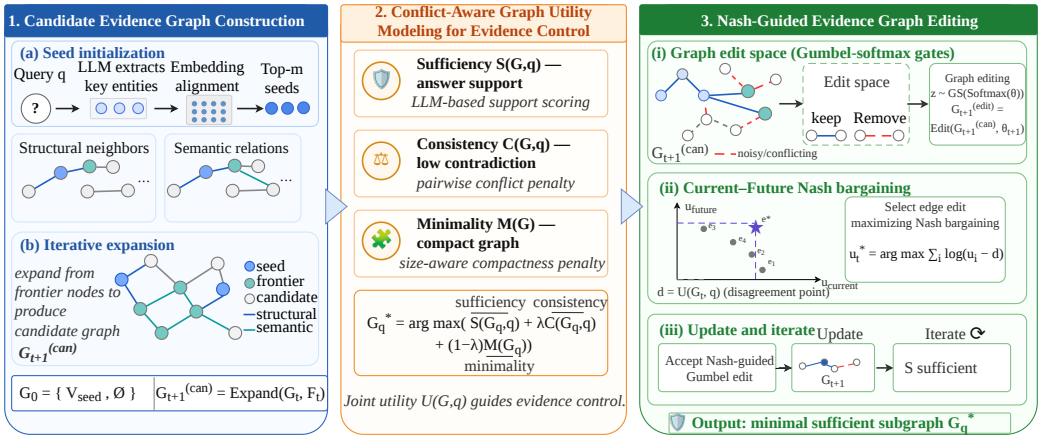


Figure 3: Overview of CONSIST. Given a query q, CONSIST constructs a candidate evidence graph via seed initialization and query-aware expansion, and then applies Gumbel-based edge editing with Nash-guided optimization to produce a compact minimal sufficient graph.

3.1 Problem Formulation

For a query q and a set of document nodes $\mathcal{V}$ with an initial edge set $\mathcal{E}$ (which can be an empty set or derived from prior knowledge such as metadata links or co-occurrence statistics), let $\mathcal{G}(q)$ denote the set of query-specific graph space reachable through edge expansion or removal over $\mathcal{V}$, where edges may originate from $\mathcal{E}$ or be dynamically introduced beyond $\mathcal{E}$. In the context of GraphRAG, *conflict* arises when newly introduced evidence is semantically inconsistent with existing evidence or introduces redundant structure in the evolving graph. Accordingly, our objective of CONSIST is to learn a minimal sufficient evidence graph $G_q \in \mathcal{G}(q)$, which should satisfy the following three criteria: (i) *sufficiency*, where G_q provides evidence that supports answering q; (ii) *consistency*, where G_q avoids contradictory or incompatible knowledge across its nodes and edges; and (iii) *minimality*, where G_q remains compact, minimizing duplicated or unnecessary evidence. Formally, the graph learning is formulated as

$$G_q^* = \arg\max_{G_q \in \mathcal{G}(q)} \underbrace{\mathcal{S}(G_q, q)}_{\text{sufficiency}} + \lambda \underbrace{\mathcal{C}(G_q, q)}_{\text{consistency}} + (1-\lambda) \underbrace{\mathcal{M}(G_q)}_{\text{minimality}}, \tag{4}$$

where λ is a trade-off weight between consistency and minimality.

3.2 Candidate Evidence Graph Construction

Seed initialization. CONSIST first prompts an LLM with the query q to extract a small set of query-relevant entities (*i.e.*, key entities implied or mentioned in q). These entities are then aligned with $\mathcal{V}$ via embedding similarity [19], forming a candidate node set $\mathcal{V}_{\text{cand}}$. Each candidate node $v \in \mathcal{V}_{\text{cand}}$ is further scored using an LLM-based relevance function $\text{LLM}(q, v)$, and the top-m nodes are selected as the seed set $\mathcal{V}_{\text{seed}} = \text{TopK}_m\big(\{\text{LLM}(q, v)\}_{v \in \mathcal{V}_{\text{cand}}}\big)$[2]. Accordingly, the initial graph is initialized as $G_0 = \{\mathcal{V}_{\text{seed}}, \varnothing\}$, where no edges are present at this stage (*i.e.*, $E_0 = \varnothing$), and the initial frontier (nodes to be expanded next) is defined as $\mathcal{F}_0 = \mathcal{V}_{\text{seed}}$.

Candidate expansion. Given the candidate graph $G_t = (\mathcal{V}_t, \mathcal{E}_t)$ and frontier $\mathcal{F}_t$ at the t-th iteration, CONSIST considers to expand G_t from two complementary sources, *i.e.*, structural neighbors and

[2]The detailed pipeline, including entity extraction, embedding-based alignment, and LLM-based reranking prompts for seed initialization, is provided in Appendix G.

semantic relations[3]. Specifically, for each $v \in \mathcal{F}_t$: (i) *structure*: one-hop neighbors induced by existing edges as $\mathcal{N}_{\text{str}}(v) = \{v_{str} \in \mathcal{V} \mid (v, v_{str}) \in \mathcal{E}\}$, and/or (ii) *semantic*: latent relations instantiated from document-level statistics (*e.g.*, embedding similarity or entity co-occurrence) as $\mathcal{N}_{\text{sem}}(v) = \{v_{sem} \in \mathcal{V} \mid s(v, v_{sem}) \geq \tau_{sem}\}$, where $s(v, v_{sem})$ denotes semantic relatedness and τ_{sem} is a threshold[4]. Consequently, the candidate node and edge sets at the $(t+1)$-th iteration are updated as

$$\begin{cases} \mathcal{V}_{t+1} = \mathcal{V}_t \cup \bigcup_{v \in \mathcal{F}_t} \left(\mathcal{N}_{\text{str}}(v) \cup \mathcal{N}_{\text{sem}}(v)\right), \\ \mathcal{E}_{t+1} = \mathcal{E}_t \cup \{(v, v_{str,sem}) \mid v \in \mathcal{F}_t,\ v_{str,sem} \in \mathcal{N}_{\text{str}}(v) \cup \mathcal{N}_{\text{sem}}(v)\}. \end{cases} \tag{5}$$

leading to the expanded candidate graph $G_{t+1}^{(can)} = (\mathcal{V}_{t+1}, \mathcal{E}_{t+1})$.

3.3 Conflict-Aware Graph Utility

In existing GraphRAG methods, the evidence graph is typically expanded iteratively to cover more potential reasoning paths [11, 14, 23, 25]. While such expansion improves coverage, it may also introduce *conflict*, such as semantically inconsistent evidence, redundant or overlapping information, and structurally distracting branches. To address this, `CONSIST` models three criteria (shown in Eq. (4)), *i.e.*, *sufficiency*, *consistency*, and *minimality*, and instantiates them as a unified graph-level utility function.

Sufficiency. We measure how well a given graph G supports answering the query q via an answer-support score:

$$\mathcal{S}(G, q) = \text{LLM}_{\text{sup}}(G, q), \tag{6}$$

where LLM_{sup} is an LLM-based scorer.

Consistency. To capture semantic conflicts, we compute pairwise consistency among edges:

$$\mathcal{C}(G, q) = 1 - \text{Avg}_{(e_i, e_j) \in \mathcal{E}(G)} \text{LLM}_{\text{pair}}(q, (e_i, e_j)), \tag{7}$$

where LLM_{pair} is an LLM-based function that scores the contradiction degree. A larger value indicates stronger conflict. Both scorers use the same frozen backbone LLM, given only the query and retrieved graph, without ground-truth answers.

Minimality. We encourage compact graphs via

$$\mathcal{M}(G) = |\mathcal{E}(G')|/|\mathcal{E}(G)|, \tag{8}$$

where G' denotes the graph before expansion to G, and $|\mathcal{E}(\cdot)|$ denotes the number of edges. This term measures the relative expansion ratio with respect to the previous graph G', acting as a proxy for compactness.

Overall Utility. The overall graph utility is defined by jointly optimizing the three criteria[5].

$$\begin{aligned} U(G, q) = \text{LLM}_{\text{sup}}(G, q) + \lambda\Big(1 - \text{Avg}_{(e_i, e_j) \in \mathcal{E}(G)} \text{LLM}_{\text{pair}}(q, (e_i, e_j))\Big) \\ + (1 - \lambda)\Big(|\mathcal{E}(G')|/|\mathcal{E}(G)|\Big). \end{aligned} \tag{9}$$

3.4 Nash-Guided Graph Editing

Given the expanded candidate graph $G_{t+1}^{(\text{can})} = (\mathcal{V}_{t+1}, \mathcal{E}_{t+1})$, the remaining question is how to perform reliable graph editing over $G_{t+1}^{(\text{can})}$. Specifically, rather than directly accepting all candidate nodes and edges, `CONSIST` is proposed to refine $G_{t+1}^{(\text{can})}$ through a Nash-style bargaining process, where current and future players evaluate candidate graph edits using the proposed utility.

Stochastic Graph Editing. To begin with, we formulate graph editing as a differentiable selection process using a Gumbel-based operator. Specifically, let $\boldsymbol{z} \in \{0, 1\}^{|\mathcal{E}_{t+1}|}$ denote a binary indicator

[3]When no prior edges are available (*i.e.*, $\mathcal{E} = \varnothing$), expansion relies solely on semantic relations.

[4]Details of structural and semantic expansion are provided in Appendix H.

[5]Detailed prompts for LLM_{sup} and LLM_{pair} are provided in Appendix I.1 and I.2.

vector over $|\mathcal{E}_{t+1}|$ candidate edges, where $z_j = 1$ indicates that the j-th edge is retained. To enable gradient-based optimization, $\boldsymbol{z}$ is sampled from a Gumbel-Softmax distribution parameterized by $\boldsymbol{\theta}$:

$$\boldsymbol{z} \sim \mathrm{GS}(\mathrm{Softmax}(\boldsymbol{\theta})),$$

where $\mathrm{GS}(\cdot)$ denotes the Gumbel-Softmax distribution [13, 24], and its approximation is given by:

$$\mathrm{GS}(\boldsymbol{\theta}) = \sigma\left(\frac{\log\epsilon - \log(1-\epsilon) + \boldsymbol{\theta}}{\Delta}\right), \tag{10}$$

where $\epsilon \sim \mathcal{U}(0,1)$ is uniform noise, Δ is the temperature parameter, and $\sigma(\cdot)$ is the sigmoid function. Furthermore, the edited graph is defined as

$$G_{t+1}^{(\mathrm{edit})} = \mathrm{Edit}\Big(G_{t+1}^{(\mathrm{can})}, \boldsymbol{\theta}_{t+1}\Big), \tag{11}$$

Notably, we perform *edge-level* editing, as nodes are implicitly determined by their incident edges and are removed if all connected edges are pruned. This edge-centric formulation enables finer-grained control over the graph structure, allowing CONSIST to selectively preserve informative relations while pruning conflicting or redundant connections. In addition, to maintain stability across iterations, we initialize the current selection $\boldsymbol{\theta}_{t+1}$ using the previous step, and assign newly introduced edges a default value of 1, providing a temporary retention preference before utility-based editing. Formally,

$$\theta_{t+1}^{e} = \begin{cases} \theta_t^e, & e \in \mathcal{E}_t, \\ 1, & otherwise, \end{cases} \tag{12}$$

where θ_t^e denotes the selection parameter for edge e at iteration t.

Nash-guided Updating. An individual edit may improve the current graph, yet introduce conflicts that emerge only after further expansion. This creates a trade-off in graph editing: a locally favourable edit can increase immediate evidence support, but may also open unreliable reasoning paths in later steps. Therefore, graph updating should not be driven solely by current benefit; it must also account for whether an edit remains stable under future expansion. Accordingly, CONSIST proposes to evaluate each edited graph using two players, *current* and *future*, each associated with utility functions u_{cur} and u_{fut}, respectively. The former favours edits that immediately improve answer support, while the latter favours edits that remain beneficial after subsequent graph expansion. As a result, edits that yield only short-term gains but later introduce query-irrelevant evidence receive low bargaining value. In contrast, edits that jointly improve answer support and reduce conflict are consistently preferred, guiding CONSIST toward learning a compact and sufficient graph. Formally, we introduce the following utilities to achieve this and the justification is provided in Appendix K.

$$u_{\mathrm{cur}}\big(G_{t+1}^{(\mathrm{edit})}; \boldsymbol{\theta}\big) = \mathbb{E}[U\big(G_{t+1}^{(\mathrm{edit})}(\boldsymbol{\theta}), q\big)],\ u_{\mathrm{fut}}\big(G_{t+1+H}^{(\mathrm{edit})}; \boldsymbol{\theta}\big) = \mathbb{E}[U\big(G_{t+1+H}^{(\mathrm{edit})}(\boldsymbol{\theta}), q\big)], \tag{13}$$

where $G_{t+1+H}^{(\mathrm{edit})}(\boldsymbol{\theta})$ is obtained by rolling out from $G_{t+1}^{(\mathrm{edit})}(\boldsymbol{\theta})$ for another H expansion steps without applying further editing. We estimate the expectation $\mathbb{E}(\cdot)$ via Monte Carlo sampling, a standard approximation for stochastic neural modules. This unedited rollout serves as a *look-ahead* stress test that evaluates whether the current edit remains stable when subsequent expansion exposes it to future (additional) evidence. Accordingly, following the Nash bargaining principle [27, 28], CONSIST selects edits by maximizing the following Nash-style product:

$$u_{t+1}^{*} = \arg\max_{u_i \in \mathcal{U}_{t+1}(\boldsymbol{\theta})} \sum_{i \in \{\mathrm{cur}, \mathrm{fut}\}} \log(u_i - d_t) \quad \text{s.t. } \forall i,\ u_i > d_t, \tag{14}$$

where $\mathcal{U}_{t+1}(\boldsymbol{\theta})$ is the utilities on an agreement set $\mathcal{A}$ and $\boldsymbol{\theta} \in \mathcal{A}$, and d_t is the disagreement point. In CONSIST, we set $d_t = \mathbb{E}[U(G_t, q)]$; thus, a candidate edit is considered feasible only when it improves both u_{cur} and u_{fut} over the unchanged graph. If no such edit exists, CONSIST keeps the current graph unchanged, yielding a conservative update rule that avoids unnecessary graph modifications. In contrast, setting $d_t = 0$ may allow candidate edits that do not improve both utilities over the current graph G_t. Notably, the induced utility set may not fully satisfies all convexity assumptions of the classical Nash Bargaining Problem or that the solution is unique. Rather, Eq. (14) uses the Nash product as a Nash-style coordination criterion: it ranks feasible relaxed edits by their joint improvement over the disagreement point and penalizes one-sided gains. As a result, this allows CONSIST to select an edit u_{t+1}^{*} that balances the immediate quality of G_{t+1} with the stability of its look-ahead rollout G_{t+1+H}. The updating process terminates when the answer-support score exceeds a threshold, $\mathcal{S}(G, q) \geq \tau_{\mathrm{stop}}$. The resulting graph G_q^{*} is then passed to produce the final answer as $\hat{a} = \mathrm{LLM}(G_q^{*}, q)$.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate CONSIST on both multi-hop question answering and conflict-oriented benchmarks. For multi-hop QA, we use 2WIKI [9], HOTPOTQA [44], and MUSIQUE [36]. For conflict-oriented QA, we use FAITHEVAL [26] and CONFIQA [1] (including all MC, MR, and QA splits).

Baselines. We compare CONSIST with four groups of baselines, from *parametric-only models* to *existing offline/online GraphRAG* and *conflict-aware reasoning*. First, we include LLM-only methods without retrieval, namely GPT-4o, DeepSeek-V3, and Llama3-8B. Second, we consider offline GraphRAG methods, including RAPTOR [32], G-RETRIEVER [8], LIGHTRAG [6], GRAPHRAG [5], and HIPPORAG [7]. Third, we compare with online GraphRAG methods that iteratively construct evidence graphs, including TOG [23], SIRERAG [49], HOPRAG [20], TCR-QF [10], and RELINK [11]. Finally, we include conflict-aware RAG methods, including OPIN(INSTR) [52], KRE [46], FAITHFULRAG [50], CAD [34], COIECD [47], and ADACAD [37]. More details about related work can be found in Appendix B.

Implementation details. To make a fair comparison, following prior work [7, 20, 49], we use 1,000 samples from each dataset for evaluation; detailed statistics are provided in Appendix D. We set m=2 (number of seeds) for multi-hop QA and m=3 for conflict-oriented QA, with semantic expansion threshold τ_{sem}=0.6/0.7, respectively. We further set $b = 6$ (expansion budget), H=2 (rollout step), utility weight λ=0.6, Δ=0.6 (Gumbel-Softmax temperature), and $\tau_{\text{stop}} = 0.7$ (termination criterion). For multi-hop QA benchmarks, we follow [20, 23] to construct corpus-level graphs. For conflict-oriented datasets, counterfactual evidence is formatted as additional knowledge and injected into the original corpus. The CONSIST is optimized using AdamW with a learning rate of 2×10^{-5}. *Baseline results are directly taken from the original papers*, while our results are averaged over 10 runs. The model performance is measured using EM and F1 from the ground-truth answers.

4.2 Main Results

Table 1 presents the performance comparison across three multi-hop QA benchmarks and two backbone LLMs. CONSIST consistently outperforms all baselines across all cases. For instance, it improves over the strongest baseline RELINK by $+9.2\%$ EM and $+9.3\%$ F1 on average.

Table 1: Performance comparison of various methods across multi-hop QA benchmarks, using DEEPSEEK-V3 and GPT-4O as the backbone LLM. The **best** and second-best results are highlighted.

Method	2Wiki		HotpotQA		MuSiQue		Avg.	
	EM(↑)	F1(↑)	EM(↑)	F1(↑)	EM(↑)	F1(↑)	EM(↑)	F1(↑)
DEEPSEEK-V3	31.2	36.5	29.4	40.2	7.8	16.4	22.8	31.0
+ RAPTOR [32]	46.7	54.8	47.2	63.1	18.8	28.6	37.6	48.8
+ TOG [23]	32.8	37.3	24.1	29.2	5.3	7.0	20.7	24.5
+ G-RETRIEVER [8]	36.2	45.5	36.9	49.3	9.2	16.9	27.4	37.2
+ LIGHTRAG [6]	30.0	36.4	38.4	49.8	10.4	17.1	26.3	34.4
+ GRAPHRAG [5]	31.8	37.9	45.0	56.9	13.8	20.6	30.2	38.5
+ HIPPORAG [7]	57.8	68.4	49.8	64.7	19.0	29.8	42.2	54.3
+ RELINK [11]	62.8	72.2	55.8	70.4	25.2	37.0	47.9	59.9
+ CONSIST	**65.3**	**73.9**	**62.6**	**77.8**	**43.5**	**55.9**	**57.1**	**69.2**
Std.	±1.0	±0.9	±1.3	±1.2	±0.7	±1.1	±1.0	±1.1
GPT-4O	29.2	35.8	33.0	42.4	10.6	21.4	24.3	33.2
+ SIRERAG [49]	59.6	67.9	61.7	76.5	40.5	53.1	53.9	65.8
+ TCR-QF [10]	59.8	68.0	55.8	70.8	30.3	43.2	48.6	60.7
+ HOPRAG [20]	61.1	68.3	62.0	76.1	42.2	54.9	55.1	66.4
+ CONSIST	**67.2**	**75.7**	**64.3**	**79.0**	**44.1**	**56.8**	**58.5**	**70.5**
Std.	±1.1	±0.8	±0.9	±0.8	±0.5	±1.0	±0.8	±0.9

Table 2: Performance comparison of various methods across conflict-oriented faithfulness benchmarks, using LLAMA-3-8B as the backbone LLM. The **best** and second-best results are highlighted.

Method	FaithEval		ConFiQA (MC)		ConFiQA (MR)		ConFiQA (QA)		Avg.	
	EM(↑)	F1(↑)	EM(↑)	F1(↑)	EM(↑)	F1(↑)	EM(↑)	F1(↑)	EM(↑)	F1(↑)
LLAMA-3-8B	6.0	27.7	2.1	5.0	1.9	6.1	1.3	6.1	2.8	11.2
+ OPIN(INSTR) [52]	15.1	34.9	57.3	67.4	54.0	65.9	67.4	76.9	48.5	61.3
+ CAD [34]	42.7	59.4	11.4	16.0	31.3	40.0	38.1	48.3	30.9	40.9
+ COIECD [47]	41.3	56.1	24.0	28.5	43.3	50.9	60.1	67.1	42.2	50.7
+ ADACAD [37]	47.7	61.4	32.5	43.3	42.5	46.7	48.1	55.7	42.7	51.8
+ KRE [46]	12.1	59.1	59.8	68.2	58.9	68.7	74.7	84.0	51.4	70.0
+ FAITHFULRAG [50]	49.1	62.4	61.2	70.3	53.1	62.4	65.2	70.4	57.2	66.4
+ CONSIST	**55.7**	**68.5**	**65.1**	**73.9**	**63.8**	**72.7**	**77.6**	**87.3**	**65.6**	**75.6**
Std.	±1.2	±1.5	±0.7	±0.8	±0.6	±0.6	±0.3	±0.2	±0.7	±0.8

In addition, Table 2 further evaluates performance on conflict-oriented benchmarks. CONSIST again achieves the best results on all datasets, reaching 65.6% EM and 75.6% F1 on average with low performance variance. This advantage stems from explicit evidence graph regulation in CONSIST: in multi-hop QA, it preserves necessary intermediate links, while in conflict-heavy settings, it suppresses inconsistent or ambiguous paths that could mislead reasoning. Consequently, CONSIST maintains more accurate reasoning trajectories, leading to more reliable predictions across both multi-step and conflict-rich scenarios.

4.3 Ablation Study

To analyze the effectiveness of CONSIST, we conduct ablation studies on 2WIKI and CONFIQA (MC). The former contains graphs with the largest average edge number, while the latter includes multiple conflicting facts, making both highly challenging scenarios.

Table 3: Impact analysis from individual components for CONSIST, using 2Wiki (GPT-4O) and ConFiQA (LLAMA-3).

	2Wiki		ConFiQA	
	EM(↑)	F1(↑)	EM(↑)	F1(↑)
Backbone	29.2	35.8	2.1	5.0
+ EXP	56.2	65.1	49.4	55.6
+ GS-LLM	59.4	68.5	54.9	60.4
+ GS-U	63.9	72.8	61.7	67.3
+ GS-SUM	64.6	73.4	63.2	70.1
+ Full	**67.2**	**75.7**	**65.1**	**73.9**

Model breakdown. The proposed CONSIST controls graph expansion through Gumbel-based edge editing, where the proposed utility function is jointly leveraged by current and future players. To assess the contribution of each component, we consider the following variants: (i) EXP: expansion-only (no subsequent editing); (ii) GS-LLM: Gumbel-based editing with an LLM-derived heuristic (without the proposed utility); (iii) GS-U: Gumbel-based editing with the proposed utility (Eq. (9)), considering only the current-player utility (*i.e.*, without future-player guidance); and (iv) GS-SUM: Gumbel-based editing with the sum of current and future utilities, replacing Nash-guided coordination. As observed from Table 3, all components positively contribute to the overall performance, with consistent gains when each component is introduced. For instance, adding Gumbel-based editing (GS-LLM) over the expansion-only baseline (EXP) yields an averaged improvement of 4.1 F1 points, indicating the importance of explicit graph edge refinement. We further observe that incorporating the utility-guided objective (GS-U) brings another 5.6 averaged F1 gain, showing that the proposed utility provides a better supervision for identifying useful and consistent edges. In addition, GS-SUM further improves performance by jointly considering current and future utilities, outperforming GS-U by 0.6 and 2.8 F1 points on 2Wiki and ConFiQA-MC, respectively. This highlights the importance of future utility, as optimizing only for current utility may preserve locally useful but subsequently distracting edges. Finally, incorporating Nash bargaining with our proposed disagreement point (FULL) achieves the best performance, highlighting the importance of requiring improvements over the unchanged graph rather than merely positive utilities.

Utility ablation. This subsection investigates the effect of the utility function. Specifically, we consider the following variants to implement Consistency ($\mathcal{C}$) and Minimality ($\mathcal{M}$), while keeping other components (*e.g.*, graph expansion and Gumbel-based editing) unchanged: (i) $\mathcal{C}_{\text{NLI}}$: replaces

the LLM-based consistency with a natural language inference-based contradiction detector (fine-tuned with ConFiQA-MC rest instances) [3]; and (ii) $\mathcal{M}_{\text{emb}}$: replaces the size-based minimality with an embedding-based redundancy to count nodes if their cosine similarity is greater than 0.75.

Fig. 4 illustrates the effect of different utility implementations. First, all utility variants consistently outperform existing baselines (see Tables 1 and 2), demonstrating the general effectiveness of the proposed method. We further observe that replacing $\mathcal{C}$ with $\mathcal{C}_{\text{NLI}}$ even achieve better EM/F1 performance on ConFiQA. This may be attributed to the fact that ConFiQA contains counterfactual and conflicting evidence, where contradiction modeling becomes particularly beneficial. Overall, the results show that the effectiveness of the proposed framework is not tied to a specific utility implementation, but rather stems from its ability to regulate evidence graphs under query-conditioned conflicts.

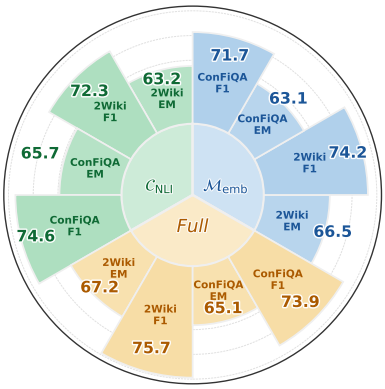
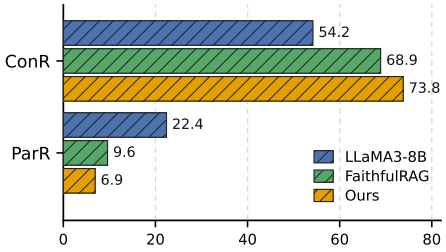
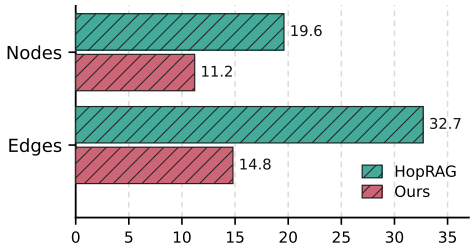


Figure 4: Ablation of the utility function with alternative implementations of Consistency ($\mathcal{C}$) and Minimality ($\mathcal{M}$).

Graph consistency and minimality. The quality of the learned evidence graphs (consistency) is first evaluated in terms of conflict resolution, by examining whether they effectively handle conflicts between retrieved context and parametric knowledge. Following [2], we adopt two metrics of *ConR* (alignment with retrieved context) and *ParR* (alignment with parametric knowledge). A higher ConR together with a lower ParR indicates better conflict resolution, where the model favors grounded context over potentially conflicting parametric knowledge. Fig. 5(a) reports the comparison of ConR and ParR across methods for ConFiQA, showing that CONSIST achieves higher ConR while reducing ParR. This demonstrates that the proposed method effectively suppresses conflicting parametric knowledge and promotes context-faithful reasoning through improved graph construction. In addition, we also evaluate the size of obtained graphs for minimality. Fig. 5(b) reports the average number of retained nodes and edges per question on 2Wiki. CONSIST retains only 11.2 nodes and 14.8 edges on average, which is substantially smaller than recent SOTA method of HopRAG (19.6 nodes, 32.7 edges). Overall, these results indicate that the proposed CONSIST constructs conflict-consistent graphs while maintaining a compact and focused structure.



(a) Conflict resolution, measuring the model tendency to favor parametric knowledge over conflicting context.



(b) Retained graph size (per question).

Figure 5: Graph consistency and minimality analysis of CONSIST.

Editing design. Table 4 investigates new-edge initialization (Eq. (12)) and disagreement-point (Eq. (14)) settings while keeping others unchanged. Initialization at 0.5 consistently underperforms our default 1.0, which allows new edges to fully participate in current and future utility evaluation. A lower initial weight may underestimate their contributions and prematurely remove potentially useful edges. On the other hand, with $d_t = 0$, positive utilities may yield edits worse than the unchanged graph, whereas our default $d_t = \mathbb{E}[U(G_t, q)]$ requires im-

Table 4: Impact analysis of new-edge initialization and disagreement-point settings.

	2Wiki		ConFiQA	
	EM(↑)	F1(↑)	EM(↑)	F1(↑)
Initialization: 0.5	66.3	74.9	63.7	72.8
$d_t = 0$	65.4	73.8	62.5	71.6
Ours	**67.2**	**75.7**	**65.1**	**73.9**

provement over the current graph. These results validate both design choices in enabling effective and conservative graph refinement.

Utility weights. In our overall utility (Eq. (9)), we fix the weight of Sufficiency at 1, while the weights of Consistency and Minimality sum to 1, treating Sufficiency as the primary reference term. To investigate the effect of different weight allocations, we accordingly assign weights α, λ, and $1-\alpha-\lambda$ to Sufficiency, Consistency, and Minimality, respectively, while keeping other settings unchanged. The relevant results are reported in Table 5. Note that, the setting $(\alpha, \lambda) = (0.5, 0.3)$ preserves the same relative weighting as our main configuration (scaled by a factor of two). The results show that assigning small weight to Sufficiency substantially reduces reasoning accuracy, while a large weight weakens the regulation from Consistency and Minimality, resulting in performance similar to methods without conflict regulation. These results support using Sufficiency as the primary term while balancing Consistency and Minimality for effective conflict-aware editing.

Table 5: Sensitivity to the Sufficiency/Consistency/Minimality coefficients.

α/λ	2Wiki		ConFiQA	
	EM(↑)	F1(↑)	EM(↑)	F1(↑)
0.3/0.3	60.9	67.8	61.6	69.7
0.5/0.3	**67.2**	**75.7**	**65.1**	**73.9**
0.7/0.2	62.4	71.5	63.8	72.1

Backbone and Prompt robustness. To assess the sensitivity of CONSIST to the backbone LLM and scoring prompt, Table 6 reports results using two backbone LLMs under three prompt conditions. Specifically, (i) QUERY + GRAPH uses only the query and retrieved graph; (ii) SCORING RULE adds additional judging criteria; and (iii) FULL employs the complete prompts as Appendix I. Across both benchmarks, Qwen2-7B and Llama2-7B exhibit a similar trend: enriching the scoring prompt yields marginal gains, with performance varying by approximately 1.21% in relative terms. These results indicate that the proposed method exhibits consistent behavior across different backbone LLMs and prompt variants.

Table 6: Robustness of CONSIST across different backbone LLMs and prompt variants.

Backbone	Prompt Variant	2Wiki		ConFiQA	
		EM(↑)	F1(↑)	EM(↑)	F1(↑)
Qwen2-7B	Query + Graph	59.4	69.2	61.0	70.7
	Scoring Rule	59.5	69.4	61.2	70.7
	Full	**59.7**	**69.4**	**61.5**	**70.8**
Llama2-7B	Query + Graph	56.6	66.0	58.2	67.9
	Scoring Rule	56.8	**66.3**	58.7	68.0
	Full	**56.9**	66.2	**58.9**	**68.1**

Other Analyses. We further assess model robustness by randomly pruning edges from the final evidence graph, showing that CONSIST retains more task-critical and less redundant edges (Appendix F). CONSIST also achieves comparable time complexity compared to existing methods (Appendix J). At last, ablation studies on all six hyperparameters are provided in Appendix E.

5 Conclusion

This paper presented CONSIST, a conflict-aware Nash-optimized framework for learning minimal sufficient evidence graphs in GraphRAG. Instead of accepting all retrieved or expanded evidence for question answering, CONSIST explicitly regulates graph construction by jointly modeling sufficiency, consistency, and minimality, and further refines candidate graphs through Gumbel-based edge editing guided by current–future Nash bargaining. Experiments on multi-hop question answering and conflict-oriented benchmarks show that CONSIST consistently improves answer accuracy while producing substantially more compact evidence graphs.

Acknowledgments

The work of Jie Yang is supported by Australian Research Council Discovery Project (DP210101426). The work of Xinguo Yu and Hao Wu are supported by General Project of the National Natural Science Foundation of China (Grant No: 62277022). The work of Hao Wu is also supported by scholarship from the China Scholarship Council.

References

[1] Baolong Bi, Shaohan Huang, Yiwei Wang, Tianchi Yang, Zihan Zhang, Haizhen Huang, Lingrui Mei, Junfeng Fang, Zehao Li, Furu Wei, Weiwei Deng, Feng Sun, Qi Zhang, and Shenghua Liu. Context-DPO: Aligning language models for context-faithfulness. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 10280–10300, 2025. URL https://aclanthology.org/2025.findings-acl.536/.

[2] Baolong Bi, Shenghua Liu, Yiwei Wang, Yilong Xu, Junfeng Fang, Lingrui Mei, and Xueqi Cheng. Parameters vs. context: Fine-grained control of knowledge reliance in language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=TJ3DqFiGau.

[3] Marc E. Canby, Xinchi Chen, Xing Niu, Jifan Chen, Bonan Min, Sergul Aydore, and Vittorio Castelli. Benchmarking query-conditioned natural language inference. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 14808–14835, 2025. URL https://aclanthology.org/2025.findings-acl.765/.

[4] Eunseong Choi, June Park, Hyeri Lee, and Jongwuk Lee. Conflict-aware soft prompting for retrieval-augmented generation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 26981–26995, 2025. URL https://aclanthology.org/2025.emnlp-main.1371/.

[5] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024. URL https://doi.org/10.48550/arXiv.2404.16130.

[6] Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. LightRAG: Simple and fast retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 10746–10761, 2025. URL https://aclanthology.org/2025.findings-emnlp.568/.

[7] Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 59532–59569, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/6ddc001d07ca4f319af96a3024f6dbd1-Paper-Conference.pdf.

[8] Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V. Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. In *Advances in Neural Information Processing Systems*, volume 37, pages 132876–132907, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/efaf1c9726648c8ba363a5c927440529-Paper-Conference.pdf.

[9] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, 2020. URL https://aclanthology.org/2020.coling-main.580/.

[10] Manzong Huang, Chenyang Bu, Yi He, and Xindong Wu. How to mitigate information loss in knowledge graphs for graphrag: leveraging triple context restoration and query-driven feedback. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, IJCAI '25, 2025. URL https://doi.org/10.24963/ijcai.2025/901.

[11] Manzong Huang, Chenyang Bu, Yi He, Xingrui Zhuo, and Xindong Wu. Relink: Constructing query-driven evidence graph on-the-fly for graphrag. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(37):31202–31210, 2026. URL https://doi.org/10.1609/aaai.v40i37.40382.

[12] Nan Huo, Jinyang Li, Bowen Qin, Ge Qu, Xiaolong Li, Xiaodong Li, Chenhao Ma, and Reynold Cheng. Micro-act: Mitigate knowledge conflict in question answering via actionable self-reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18550–18574, 2025. URL https://aclanthology.org/2025.acl-long.909/.

[13] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=rkE3y85ee.

[14] Pengcheng Jiang, Lang Cao, Ruike Zhu, Minhao Jiang, Yunyi Zhang, Jiaming Shen, Jimeng Sun, and Jiawei Han. RAS: Retrieval-and-structuring for knowledge-intensive LLM generation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=fqqmeg61yd.

[15] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 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, 2020. URL https://aclanthology.org/2020.emnlp-main.550/.

[16] Jungyeon Lee, Lee Kangmin, and Taeuk Kim. MAGIC: A multi-hop and graph-based benchmark for inter-context conflicts in retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 8783–8803, 2025. URL https://aclanthology.org/2025.findings-emnlp.466/.

[17] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

[18] Kun Li, Tianhua Zhang, Xixin Wu, Hongyin Luo, James Glass, and Helen Meng. Decoding on graphs: Faithful and sound reasoning on knowledge graphs through generation of well-formed chains. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24349–24364, 2025. URL https://aclanthology.org/2025.acl-long.1186/.

[19] Mufei Li, Siqi Miao, and Pan Li. Simple is effective: The roles of graphs and large language models in knowledge-graph-based retrieval-augmented generation. In *International Conference on Learning Representations*, volume 2025, pages 6061–6089, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/11e1900e680f5fe1893a8e27362dbe2c-Paper-Conference.pdf.

[20] Hao Liu, Zhengren Wang, Xi Chen, Zhiyu Li, Feiyu Xiong, Qinhan Yu, and Wentao Zhang. HopRAG: Multi-hop reasoning for logic-aware retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 1897–1913, 2025. URL https://aclanthology.org/2025.findings-acl.97/.

[21] Siyi Liu, Qiang Ning, Kishaloy Halder, Zheng Qi, Wei Xiao, Phu Mon Htut, Yi Zhang, Neha Anna John, Bonan Min, Yassine Benajiba, et al. Open domain question answering with conflicting contexts. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1838–1854, 2025. URL https://aclanthology.org/2025.findings-naacl.99/.

[22] Linhao Luo, Yuan-Fang Li, Reza Haffari, and Shirui Pan. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *International Conference on Learning Representations*, volume 2024, pages 14400–14423, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/3e2aeb66481dd63a32421bf032b70384-Paper-Conference.pdf.

[23] Shengjie Ma, Chengjin Xu, Xuhui Jiang, Muzhi Li, Huaren Qu, Cehao Yang, Jiaxin Mao, and Jian Guo. Think-on-graph 2.0: Deep and faithful large language model reasoning with knowledge-guided retrieval augmented generation. In *International Conference on Learning Representations*, volume 2025, pages 52782–52806, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/830b1abc6d2da85f23d41169fa44d185-Paper-Conference.pdf.

[24] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=S1jE5L5gl.

[25] Costas Mavromatis, Soji Adeshina, Vassilis N. Ioannidis, Zhen Han, Qi Zhu, Ian Robinson, Bryan Thompson, Huzefa Rangwala, and George Karypis. BYOKG-RAG: Multi-strategy graph retrieval for knowledge graph question answering. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 27881–27898, 2025. URL https://aclanthology.org/2025.emnlp-main.1417/.

[26] Yifei Ming, Senthil Purushwalkam, Shrey Pandit, Zixuan Ke, Xuan-Phi Nguyen, Caiming Xiong, and Shafiq Joty. Faitheval: Can your language model stay faithful to context, even if "the moon is made of marshmallows". In *International Conference on Learning Representations*, volume 2025, pages 29430–29456, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/48404cd9ce03946c6b7177691f3267a1-Paper-Conference.pdf.

[27] John Nash. The bargaining problem. *Econometrica*, 18(2):155–162, 1950.

[28] John Nash. Two-person cooperative games. *Econometrica*, 21(1):128–140, 1953.

[29] Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 16428–16446, 2022. URL https://proceedings.mlr.press/v162/navon22a.html.

[30] Dung Viet Nguyen, Anh Nguyen Thi, Minh Hoang Nguyen, Luc Nguyen, Shiqi Jiang, Ethan Fetaya, Linh Duy Tran, Gal Chechik, and Tan Minh Nguyen. Expert merging in sparse mixture of experts with nash bargaining. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=JLe9xfd0ln.

[31] Stephen Robertson, Hugo Zaragoza, and Michael Taylor. Simple bm25 extension to multiple weighted fields. In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, page 42–49, 2004. URL https://doi.org/10.1145/1031171.1031181.

[32] Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher Manning. Raptor: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations*, volume 2024, pages 32628–32649, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/8a2acd174940dbca361a6398a4f9df91-Paper-Conference.pdf.

[33] Bilgehan Sel, Priya Shanmugasundaram, Mohammad Kachuee, Kun Zhou, Ruoxi Jia, and Ming Jin. Skin-in-the-game: Decision making via multi-stakeholder alignment in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13921–13959, 2024. URL https://aclanthology.org/2024.acl-long.751/.

[34] Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 783–791, 2024. URL https://aclanthology.org/2024.naacl-short.69/.

[35] Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. Is chatgpt good at search? investigating large language models as re-ranking agents. In *Proceedings of the 2023 conference on empirical methods in natural*

language processing, pages 14918–14937, 2023. URL https://aclanthology.org/2023.emnlp-main.923/.

[36] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. URL https://aclanthology.org/2022.tacl-1.31/.

[37] Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 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 (Volume 1: Long Papers)*, pages 11636–11652, 2025. URL https://aclanthology.org/2025.naacl-long.581/.

[38] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022. URL https://doi.org/10.48550/arXiv.2212.03533.

[39] Zichong Wang, Jie Yang, Jun Zhuang, Puqing Jiang, Mingzhe Chen, Ye Hu, and Wenbin Zhang. Fair graph learning with limited sensitive attribute information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 39423–39431, 2026. URL https://doi.org/10.1609/aaai.v40i46.41292.

[40] Hao Wu, Hongru Sun, Wanqing Li, Xinguo Yu, Hao Ming, Xiao Luo, Wenbin Zhang, Jiahong Zhao, Yi Guo, and Jie Yang. FLAIR: Steering LLM mathematical problem solving based on a fuzzy-logic-AssIsted reasoner. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 38638–38661, 2026. URL https://aclanthology.org/2026.acl-long.1790/.

[41] Jing Wu and Mehrtash Harandi. Munba: Machine unlearning via nash bargaining. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1–12, 2025. URL https://doi.org/10.1109/ICCV51701.2025.00452.

[42] Xiaojun Wu, Cehao Yang, Xueyuan Lin, Chengjin Xu, Xuhui Jiang, Yuanliang Sun, Hui Xiong, Jia Li, and Jian Guo. Think-on-graph 3.0: Efficient and adaptive llm reasoning on heterogeneous graphs via multi-agent dual-evolving context retrieval. *arXiv preprint arXiv:2509.21710*, 2025. URL https://doi.org/10.48550/arXiv.2509.21710.

[43] Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pages 641–649, 2024. URL https://doi.org/10.1145/3626772.3657878.

[44] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, 2018. URL https://aclanthology.org/D18-1259/.

[45] Shamim Yazdani, Akansha Singh, Nripsuta Saxena, Zichong Wang, Avash Palikhe, Deng Pan, Umapada Pal, Jie Yang, and Wenbin Zhang. Generative ai in depth: A survey of recent advances, model variants, and real-world applications. *Journal of Big Data*, 12(1):230, 2025. URL https://doi.org/10.1186/s40537-025-01247-x.

[46] Jiahao Ying, Yixin Cao, Kai Xiong, Long Cui, Yidong He, and Yongbin Liu. Intuitive or dependent? investigating llms' behavior style to conflicting prompts. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4221–4246, 2024. URL https://aclanthology.org/2024.acl-long.232/.

[47] Xiaowei Yuan, Zhao Yang, Yequan Wang, Shengping Liu, Jun Zhao, and Kang Liu. Discerning and resolving knowledge conflicts through adaptive decoding with contextual information-entropy constraint. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3903–3922, 2024. URL https://aclanthology.org/2024.findings-acl.234/.

[48] Yi Zeng, Xuelin Yang, Li Chen, Cristian Canton Ferrer, Ming Jin, Michael I. Jordan, and Ruoxi Jia. Fairness-aware meta-learning via nash bargaining. In *Advances in Neural Information Processing Systems*, volume 37, pages 83235–83267, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/97a0f7d7fc2c26732943860241af1c7a-Paper-Conference.pdf.

[49] Nan Zhang, Prafulla Kumar Choubey, Alexander Fabbri, Gabriel Bernadett-Shapiro, Rui Zhang, Prasenjit Mitra, Caiming Xiong, and Chien-Sheng Wu. Sirerag: Indexing similar and related information for multihop reasoning. In *International Conference on Learning Representations*, volume 2025, pages 100413–100432, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/f9668d223e713943634dce9c66e8f2c1-Paper-Conference.pdf.

[50] Qinggang Zhang, Zhishang Xiang, Yilin Xiao, Le Wang, Junhui Li, Xinrun Wang, and Jinsong Su. Faithfulrag: Fact-level conflict modeling for context-faithful retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 21863–21882, 2025. URL https://aclanthology.org/2025.acl-long.1062/.

[51] Wenzheng Zhang, Wenyue Hua, and Karl Stratos. EntQA: Entity linking as question answering. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=US2rTP5nm_.

[52] Wenxuan Zhou, Sheng Zhang, Hoifung Poon, and Muhao Chen. Context-faithful prompting for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14544–14556, 2023. URL https://aclanthology.org/2023.findings-emnlp.968/.

[53] Xiangrong Zhu, Yuexiang Xie, Yi Liu, Yaliang Li, and Wei Hu. Knowledge graph-guided retrieval augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8912–8924, 2025. URL https://aclanthology.org/2025.naacl-long.449/.

[54] Luyao Zhuang, Shengyuan Chen, Yilin Xiao, Huachi Zhou, Yujing Zhang, Hao Chen, Qinggang Zhang, and Xiao Huang. LinearRAG: Linear graph retrieval augmented generation on large-scale corpora. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=mCtfkypdm6.

Learning Minimal Sufficient Evidence Graphs for GraphRAG via Nash-Guided Optimization (Appendix)

Contents

A Notation and Terminology **18**

B Related Work **19**

B.1 GraphRAG 19

B.2 Nash Bargaining 19

C Analysis for Query-Conditioned Knowledge Conflict **21**

D Dataset Details **22**

E Hyperparameter Analysis **23**

E.1 Effect of m and τ_{sem} 23

E.2 Effect of H and Δ 23

E.3 Effect of λ and τ_{stop} 24

E.4 Alternative Minimality measures 25

F Model Robustness via Evidence Pruning **26**

G Details of Seed Initialization **27**

G.1 Query-mentioned Original Entity Extraction 27

G.2 Embedding-based Node Alignment 28

G.3 LLM-based Seed Reranking 29

H Details of Structural and Semantic Expansion **31**

H.1 Structural Expansion 31

H.2 Semantic Expansion 31

H.3 Candidate Edge Update 32

I Utility Scoring and Answer Generation **33**

I.1 Prompt for Answer-Support Scoring 33

I.2 Prompt for Conflict Scoring 36

I.3 Evidence overlap across backbone LLMs 39

J Time Complexity Analysis **40**

K Discussion on Nash-guided optimization **42**

L Limitation and Future Work **44**

A Notation and Terminology

Table 7 summarizes the main notations used in CONSIST. The notations are organized according to the main components of CONSIST: candidate evidence graph construction, conflict-aware graph utility modeling, and Nash-guided Gumbel edge editing for evidence control.

Table 7: Main notations used in CONSIST.

Queries, evidence graphs, and conflict sources	
q	Input query.
$G_t = (\mathcal{V}_t, \mathcal{E}_t)$	Evidence graph at expansion/editing step t, with node set $\mathcal{V}_t$ and edge set $\mathcal{E}_t$.
G_q	Query-specific evidence graph used for answering q.
$\mathcal{G}(q)$	Query-dependent graph space reachable through online expansion and editing.
G_q^*	Final minimal sufficient graph.
$\mathcal{F}_t$	Frontier node set expanded at step t.
Candidate evidence graph construction	
$\mathcal{V}_{\text{cand}}$	Candidate nodes obtained by query-aware entity extraction and/or embedding-based node alignment.
$\mathcal{V}_{\text{seed}}$	Seed node set selected from $\mathcal{V}_{\text{cand}}$.
m	Number of selected seed nodes.
$G_0 = (\mathcal{V}_{\text{seed}}, \varnothing)$	Initial evidence graph, where no edges are included at initialization.
$\mathcal{N}_{\text{str}}(v)$	Structural neighbors of node v induced by existing graph edges.
$\mathcal{N}_{\text{sem}}(v)$	Semantic neighbors of node v instantiated from semantic relatedness.
$s(v, u)$	Semantic relatedness score between nodes v and u.
τ_{sem}	Threshold for retaining semantic neighbors during expansion.
$G_{t+1}^{(\text{can})}$	Expanded candidate graph before edge editing.
Conflict-aware graph utility	
$\mathcal{S}(G, q)$	Sufficiency score measuring how well graph G supports answering query q.
$\mathcal{C}(G, q)$	Consistency score measuring how conflict-free graph G is under query q.
$\mathcal{M}(G)$	Minimality score encouraging compact evidence graphs.
$U(G, q)$	Overall graph utility jointly combining sufficiency, consistency, and minimality.
λ	Utility trade-off weight between consistency and minimality.
$\text{LLM}_{\text{sup}}(G, q)$	LLM-based scorer used for sufficiency estimation.
$\text{LLM}_{\text{pair}}(q, (e_i, e_j))$	LLM-based pairwise conflict scorer.
Nash-guided Gumbel edge editing	
$\boldsymbol{z}$	Binary edge-retention vector, where $z_j = 1$ means the j-th candidate edge is retained.
$\boldsymbol{\theta}$	Learnable edge-gate parameters for differentiable graph editing.
$\text{GS}(\cdot)$	Gumbel-Softmax sampling.
Δ	Gumbel-Softmax temperature.
$\text{Edit}(G_{t+1}^{(\text{can})}, \boldsymbol{\theta}_{t+1})$	Edge-level editing operator that edits candidate edges.
u_{cur}	Current utility player in Nash bargaining, measuring immediate graph structures.
u_{fut}	Future utility player in Nash bargaining, measuring future graph structures.
H	Rollout steps.
d_t	Disagreement point at step t.
$\mathcal{U}_t(\theta)$	Feasible utility region induced by the relaxed edit space.
τ_{stop}	Termination threshold for stopping graph editing.
Answer generation	
$\text{LLM}(G_q^*, q)$	Final answer generator given the input G_q^* and q.
$\hat{a}$	Final generated answer.

B Related Work

B.1 GraphRAG

Retrieval-Augmented Generation (RAG) augments Large Language Models (LLMs) with external knowledge by retrieving and incorporating relevant evidence during generation [17]. Graph-enhanced RAG (GraphRAG) further organizes retrieved evidence as a graph, enabling reasoning over interconnected evidence rather than isolated textual chunks [5, 53]. Existing GraphRAG methods can be broadly categorized into two complementary settings: Offline and Online Graph Construction.

Offline GraphRAG. This line of work constructs a static evidence graph from the corpus in advance, and performs query-time reasoning by retrieving a query-relevant subgraph. Existing methods mainly differ in the graph construction strategy. For instance, GraphRAG [5] and HippoRAG [7] employ LLMs to extract entities and relations, forming a structured entity–relation graph that captures semantic connections across documents. LightRAG [6] follows a similar pipeline but further grounds graph nodes in their supporting textual contexts. RAPTOR [32] instead organizes document nodes via recursive clustering and bottom-up summarization, producing a hierarchical-tree graph. In addition, LinearRAG [54] constructs a lightweight hierarchical graph via entity-centric representations with semantic linking. During retrieval, the query is matched against graph nodes and/or their associated textual descriptions to identify relevant evidence. *However, since the graph structure is fixed prior to reasoning, these methods cannot adaptively expand or refine the evidence graph for a given query during retrieval.*

Online GraphRAG. In contrast, this line of work incrementally constructs and refines the evidence graph during retrieval, allowing the structure to evolve with respect to the target query rather than being fixed in advance. Compared to offline approaches, where the graph is largely determined by corpus-level construction procedures, these methods adapt both the graph structure and the reasoning trajectory to query-specific needs. Typically, these approaches expand the evidence graph through iterative exploration guided by LLMs, including path planning [22], tool-augmented graph interactions [25], and hybrid graph–text navigation [23]. Additional evidence can be incorporated on demand, either by retrieving new passages [14] or by inferring missing evidence connections [11]. Therefore, compared to offline approaches, where the graph is largely determined by corpus-level construction procedures, online methods adapt both the graph structure and the reasoning trajectory to the target query, progressively forming a query-specific evidence graph. *However, these methods often overlook knowledge conflicts among evidence: as the graph grows incrementally, newly introduced evidence may contradict or interfere with previously accumulated information.*

Knowledge Conflict in (Graph)RAG. Knowledge conflict naturally arises in (Graph)RAG when multiple retrieved documents provide inconsistent or incompatible information for the same query. Such conflicts typically appear as noisy evidence [34], stale information [21], semantic ambiguity [16], or erroneous matching [51]. To address this issue, prior work either steers generation toward more reliable evidence via context-aware control [34], adaptive decoding [37], plug-and-play probability adjustment [2], soft-prompt-based reliability modulation [4], and structured decoding [18], or explicitly equips the model with conflict reasoning capabilities to identify, compare, and reconcile inconsistent evidence before producing the final answer [12, 50, 39, 40, 45].

However, *these approaches are developed for standard RAG settings, where evidence is treated as an unordered set of retrieved chunks, and to the best of our knowledge, they have not been extended to GraphRAG*. Applying them to GraphRAG is also *non-trivial* due to the structured and evolving nature of graph-based evidence. In an incrementally constructed graph, newly introduced evidence affects not only answer-level consistency, but also the graph topology itself, including node retention or structural connectivity. To address this issue, we propose a novel *Nash-Optimized Minimal Subgraph Construction* algorithm.

B.2 Nash Bargaining

Nash bargaining is introduced in [27, 28] to model cooperative decision-making in bargaining situations, where multiple players seek an agreement that improves upon a disagreement outcome. It provides a principled mechanism for balancing competing utilities by selecting solutions that maximize the joint benefit of all participants while discouraging certain dominance. In a standard utility-based formulation, given a feasible utility set $\mathcal{U}$ and a disagreement point d, the Nash bargaining

solution can be calculated as

$$u^* = \arg\max_{u \in \mathcal{U}} \sum_i \log(u_i - d_i), \quad \text{s.t. } \forall i: \ u_i > d_i. \tag{15}$$

The classical Nash bargaining solution is typically characterized by four axioms: Pareto optimality, symmetry, independence of irrelevant alternatives, and invariance to positive affine transformations. Pareto optimality requires that no other feasible agreement can improve one player's utility without decreasing another's. Symmetry states that players with identical bargaining roles should receive equal treatment in the solution. Independence of irrelevant alternatives requires that the selected solution remains unchanged when infeasible or unselected alternatives are removed from the feasible set, as long as the original solution remains feasible. Invariance to positive affine transformations ensures that the solution is unaffected by equivalent rescaling or shifting of utility functions. Overall, *they provide an axiomatic justification for the Nash product as a principled criterion for selecting a balanced agreement among competing players*.

Recently, Nash-inspired methods have been adopted in various downstream tasks to balance competing objectives, including multi-task learning [29], fairness-aware meta-learning [48], machine unlearning [41], and expert merging [30]. We extend this idea to GraphRAG, introducing Nash-guided editing to balance sufficiency, consistency, and minimality.

C Analysis for Query-Conditioned Knowledge Conflict

This section details the conflict analysis supporting Fig. 2.

Background. ConFiQA [1] is a multi-hop conflict benchmark designed to evaluate context-faithfulness in RAG scenarios with knowledge conflicts. ConFiQA contains QA, MR, and MC splits, where **MC contains multi-hop questions with multiple related counterfactual facts as the most challenging case**. This setting is close to online GraphRAG, where expanded evidence may contain multiple conflicting nodes or edges along the reasoning path.

ConFiQA evaluates whether a model follows the provided context or instead relies on its parametric knowledge from the backbone LLM. It categorizes responses into: (1) **Context-Faithful**, where the answer follows the given context; (2) **Stubborn**, where the model relies on its parametric knowledge rather than the context; and (3) **Other**, where the response is neither clearly context-faithful nor stubborn. Accordingly, our analysis focuses on **Stubborn** and **Other** responses, as they reflect model behavior under context–knowledge conflict.

Procedure. LLM-generated responses do not directly reveal the underlying causes of errors (*e.g.*, whether they arise from misleading evidence or conflicts). Therefore, we further annotate error responses from the evidence perspective. Specifically, we collect errors produced by LLaMA3-8B on ConFiQA-MC and uniformly sample 1,000 erroneous responses. Each case includes the question, conflicting context, context-faithful answer, original factual answer, model response, and response-level error type. For each sampled case, we assign one evidence-side conflict source: **S1: Retrieval noise**, where irrelevant or incorrectly linked evidence distracts the reasoning path; **S2: Counterfactual fact**, where intentionally injected counterfactual evidence is present; **S3: Semantic ambiguity**, where ambiguous entities, relations, or multi-hop paths lead to query-misaligned reasoning; and **Unknown**, for errors that cannot be confidently attributed to S1–S3.

Evidence conflict source	Stubborn	Other	Total
S1: Retrieval noise	71	152	223 (22.3%)
S2: Counterfactual fact	372	174	546 (54.6%)
S3: Semantic ambiguity	53	129	182 (18.2%)
Unknown	16	33	49 (4.9%)
Total	512 (51.2%)	488 (48.8%)	1,000 (100.0%)

Table 8: Evidence-side annotation of 1,000 erroneous responses sampled from ConFiQA-MC. Context-faithful responses are excluded as correct cases. The remaining errors are categorized into two response-level types, *Stubborn* and *Other*, while our annotation identifies the dominant evidence-side conflict source.

Summary. Table 8 shows how each conflict source is distributed across response-level error types. Overall, S1–S3 account for 22.3%, 54.6%, and 18.2% of the sampled errors, respectively, with the remaining 4.9% assigned to the Unknown category.

These results indicate that a large majority of errors can be attributed to identifiable evidence-side conflicts, highlighting that conflict is both prevalent and systematic. In particular, the dominance of S2 suggests that counterfactual or misleading evidence is a primary source of reasoning failure, highlighting the necessity of modeling and resolving conflicts in GraphRAG.

D Dataset Details

This section summarizes the datasets used in our experiments, covering both multi-hop question answering and conflict-oriented faithfulness benchmarks.

Multi-hop QA datasets. We use 2Wiki, HotpotQA, and MuSiQue for multi-hop QA. 2Wiki provides annotated evidence along multi-hop reasoning paths [9]. HotpotQA offers sentence-level supporting facts for explainable multi-hop reasoning [44]. MuSiQue contains 2–4 hop questions constructed via compositional single-hop question generation [36]. Together, these datasets evaluate a model's ability to dynamically construct compact and sufficient evidence graphs for multi-step reasoning. To make a fair comparison, following prior work [7, 20, 49], we sample 1,000 examples from each dataset, and further construct the corpus-level graph by first segmenting documents according to dataset-specific supporting units (*e.g.*, sentence-level for 2Wiki and HotpotQA) to preserve semantic integrity. Each segment is encoded using the *BGE-base* model, and key entities are extracted via POS tagging with PaddleNLP. The graph is stored in Neo4j, where edges are generated through LLM prompting, producing both incoming and outgoing question-based links to ensure sufficient coverage of each node. Table 9 summarizes the target dataset and resulting graph statistics.

Table 9: Statistics of the sampled multi-hop QA datasets and constructed corpus-level graphs, where *Docs* denotes the number of related documents or passages, *Supporting Facts* denotes the annotated gold evidence units, and *Vertices* and *Edges* indicate the numbers of passage-level nodes and directed evidence links in the graph, respectively.

Dataset	Questions	Docs	Supporting Facts	Vertices	Edges	Avg. Text Length	Avg. Edge Number
2Wiki	1000	10000	2388	23360	167068	116.07	7.15
HotpotQA	1000	9942	2458	40534	171946	132.44	4.24
MuSiQue	1000	19990	2800	13086	81348	489.52	6.22
Average	1000	13311	2549	25660	140121	246.01	5.87

As observed from Table 9, MuSiQue yields relatively compact evidence graphs after preprocessing, while HotpotQA exhibits broader but comparatively sparser evidence coverage. In contrast, 2Wiki produces denser graphs with richer connectivity along multi-hop reasoning paths, making it a more challenging setting for evaluating whether a method can identify useful reasoning links while pruning redundant or distracting edges. **We therefore select 2Wiki as the primary dataset for our ablation analysis**.

Faithfulness and conflict-oriented datasets. We use ConFiQA and FaithEval as conflict-oriented benchmarks. ConFiQA evaluates context faithfulness under knowledge conflicts and includes three subsets: QA, MR, and MC, each containing 6,000 instances [1]. FaithEval assesses contextual faithfulness across unanswerable, inconsistent, and counterfactual settings, with a total of 4,900 samples [26]. Dataset statistics are summarized in Table 10.

Table 10: Statistics of conflict-oriented benchmarks.

Dataset	Subset	#Examples	Annotation / Evidence Type
ConFiQA	QA	6,000	Single-hop questions with one counterfactual
ConFiQA	MR	6,000	Multi-hop reasoning with one related counterfactual
ConFiQA	MC	6,000	Multi-hop reasoning with multiple related counterfactuals
FaithEval	Unanswerable Context	2,400	Context lacks sufficient evidence for the answer
FaithEval	Inconsistent Context	1,500	Context contains conflicting answers
FaithEval	Counterfactual Context	1,000	Context contradicts world knowledge

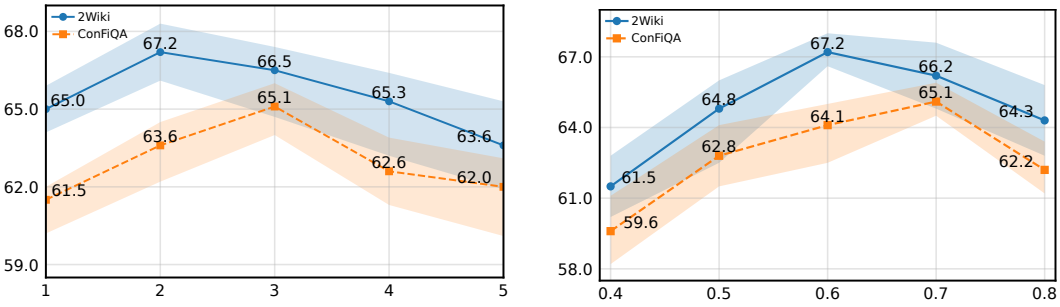
ConFiQA and FaithEval complement multip-hop QA benchmarks by evaluating whether models remain stable under counterfactual or conflicting evidence.

E Hyperparameter Analysis

For CONSIST, we consider six hyperparameters: the number of seeds (m), the semantic expansion threshold τ_{sem}, the utility weight λ, the rollout step H, the Gumbel-Softmax temperature Δ, and the termination threshold τ_{stop}. This section provides a detailed sensitivity analysis of these parameters. We select H and τ_{stop} via grid search, with their sensitivity analyzed below. The agreement set in Eq. (14) is the feasible domain of the edge-selection parameters, rather than a tunable hyperparameter. Additional ablations of new-edge initialization and utility weights are reported in Tables 4 and 5, respectively.

E.1 Effect of m and τ_{sem}

To begin with, we analyze how candidate graph construction affects CONSIST from two key factors: the number of seeds m and the semantic expansion threshold τ_{sem}. For 2Wiki, we report the GPT-4o results, while for ConFiQA(-MC), we report the LLaMA-3-8B results, following the backbone settings used in the main experiments. For m and τ_{sem}, different values are explored.



(a) Number of seeds (m). (b) Semantic expansion threshold τ_{sem}.

Figure 6: Effect of candidate graph construction using the number of seeds m and semantic expansion threshold τ_{sem}.

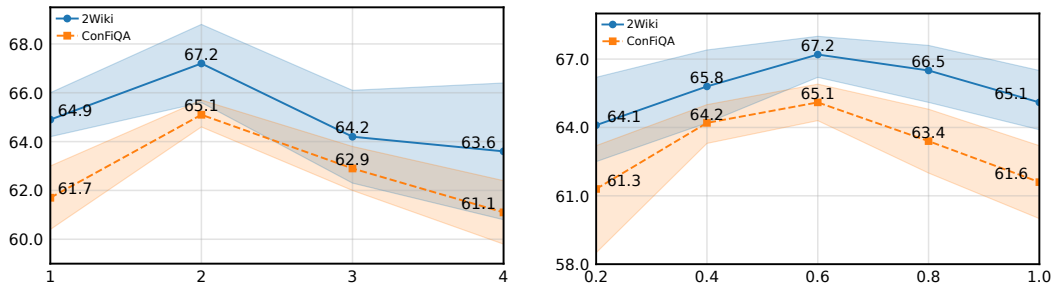
As shown in Fig. 6(a), a similar trend is observed for m across two datasets, although the optimal values differ slightly. For 2Wiki, increasing the number of seeds from m=1 to m=2 improves EM from 65.0% to 67.2%, after which performance gradually declines to 66.5%, 65.3%, and 63.6% for m=3, 4, 5, respectively. This suggests that introducing a second seed helps recover missing query-critical evidence, while too many seeds introduce distracting or weakly related starting points, reducing expansion effectiveness. On ConFiQA, EM increases from 61.5% at m=1 to 63.6% at m=2, peaks at 65.1% for m=3, and then drops to 62.6% and 62.0% at m=4 and m=5. This indicates that conflict-oriented reasoning benefits from slightly broader initial coverage, but excessive seeds still introduce ambiguity and noise. Overall, these results show that a small seed set is sufficient, while overly large seed sets tend to degrade evidence quality.

In addition, Fig. 6(b) shows the effect of the semantic expansion threshold τ_{sem}. For 2Wiki, EM improves from 61.5% at τ_{sem}=0.4 to a peak of 67.2% at 0.6, and then decreases to 66.2% and 64.3% at 0.7 and 0.8. For ConFiQA, EM increases from 59.6% at 0.4 to 62.8% and 64.1% at 0.5 and 0.6, reaches the best result of 65.1% at 0.7, and drops to 62.2% at 0.8. These results indicate that a small threshold leads to aggressive expansion and introduces unnecessary neighbors, while a large threshold makes expansion overly conservative and may miss useful reasoning links. Overall, a moderate threshold achieves the best trade-off between evidence coverage and noise control.

E.2 Effect of H and Δ

In graph editing, we employ Gumbel-Softmax to enable differentiable edge selection and incorporate future rollout to estimate long-term utility. Accordingly, we analyze the sensitivity of the rollout step H and the Gumbel-Softmax temperature Δ hereafter. Again, the 2Wiki and ConFiQA(-MC) datasets are considered.

As observed from Fig. 7(a), the best performance is obtained at $H = 2$, indicating that a short future-aware rollout is sufficient to evaluate whether an edit remains beneficial after subsequent



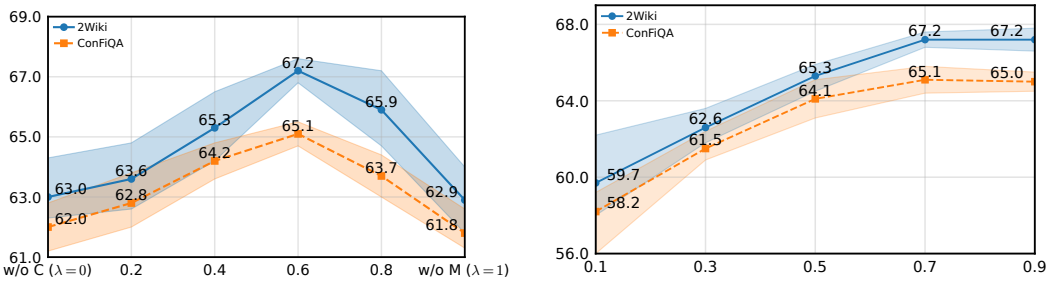
(a) Rollout step H. (b) Gumbel-Softmax temperature Δ.

Figure 7: Hyperparameter sensitivity analysis of the rollout horizon H (controlling look-ahead steps in Nash-guided graph editing) and the Gumbel-Softmax temperature Δ.

graph expansion. When $H = 1$, the future signal is still under-explored, so the editing decision may retain edges that appear useful at one expansion but become harmful with more deep expansion. In contrast, larger horizons introduce more uncertainty during rollout and may accumulate noisy or query-irrelevant candidates, also leading to less reliable future-utility estimation. Additionally, Fig. 7(b) studies the Gumbel-Softmax temperature Δ. The peak at $\Delta = 0.6$ shows that a moderate relaxation provides the best balance between stable optimization and selective edge pruning. A smaller temperature makes edge editing close to hard binary pruning and can cause unstable early edits, whereas a larger temperature weakens the ability to remove conflicting or redundant evidence. Overall, these results indicate that a moderate temperature is crucial for balancing optimization stability and effective evidence filtering.

E.3 Effect of λ and τ_{stop}

We then analyze λ and τ_{stop} as both parameters directly govern the graph-editing process. Specifically, λ controls the trade-off between consistency and minimality in the utility function (*i.e.*, how the graph is edit), while τ_{stop} determines when the editing process terminates.



(a) Utility hyperparameter λ. (b) Termination threshold τ_{stop}.

Figure 8: Impact from the utility trade-off λ and termination threshold τ_{stop}. Again, the datasets of 2Wiki and ConFiQA(-MC) are leveraged.

In Fig. 8(a), performance first increases and then decreases as λ grows, with the best results achieved at $\lambda = 0.6$ for both 2Wiki and ConFiQA. This suggests that consistency should be slightly emphasized when handling query-conditioned conflicts, while the minimality term remains necessary to prevent over-expanded evidence graphs. Removing either consistency (λ=0) or minimality (λ=1) degrades performance, indicating that the two terms are complementary. Moreover, as observed in Fig. 8(b), performance improves as τ_{stop} increases from 0.1 to 0.5 on both datasets, indicating that early stopping may leave the evidence graph under-developed. When τ_{stop} further increases to 0.7 and beyond, performance stabilizes, suggesting that the graph has reached a sufficiently refined structure.

E.4 Alternative Minimality measures

To examine alternative compactness measures, we further replace the edge-based term with a node-growth ratio $|V(G')|/|V(G)|$, while keeping all other components unchanged. Table 11 compares this variant with the embedding-based measure M_{emb} and our edge-growth ratio.

Table 11: Impact analysis of alternative Minimality measures.

Minimality measure	2Wiki		ConFiQA	
	EM(↑)	F1(↑)	EM(↑)	F1(↑)
M_{emb}	66.5	74.2	63.1	71.7
Node-growth ratio	66.7	74.8	64.5	73.2
Edge-growth ratio (Ours)	**67.2**	**75.7**	**65.1**	**73.9**

As shown in Table 11, both alternative measures remain competitive, while the proposed edge-growth ratio consistently achieves the best performance. We attribute this advantage to its closer alignment with the proposed edge-level editing, providing a more immediate compactness measure. In contrast, the node-growth ratio may remain unchanged if multiple redundant edges are introduced between existing nodes. These results support the edge-growth ratio as a simple yet effective choice for Minimality.

F Model Robustness via Evidence Pruning

We evaluate robustness under evidence perturbation to assess how sensitive the proposed method is to changes in the constructed evidence graph. This analysis reveals whether retained edges are redundant or task-critical, and whether the model relies on compact, high-quality evidence or redundant structures. Specifically, we conduct the experiment on 2WIKI using GPT-4O as the backbone, and compare CONSIST with SIRERAG and HOPRAG. For each case, we randomly remove $\xi\%$ of edges from the final evidence graph, with $\xi \in \{0\%, 10\%, 20\%, 30\%, 40\%\}$, and evaluate the resulting performance.

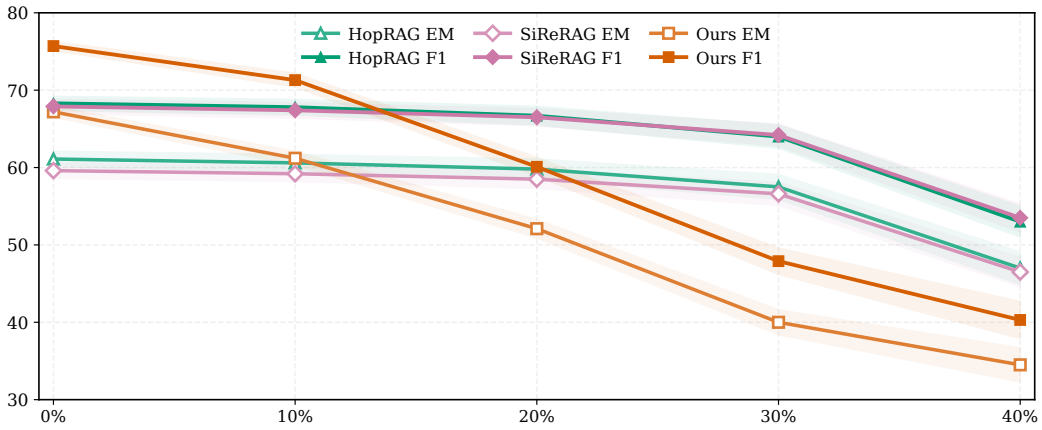


Figure 9: Robustness analysis of GraphRAG methods on 2Wiki with GPT-4o.

Fig. 9 illustrates the EM and F1 results on 2WIKI after randomly pruning edges from the final evidence graph. As expected, CONSIST degrades more sharply than baseline methods. This is because CONSIST explicitly removes redundant and conflicting branches during graph construction, resulting in a higher proportion of task-critical edges. In contrast, baseline methods retain larger graphs with more redundant evidence and therefore exhibit smoother degradation under moderate pruning. These findings are consistent with the analysis of **Graph Consistency and Minimality** (from Section 4.3), indicating that the evidence selected by CONSIST is more compact and less redundant: once these critical edges are removed, performance drops more quickly due to the lack of duplicated reasoning paths.

G Details of Seed Initialization

This appendix provides the full seed initialization pipeline used by CONSIST. Seed initialization contains three stages: (i) query-mentioned entity extraction, (ii) embedding based node alignment, and (iii) LLM-based seed reranking. The purpose of this stage is not to infer missing bridge entities or answer-side nodes, but to identify reliable entry points explicitly grounded in the query. Bridge entities and latent answer paths are discovered later through structural and semantic expansion. The final scalar used for seed selection is `candidate_assessment.seed_score`, returned by $\mathrm{LLM}_{\mathrm{seed}}(q, v, \mathcal{V}_o, \mathcal{A}_q(v))$, where $\mathcal{V}_o$ denotes the set of original query-mentioned entity records extracted from q, and $\mathcal{A}_q(v) \subseteq \mathcal{V}_o$ denotes the subset of extracted query mentions that are aligned to the candidate graph node v.

G.1 Query-mentioned Original Entity Extraction

Given a query q, we first prompt the backbone LLM to produce a compact set of query-mentioned original entity records $\mathcal{V}_o = \{v_o^{(1)}, \ldots, v_o^{(r)}\}$. Following prior GraphRAG pipelines that use LLMs to extract or organize entity-centric anchors for graph construction and graph-based retrieval [5–7], each $v_o \in \mathcal{V}_o$ corresponds to an entity or noun phrase explicitly mentioned in q and contains its surface form, canonical name, aliases, entity type, query role, and retrieval-oriented description. We intentionally do not ask the LLM to hypothesize unmentioned bridge entities at this stage, since doing so may inject unsupported nodes before graph expansion. Instead, seed initialization only provides high-precision query anchors from which CONSIST subsequently expands to neighboring structural or semantic evidence.

Prompt for Query-Mentioned Original Entity Extraction

```
System:
You are an entity extractor for query-aware graph retrieval.
Your task is to extract a compact set of original query-mentioned entities that can
     serve as high-precision seed anchors for graph retrieval.

You must identify:
- entities or noun phrases explicitly mentioned in the query,
- their canonical forms and aliases,
- their entity types,
- their roles in the query,
- retrieval descriptions useful for matching them to graph nodes.

Do NOT answer the query.
Do NOT infer hidden bridge entities.
Do NOT invent unmentioned entities.
Do NOT include answer-side entities unless they are explicitly mentioned in the query.
Your output should only contain entities or noun phrases grounded in the query text.

User:
Query q:
{q}

Return EXACTLY one JSON object:
{
  "query_analysis": {
    "target_answer_slot": "...",
    "reasoning_type": "<single-hop / multi-hop / comparison / bridge / constraint>",
    "explicit_query_mentions": ["..."]
  },
  "vo": [
    {
      "surface_form": "...",
      "canonical_name": "...",
      "aliases": ["..."],
      "entity_type": "...",
```

```
      "query_role": "<topic_entity / constraint_entity / relation_mention /
     answer_type_mention>",
      "mention_span": "...",
      "retrieval_description": "...",
      "confidence": 0.0
    }
  ]
}

Guidelines:
1. Extract only entities or noun phrases explicitly grounded in the query.
2. Normalize surface forms into canonical names when possible.
3. Include aliases only when they are standard variants of the same mentioned entity.
4. Do not include implicit bridge entities, missing intermediate entities, or guessed
     answers.
5. Keep the original entity set small; usually 1--4 records are sufficient.
6. All numeric fields must be JSON numbers in [0,1], not strings.
7. All boolean fields must be JSON booleans, not strings.

Generate the output in the exact JSON format above.
- Output ONLY the JSON-format result.
- Do not include any other text or explanations.
```

G.2 Embedding-based Node Alignment

Each graph node $v \in \mathcal{V}$ is represented by a textual node description c_v, which concatenates the node name, aliases, type, description, and supporting context when available:

$$c_v = \mathrm{cat}\big(\mathrm{name}(v), \mathrm{alias}(v), \mathrm{type}(v), \mathrm{desc}(v), \mathrm{ctx}(v)\big).$$

We encode c_v once offline using a dense text encoder $\phi(\cdot)$, implemented with an off-the-shelf dense retriever or embedding model such as DPR, E5, or BGE [15, 38, 43]. For each extracted original query mention $v_o \in \mathcal{V}_o$, we construct a short retrieval query r_{v_o} by concatenating its surface form, canonical name, aliases, type, retrieval description, and the original query:

$$r_{v_o} = \mathrm{cat}\big(\mathrm{surf}(v_o), \mathrm{canon}(v_o), \mathrm{alias}(v_o), \mathrm{type}(v_o), \mathrm{desc}(v_o), q\big).$$

The dense alignment score between the query-side mention v_o and a graph node v is computed as

$$s_{\mathrm{emb}}(v_o, v) = \cos\big(\phi(r_{v_o}), \phi(c_v)\big).$$

Since named-entity matching often benefits from both semantic and lexical signals, we combine dense retrieval with lexical matching, following the common hybrid retrieval practice in open-domain QA and RAG [15, 31]:

$$s_{\mathrm{align}}(v_o, v) = \alpha s_{\mathrm{emb}}(v_o, v) + (1-\alpha)\widetilde{s}_{\mathrm{BM25}}(r_{v_o}, c_v) + \beta \mathbb{I}[\mathrm{alias}(v_o) \cap \mathrm{alias}(v) \neq \varnothing]\,.$$

Here, $\widetilde{s}_{\mathrm{BM25}}$ is a normalized BM25 score, $\alpha \in [0,1]$ controls the dense–lexical trade-off, and the last term rewards exact surface-form or alias matches. For each $v_o \in \mathcal{V}_o$, we retrieve the top- k_{align} graph nodes under s_{align}:

$$\mathcal{R}(v_o) = \mathrm{TopK}_{k_{\mathrm{align}}}\left(\{s_{\mathrm{align}}(v_o, v)\}_{v \in \mathcal{V}}\right).$$

The candidate seed node set is the deduplicated union over all extracted original query mentions:

$$\mathcal{V}_{\mathrm{cand}} = \mathrm{Dedup}\left(\bigcup_{v_o \in \mathcal{V}_o} \mathcal{R}(v_o)\right).$$

If multiple query mentions retrieve the same graph node, we keep all matched mentions as alignment evidence for the LLM reranker.

G.3 LLM-based Seed Reranking

The alignment retriever is designed to maximize recall and may include aliases, partial matches, or semantically related but query-irrelevant nodes. We therefore rerank every candidate node $v \in \mathcal{V}_{\text{cand}}$ using an LLM-based seed scorer, following recent work showing that LLMs can serve as effective query-aware rerankers when given a query and a small candidate set [35]:

$$\rho_q(v) = \text{LLM}_{\text{seed}}\left(q, v, \mathcal{V}_o, \mathcal{A}_q(v)\right).$$

The value $\rho_q(v)$ is instantiated as `candidate_assessment.seed_score`. The final seed set is

$$\mathcal{V}_{\text{seed}} = \{v \in \mathcal{V}_{\text{cand}} \mid \rho_q(v) \in \text{TopK}_m\{\rho_q\}\}.$$

The initial graph and frontier are then initialized as $G_0 = (\mathcal{V}_{\text{seed}}, \varnothing)$ and $\mathcal{F}_0 = \mathcal{V}_{\text{seed}}$.

Prompt for Seed Set Selection

```
System:
You are a verifier for query-aware seed scoring in graph-based retrieval.
Your task is to assess how suitable a candidate graph node is as an initial seed for
     answering the query.

The seed stage only selects high-precision entry points explicitly grounded in the
     query.
Do NOT infer hidden bridge entities.
Do NOT reward a node merely because it may become useful after expansion.
Do NOT answer the question itself.
Do NOT score generic popularity or graph degree.
Score the node only by its usefulness as an initial query anchor.

User:
Query q:
{q}

Generated original query mentions Vo:
{vo}

Candidate graph node v:
{node_name}
{node_type_if_available}
{node_description_if_available}
{supporting_context_if_available}

Retriever evidence:
- matched_original_mentions: {matched_vo}
- dense_alignment_score: {dense_alignment_score}
- lexical_alignment_score: {lexical_alignment_score}
- exact_alias_match: {true_or_false}

Return EXACTLY one JSON object:
{
  "query_requirements": {
    "explicit_query_mentions": ["..."],
    "target_answer_slot": "...",
    "required_seed_roles": [
      "topic_entity",
      "constraint_entity",
      "relation_mention",
      "answer_type_mention"
    ]
  },
  "candidate_assessment": {
    "node": "...",
    "seed_score": 0.0,
```

```
    "seed_role": "<topic_entity / constraint_entity / relation_mention /
     answer_type_mention / peripheral_entity / distractor>",
    "matches_explicit_query_mention": true,
    "canonicalizes_query_mention": true,
    "supports_initial_graph_entry": true,
    "alignment_confidence": 0.0,
    "risk_of_query_toxicity": 0.0
  },
  "reasoning": {
    "why_useful_as_seed": "...",
    "why_not_useful_as_seed": "...",
    "likely_initial_retrieval_role": "..."
  }
}

Scoring rubric for seed_score:
- [0.0, 0.2): almost unusable as a seed.
  Example: the node does not match any explicit query mention and is likely a
     distractor.
- [0.2, 0.4): weak seed.
  Example: the node is topically related but does not reliably canonicalize a
     query-mentioned entity.
- [0.4, 0.6): moderate seed.
  Example: the node partially matches a query mention, but the alignment is ambiguous.
- [0.6, 0.8): strong seed.
  Example: the node clearly matches or canonicalizes an explicit query mention.
- [0.8, 1.0): highly preferred seed.
  Example: the node is a central query-mentioned entity and provides a reliable entry
     point for graph expansion.

Scoring guidelines:
1. Prefer nodes that correspond to explicitly mentioned query entities or noun
     phrases.
2. Reward canonical nodes that resolve aliases or surface-form variants of query
     mentions.
3. Use retriever evidence only as alignment evidence; do not blindly trust high
     embedding similarity.
4. Penalize nodes that require guessing an unmentioned bridge entity.
5. Penalize nodes that are salient but not grounded in the query text.
6. Penalize nodes that mainly lead to factual but query-toxic branches.
7. All numeric fields must be JSON numbers in [0,1], not strings.
8. All boolean fields must be JSON booleans, not strings.
9. The scalar used for seed selection is "candidate_assessment.seed_score".

Selection rule:
- The final seed set is obtained by ranking all candidate nodes by
     "candidate_assessment.seed_score" and selecting the top-m nodes.
- Multiple nodes may receive high seed_score when the query contains multiple
     complementary explicit anchors.

Generate the output in the exact JSON format above.
- Output ONLY the JSON-format result.
- Do not include any other text or explanations.
```

H Details of Structural and Semantic Expansion

This appendix provides implementation details for candidate expansion. At iteration t, CONSIST expands the current candidate graph $G_t = (\mathcal{V}_t, \mathcal{E}_t)$ from the frontier $\mathcal{F}_t$ using two complementary sources: structural edges already present in the explicit graph and semantic edges instantiated from latent document-level relations. Structural expansion preserves high-precision observed connectivity, while semantic expansion repairs missing bridges that are not explicitly encoded in the graph. This design follows the recent query-driven graph construction paradigm, where a query-specific evidence graph is constructed on the fly rather than being restricted to a static corpus-level graph [11, 14].

H.1 Structural Expansion

Structural expansion uses the existing edge set $\mathcal{E}$ as a high-precision backbone. For each frontier node $v_i \in \mathcal{F}_t$, we retrieve its one-hop structural neighbors from both outgoing and incoming observed edges:

$$\mathcal{N}_{\mathrm{str}}(v_i) = \{(v_j, r) \mid (v_i, r, v_j) \in \mathcal{E}\} \cup \{(v_j, r^{-1}) \mid (v_j, r, v_i) \in \mathcal{E}\}.$$

When an observed edge is matched in the reverse direction, we keep the direction explicit by using the inverse relation r^{-1}. The structural candidate edges proposed from v_i are then defined as

$$\mathcal{E}_t^{\mathrm{graph}}(v_i) = \{(v_i, r, v_j, \mathrm{str}, \mathrm{prov}(v_i, r, v_j)) \mid (v_j, r) \in \mathcal{N}_{\mathrm{str}}(v_i)\},$$

where the field str records that the edge is inherited from the explicit graph, and $\mathrm{prov}(v_i, r, v_j)$ denotes the original graph triple and its associated supporting text when available. The structural candidate set at step t is

$$\mathcal{E}_t^{\mathrm{graph}} = \bigcup_{v_i \in \mathcal{F}_t} \mathcal{E}_t^{\mathrm{graph}}(v_i).$$

To avoid expanding many topically related but answer-irrelevant neighbors, we optionally keep only the top-k_{str} structural edges per frontier node according to a query-aware edge score:

$$\rho_q(e) = \eta_1 \cos\big(\phi(q), \phi(\mathrm{prov}(e))\big) + \eta_2 \cos\big(\phi(q), \phi(c_{\mathrm{tail}(e)})\big),$$

where $e = (v_i, r, v_j, \mathrm{str}, \mathrm{prov}(e))$, $\mathrm{tail}(e) = v_j$, $c_{\mathrm{tail}(e)}$ is the textual description of the tail node, and $\eta_1 + \eta_2 = 1$. This deterministic scoring function keeps the expansion reproducible and avoids introducing an additional LLM scorer that is not part of the main utility definition.

H.2 Semantic Expansion

Structural expansion alone may fail when the explicit graph is incomplete. To recover missing bridges, CONSIST constructs a latent relation pool from document-level statistics. Let c_{v_i} denote the textual description of node v_i, including its name, aliases, type, description, and supporting context. We build a pool of candidate latent pairs:

$$\mathcal{P}_{\mathrm{lat}} = \{(v_i, v_j, \Gamma_{ij}) \mid \widetilde{\mathrm{PMI}}(v_i, v_j) \geq \tau_{\mathrm{pmi}} \text{ or } \cos(\phi(c_{v_i}), \phi(c_{v_j})) \geq \tau_{\mathrm{emb}}\},$$

where Γ_{ij} denotes the set of sentences or passages in which v_i and v_j co-occur or are semantically linked, $\phi(\cdot)$ is the text encoder, $\widetilde{\mathrm{PMI}}$ is the normalized PMI score, τ_{pmi} is the co-occurrence threshold, and τ_{emb} is the embedding-similarity threshold. This pool is high-recall: a pair in $\mathcal{P}_{\mathrm{lat}}$ is not treated as a factual edge yet, but only as a candidate implicit connection.

For each frontier node $v_i \in \mathcal{F}_t$, we score every latent pair (v_i, v_j, Γ_{ij}) by combining endpoint similarity, query-provenance relevance, and normalized co-occurrence strength:

$$s(v_i, v_j \mid q) = \beta_1 \cos(\phi(c_{v_i}), \phi(c_{v_j})) + \beta_2 \max_{\gamma \in \Gamma_{ij}} \cos(\phi(q), \phi(\gamma)) + \beta_3 \widetilde{\mathrm{PMI}}(v_i, v_j),$$

where $\beta_1, \beta_2, \beta_3 \geq 0$ and $\beta_1 + \beta_2 + \beta_3 = 1$. The semantic neighbor set is then defined as

$$\mathcal{N}_{\mathrm{sem}}(v_i) = \mathrm{TopK}_{k_{\mathrm{sem}}}\{v_j \in \mathcal{V} \mid (v_i, v_j, \Gamma_{ij}) \in \mathcal{P}_{\mathrm{lat}},\ s(v_i, v_j \mid q) \geq \tau_{\mathrm{sem}}\}.$$

For each selected semantic neighbor $v_j \in \mathcal{N}_{\mathrm{sem}}(v_i)$, CONSIST instantiates an implicit edge by prompting the LLM with the query, the two endpoint descriptions, and the provenance context:

$$\widehat{r}_q(v_i, v_j),\ a_q(v_i, v_j) = \mathrm{LLM}_{\mathrm{rel}}(q, c_{v_i}, c_{v_j}, \Gamma_{ij}),$$

where $\widehat{r}_q(v_i, v_j)$ is the query-conditioned relation label and $a_q(v_i, v_j) \in [0, 1]$ is the relation-instantiation confidence. The latent edge set proposed from v_i is

$$\mathcal{E}_t^{\text{latent}}(v_i) = \{(v_i, \widehat{r}_q(v_i, v_j), v_j, \text{sem}, s(v_i, v_j \mid q), a_q(v_i, v_j), \Gamma_{ij}) \mid v_j \in \mathcal{N}_{\text{sem}}(v_i),\ a_q(v_i, v_j) \geq \tau_{\text{rel}}\}.$$

The semantic candidate set at step t is

$$\mathcal{E}_t^{\text{latent}} = \bigcup_{v_i \in \mathcal{F}_t} \mathcal{E}_t^{\text{latent}}(v_i).$$

Unlike $\mathcal{E}_t^{\text{graph}}$, edges in $\mathcal{E}_t^{\text{latent}}$ are not assumed to pre-exist in E. They are query-conditioned candidates instantiated from supporting text and are passed to the same Nash-guided editing module as structural edges.

H.3 Candidate Edge Update

The newly proposed edges at iteration t are obtained by combining structural and semantic sources:

$$\mathcal{E}_t^{\text{new}} = \mathcal{E}_t^{\text{graph}} \cup \mathcal{E}_t^{\text{latent}}.$$

When $|\mathcal{E}_t^{\text{new}}|$ exceeds the expansion budget, we rank all structural and semantic edges in a unified candidate pool, select edges with the highest relevance scores:

$$\widetilde{\mathcal{E}}_t^{\text{new}} = \left\{ e \in \mathcal{E}_t^{\text{new}} \;\middle|\; \rho_q(e) \in \text{TopK}_{K_{\text{exp}}} \{\rho_q\} \right\}$$

and replace $\mathcal{E}_t^{\text{new}}$ with $\widetilde{\mathcal{E}}_t^{\text{new}}$ before constructing the expanded candidate graph. For structural edges, $\rho_q(e)$ measures how relevant the observed edge is to the query by considering both the supporting provenance of the edge and the textual description of its tail node. For semantic edges, we use

$$\rho_q(e) = \xi_1 s(v_i, v_j \mid q) + \xi_2 a_q(v_i, v_j) + \xi_3 \max_{\gamma \in \Gamma_{ij}} \cos(\phi(q), \phi(\gamma)),$$

where $e = (v_i, \widehat{r}_q(v_i, v_j), v_j, \text{sem}, s(v_i, v_j \mid q), a_q(v_i, v_j), \Gamma_{ij})$, $\xi_1, \xi_2, \xi_3 \geq 0$, and $\xi_1 + \xi_2 + \xi_3 = 1$. This unified ranking prevents the system from blindly favoring pre-existing graph edges over latent edges, or vice versa.

Accordingly, the expanded candidate graph before editing is updated as

$$\mathcal{V}_{t+1}^{(\text{can})} = \mathcal{V}_t \cup \{v_j \mid (v_i, r, v_j, \cdot) \in \mathcal{E}_t^{\text{new}},\ v_i \in \mathcal{F}_t\},$$

$$\mathcal{E}_{t+1}^{(\text{can})} = \mathcal{E}_t \cup \mathcal{E}_t^{\text{new}},$$

$$G_{t+1}^{(\text{can})} = (\mathcal{V}_{t+1}^{(\text{can})}, \mathcal{E}_{t+1}^{(\text{can})}).$$

The final decision of whether each proposed edge should be retained is made by the Nash-guided graph editing step. After editing, we denote the retained graph as

$$G_{t+1} = (\mathcal{V}_{t+1}, \mathcal{E}_{t+1}),$$

and update the next frontier from newly retained nodes:

$$\mathcal{F}_{t+1} = \{v_j \in \mathcal{V}_{t+1} \setminus \mathcal{V}_t \mid \exists (v_i, r, v_j, \cdot) \in \mathcal{E}_{t+1},\ v_i \in \mathcal{F}_t\}.$$

For clarity, `CONSIST` distinguishes observed structural edges from instantiated semantic edges when serializing the retained evidence graph for LLM-based scoring and answer generation. Structural edges are marked as `[STR]` and correspond to triples inherited from the explicit graph, while semantic edges are marked as `[SEM]` and correspond to query-conditioned latent relations instantiated from supporting text. For each semantic edge $(v_i, \widehat{r}_q(v_i, v_j), v_j)$, we additionally record its semantic score $s(v_i, v_j \mid q)$, relation-instantiation confidence $a_q(v_i, v_j)$, and provenance Γ_{ij}. This distinction prevents implicit semantic links from being conflated with original graph triples, and allows the utility scorer and final generator to identify whether a retained edge is an observed structural fact or a query-conditioned semantic bridge.

I Utility Scoring and Answer Generation

This section provides the prompts used by CONSIST for graph utility scoring and final answer generation. In our framework, the LLM is used only as a frozen verifier or generator. Specifically, the answer-support verifier estimates $LLM_{\text{sup}}(G, q)$, the pairwise conflict verifier estimates $LLM_{\text{pair}}(q, (e_i, e_j))$, and the final generator produces the answer from the terminal graph G_q^*. The minimality term does not require an LLM prompt, since it is computed directly from the number of retained edges.

I.1 Prompt for Answer-Support Scoring

The following prompt is used to score whether a candidate evidence graph G is sufficient for answering the query q. The scalar used as $LLM_{\text{sup}}(G, q)$ in Eq. (6) is `graph_verdict.support_score`. All other fields are used only for diagnostics and analysis.

Prompt for Answer-Support Scoring

```
System:
You are an expert verifier for graph-grounded question answering.
Your task is to judge whether the provided evidence graph G is sufficient to answer
     the query q.

Use ONLY the provided query, evidence graph, and edge inventory.
Do NOT use outside knowledge.
Do NOT rely on your parametric knowledge.
Do NOT infer missing facts beyond the graph.
Do NOT generate a final answer from your own knowledge.
Your judgment should focus on whether G contains a coherent, query-aligned, and
     graph-grounded reasoning path that is sufficient for answering q.

The goal is to score graph-level answer support, not to solve the question
     independently.
A graph is sufficient only if the answer can be derived from explicit nodes, edges,
     relations, or text spans in G without unsupported inference leaps.

User:
Query q:
{q}

Evidence graph G:
{G}

Edge inventory with IDs:
{EdgeListWithIDs}

Return EXACTLY one JSON object:
{
  "query_analysis": {
    "question_type": "<span / entity / relation / temporal / comparison / multi_hop /
     other>",
    "target_answer_variable": "<text>",
    "core_entities": ["<entity_1>", "<entity_2>", "..."],
    "required_reasoning_steps": ["<step_1>", "<step_2>", "..."]
  },
  "graph_evidence_coverage": {
    "contains_query_entities": true,
    "contains_bridge_entities": true,
    "contains_answer_anchor": true,
    "contains_complete_reasoning_path": true,
    "requires_outside_knowledge": false,
    "requires_unsupported_inference": false
  },
  "graph_verdict": {
    "answerable_from_graph_only": true,
```

```
    "support_score": 0.0,
    "sufficiency_score": 0.0,
    "path_coherence_score": 0.0,
    "missing_evidence_score": 0.0,
    "distractor_risk_score": 0.0,
    "overall_confidence": 0.0
  },
  "answer_grounding": {
    "answer_status": "<fully_supported / partially_supported / unsupported>",
    "answer_anchor_type": "<node / edge / span / none>",
    "answer_anchor_id_or_text": "<node_id, edge_id, exact graph span, or
     INSUFFICIENT_GRAPH_EVIDENCE>",
    "answer_anchor_is_verbatim_from_graph": true
  },
  "supporting_edges": [
    {
      "edge_id": "<edge_id>",
      "support_role": "<direct_answer / bridge / constraint / context / distractor>",
      "is_critical": true,
      "reason": "<brief explanation>"
    }
  ],
  "missing_information": [
    {
      "missing_step": "<text>",
      "why_missing": "<brief explanation>"
    }
  ],
  "final_explanation": "<brief explanation>"
}

Scoring rubric for support_score:

- [0.0, 0.2): almost no support.
  The graph may mention some query entities, but it does not contain a useful
     query-aligned reasoning path.

- [0.2, 0.4): weak support.
  The graph contains partial evidence, but one or more key entities, bridge links, or
     answer-supporting relations are missing.

- [0.4, 0.6): partial support.
  The graph suggests a plausible reasoning path, but answering q still requires
     unsupported inference, missing links, or knowledge outside G.

- [0.6, 0.8): strong support.
  The graph contains a mostly coherent answer path, but has minor incompleteness,
     weak grounding, or mild distracting evidence.

- [0.8, 1.0): near-complete support.
  The graph contains a direct or clearly connected evidence path sufficient to answer
     q using only G.

Additional score meanings:

- sufficiency_score:
  Measures whether G contains enough explicit evidence to answer q without outside
     knowledge.

- path_coherence_score:
  Measures whether the supporting edges form a coherent query-aligned reasoning chain
     rather than disconnected fragments.

- missing_evidence_score:
  Measures how much required evidence is missing.
  Higher means more required information is absent.
```

```
- distractor_risk_score:
  Measures how likely irrelevant, misleading, or off-path edges are to distract
     downstream reasoning.
  Redundancy alone should not strongly increase this score unless it makes the
     reasoning path ambiguous or misleading.

- overall_confidence:
  Measures confidence in this graph-level judgment, not confidence in an
     independently generated answer.

Scoring consistency rules:

- support_score should be high only when sufficiency_score and path_coherence_score
     are both high.
- support_score should be penalized when missing_evidence_score is high.
- support_score should be penalized when distractor_risk_score is high and the
     distractors interfere with the query-aligned path.
- A graph with high distractor_risk_score may still receive a moderate or high
     support_score if a complete and unambiguous supporting path is present.
- A graph with disconnected relevant facts but no bridge path should receive at most
     partial support.
- If answerable_from_graph_only is false, support_score should normally be below 0.6.
- If requires_outside_knowledge is true or requires_unsupported_inference is true,
     support_score should normally be below 0.6.
- If contains_complete_reasoning_path is false, support_score should normally be
     below 0.7.

Rules for answer grounding:

- Do NOT generate a free-form candidate answer from outside knowledge.
- If the answer is explicitly grounded in G, set answer_anchor_id_or_text to the
     corresponding node ID, edge ID, or exact text span from G.
- If the answer is not explicitly grounded in G, set answer_anchor_id_or_text to
     "INSUFFICIENT_GRAPH_EVIDENCE".
- answer_anchor_id_or_text must be copied from the provided graph or edge inventory
     whenever possible.
- Do not normalize, paraphrase, or complete the answer using outside knowledge.
- answer_anchor_is_verbatim_from_graph must be true only when the anchor is directly
     copied from G.

Decision rules:

1. Judge support only under q, not in the abstract.
2. Use ONLY the provided graph and edge inventory.
3. Do NOT reward unsupported inference leaps.
4. Do NOT penalize a graph merely because it contains redundant evidence; redundancy
     is mainly handled by the minimality term.
5. A bridge edge can be critical even if it does not contain the final answer.
6. Missing evidence is different from conflicting evidence.
7. If multiple answer paths exist, prefer the path that is most direct, complete,
     query-aligned, and graph-grounded.
8. If multiple answer paths exist but they are mutually incompatible or make the
     answer ambiguous, increase distractor_risk_score and reduce support_score.
9. If the graph contains relevant entities but lacks the relation needed to connect
     them, assign weak or partial support.
10. If the graph contains the final answer entity but lacks the reasoning path
     required by q, do not assign near-complete support.
11. The scalar used as LLM_sup(G,q) is "graph_verdict.support_score".

Output rules:

- Every score must be a real number in [0,1].
- Boolean fields must be true or false, not strings.
- "supporting_edges.edge_id" must refer only to the provided edge IDs.
```

```
- "answer_anchor_id_or_text" must refer only to content present in the provided
    graph, edge inventory, or "INSUFFICIENT_GRAPH_EVIDENCE".
- Output ONLY the JSON-format result.
- Do not include any other text or explanations.
```

I.2 Prompt for Conflict Scoring

Following prior methods that model evidence through relations or logical connections between graph units [5, 20], we compute consistency over edge pairs rather than isolated nodes, since query-conditioned conflicts usually arise from misleading reasoning paths instead of individual evidence units. The following prompt is used to score the degree of query-conditioned conflict between retained edges. For a given query–edge-pair input (q, e_i, e_j), the prompt estimates whether the two edges introduce query-conditioned knowledge conflict according to the three conflict sources defined: retrieval noise, counterfactual fact, and semantic ambiguity. The scalar used as $LLM_{\text{pair}}(q, (e_i, e_j))$ in Eq. (7) is `conflict_assessment.conflict_score`. A higher score indicates stronger query-conditioned conflict, which is subtracted in the consistency term of the overall graph utility.

In conflict-oriented benchmarks such as ConFiQA, the prompt is further required to evaluate faithfulness with respect to the provided query-specific context rather than the model's parametric knowledge. Therefore, an edge should not be penalized merely because it contradicts common world knowledge or the model's prior belief, as long as it is supported by the provided context and is useful for answering the query. This context-faithfulness rule prevents counterfactual but query-supported evidence from being incorrectly treated as conflict, and encourages the graph editing process to preserve evidence that is faithful to the given context.

Prompt for Pairwise Query-Conditioned Conflict Scoring

```
System:
You are an expert verifier for query-conditioned conflict detection in graph-based
    retrieval.
Your task is to judge whether two evidence edges introduce query-conditioned
    knowledge conflict under the current query q.

Use ONLY the provided query, evidence edges, and local graph context.
Do NOT use outside knowledge.
Do NOT rely on your parametric knowledge.
Do NOT override the provided context with common world knowledge.
In conflict-oriented datasets such as ConFiQA, the provided context is the
    authoritative source for answering the query, even if it contradicts common
    world knowledge.

Two edges are NOT conflicting merely because they are different.
Two edges are NOT conflicting merely because they are redundant; redundancy is
    handled by the minimality term.
Two edges are NOT conflicting merely because they contain counterfactual information.
A counterfactual edge should be treated as useful, not conflicting, if it is
    supported by the provided context and helps answer q.

Two edges conflict only if retaining both would make the evidence graph less
    reliable, less query-aligned, or less faithful to the provided context for
    answering q.

The conflict source must be judged according to the following three types:

S1 Retrieval noise:
The conflict is caused by irrelevant, unreliable, weakly grounded, spuriously
    retrieved, or incorrectly linked evidence.
This includes erroneous entity matching, wrong relation grounding, spurious retrieval
    results, or distracting edges that push the graph away from the query-aligned
    and context-supported reasoning path.
```

```
S2 Counterfactual fact:
The conflict is caused by competing evidence paths involving counterfactual or
     context-conflicting facts.
This applies when the provided graph contains a context-supported counterfactual path
     for q, but another edge preserves or reintroduces a competing unsupported
     factual/original/parametric path.
This also applies when an injected counterfactual edge is not supported by the
     provided context, is mis-grounded, or steers the graph toward a query-misaligned
     answer.
Do NOT assign a high counterfactual conflict score merely because an edge contradicts
     common world knowledge.
Assign high counterfactual conflict only when the conflict is visible from the
     provided edges or local graph context.

S3 Semantic ambiguity:
The conflict is caused by ambiguous entities, relations, mentions, or reasoning paths.
This applies when multiple plausible interpretations are mixed together, causing the
     graph to follow a query-misaligned path or to preserve competing answer paths
     that cannot jointly support q under the provided context.

User:
Query q:
{q}

Edge e_i:
{e_i_with_id_and_context}

Edge e_j:
{e_j_with_id_and_context}

Optional local graph context:
{local_graph_context}

Return EXACTLY one JSON object:
{
  "pair_identification": {
    "edge_i_id": "<edge_id>",
    "edge_j_id": "<edge_id>"
  },
  "query_conditioned_relevance": {
    "edge_i_relevant_to_q": true,
    "edge_j_relevant_to_q": true,
    "shared_query_dependency": true,
    "same_reasoning_slot": true,
    "same_answer_space": true,
    "context_supported_edge_i": true,
    "context_supported_edge_j": true,
    "context_faithful_pair": true
  },
  "conflict_sources": {
    "retrieval_noise_score": 0.0,
    "counterfactual_conflict_score": 0.0,
    "semantic_ambiguity_score": 0.0
  },
  "conflict_assessment": {
    "can_coexist": true,
    "dominant_conflict_source": "none",
    "conflict_score": 0.0,
    "keep_both_risk": 0.0,
    "brief_reason": "<brief explanation>"
  }
}

Scoring rules for context faithfulness:
```

```
- Judge whether each edge is supported by the provided context and whether it helps
    answer q.
- In ConFiQA-style cases, prefer the provided context over common world knowledge or
    parametric knowledge.
- If an edge is counterfactual but explicitly supported by the provided context, do
    not assign a high conflict score for that reason alone.
- If two edges jointly support the context-faithful answer to q, they should receive
    low conflict scores even if they contradict common factual knowledge.
- If one edge supports the provided context while the other edge supports a competing
    answer path that is explicitly present in the provided edges or local graph
    context but is unsupported by the authoritative context for q, assign a high
    conflict score.
- Do not infer an unsupported original factual answer from outside knowledge. Such a
    competing path must be visible in the provided edges or local graph context.
- If both edges are unsupported by the provided context and distract from q, treat
    them as retrieval noise, counterfactual conflict, or semantic ambiguity
    depending on the source of the error.

Scoring rules for conflict sources:

- retrieval_noise_score:
  Score high when one or both edges are irrelevant, unreliable, incorrectly grounded,
     spuriously retrieved, or linked to the wrong entity or relation under q.
  Score high when retaining the pair would introduce a distracting branch that
     weakens the query-aligned and context-supported reasoning path.
  Score high when an edge is unsupported by the provided context and encourages an
     answer that is not faithful to that context.
  Score low when both edges are well grounded in the provided context and jointly
     useful for answering q.

- counterfactual_conflict_score:
  Score high when one edge supports the context-faithful counterfactual answer while
     the other edge supports a competing non-context-faithful answer path.
  Score high when an injected or counterfactual edge is not supported by the provided
     context and steers the graph toward a misleading answer.
  Score high when retaining both edges would mix context-supported counterfactual
     evidence with unsupported original, factual, or parametric evidence, creating
     competing answer paths under q.
  Score low when the counterfactual edge is explicitly supported by the provided
     context and helps answer q.
  Score low when both edges support the same context-faithful counterfactual answer.
  Do not use common world knowledge to decide whether an edge is factual or
     counterfactual; rely only on the provided context and local graph context.

- semantic_ambiguity_score:
  Score high when the two edges reflect different plausible interpretations of the
     same query entity, relation, mention, or reasoning slot.
  Score high when retaining both edges would preserve competing reasoning paths or
     answer candidates caused by ambiguity.
  Score high when ambiguity causes the graph to move away from the context-supported
     answer to q.
  Score low when the two edges support complementary steps of the same query-aligned
     and context-faithful reasoning path.

Score calibration:
- 0.0 means no conflict.
- 0.1 to 0.3 means weak or unlikely conflict.
- 0.4 to 0.6 means moderate conflict or uncertainty.
- 0.7 to 0.9 means strong conflict.
- 1.0 means severe conflict that clearly harms query-faithful reasoning.

Combine the three source scores as follows:
Let
r = retrieval_noise_score,
c = counterfactual_conflict_score,
s = semantic_ambiguity_score.
```

```
Set
conflict_score = 1 - (1 - r) * (1 - c) * (1 - s).

Scoring rules for keep_both_risk:
- Low when the two edges are jointly useful or complementary.
- Low when both edges support the provided context and help answer q.
- High when retaining both increases the risk of preserving misleading, unsupported,
    ambiguous, or answer-disruptive evidence.
- High when retaining both increases the risk of answering from an unsupported
    competing path instead of the provided context.
- keep_both_risk should usually be close to conflict_score, but may be slightly
    higher when the conflicting pair is central to the query reasoning path.

Decision rules:
1. Judge conflict only under q, not in the abstract.
2. Treat the provided context as authoritative when judging context-faithfulness.
3. Do not use outside knowledge or parametric knowledge to override the provided
    context.
4. If either edge is irrelevant to q, assign low conflict unless it actively misleads
    the reasoning process.
5. If the two edges support different but complementary steps of a valid reasoning
    chain, assign low conflict.
6. If both edges support the context-faithful answer, assign low conflict even if the
    answer is counterfactual.
7. If one edge supports the context-faithful answer and the other supports an
    unsupported competing answer path that is visible in the provided edges or local
    graph context, assign high conflict.
8. Missing evidence or incompleteness alone is not conflict.
9. Redundancy alone is not conflict; redundancy is handled by the minimality term.
10. If no source score is above 0.5, set "dominant_conflict_source" to "none".
11. Otherwise, set "dominant_conflict_source" to the source with the highest score
    among retrieval_noise_score, counterfactual_conflict_score, and
    semantic_ambiguity_score.
12. The scalar used as LLM_pair(q,(e_i,e_j)) is "conflict_assessment.conflict_score".

Output rules:
- Every score must be a real number in [0,1].
- Boolean fields must be true or false, not strings.
- "dominant_conflict_source" must be one of: "retrieval_noise",
    "counterfactual_conflict", "semantic_ambiguity", "none".
- "conflict_assessment.conflict_score" must be computed from the three source scores
    using the formula above.
- Output ONLY the JSON-format result.
- Do not include any other text or explanations.
```

I.3 Evidence overlap across backbone LLMs

To assess how different backbone LLMs affect the resulting evidence graphs, Table 12 reports the overlap between graphs produced using GPT-4o and other LLMs on 2Wiki. Overall, the resulting graphs share most nodes, with even higher structural overlap for single-hop queries.

Table 12: Evidence overlap (%) between GPT-4o and other LLMs for 2Wiki, where numbers in brackets show the single-hop queries.

	Nodes	Edges
DeepSeek-V3	89.5 (95.7)	80.2 (91.3)
Qwen2-7B	84.7 (92.6)	73.9 (88.4)
LLaMA2-7B	82.2 (90.9)	66.4 (84.7)

J Time Complexity Analysis

This section analyzes the time complexity of CONSIST and existing baselines, including GRAPHRAG [5], HIPPORAG [7], RELINK [11], HOPRAG [20], and TOG-3 [42].

For a query q, let $G_t = (\mathcal{V}_t, \mathcal{E}_t)$ denote the evidence graph at iteration t, and let $\mathcal{F}_t$ denote its frontier. Existing methods typically employ a budget hyperparameter b to limit the number of newly retrieved or expanded evidence units at each iteration. Accordingly, the number of newly added candidate edges is bounded by

$$\left| |\mathcal{E}_{t+1}^{(\mathrm{can})}| - |\mathcal{E}_t| \right| \le b,$$

where $|\mathcal{E}|$ denotes the number of edges. In addition, let C_{exp} denote the cost of expanding one candidate edge, such as neighbour retrieval or evidence linking. Let C_{LLM} denote the cost of one LLM-based operation over a selected candidate unit, including scoring, reranking, reasoning, reflection, or utility evaluation.

GRAPHRAG constructs the entity graph offline. During inference, it retrieves at most b relevant evidence units from the pre-constructed graph. The selected evidence is then processed by the LLM for summarization, aggregation, and answer generation. Thus, the inference-time complexity is

$$\boxed{\text{GRAPHRAG}: \quad \mathcal{O}(bC_{\mathrm{LLM}})\,.}$$

HIPPORAG relies on a pre-built memory graph. During inference, it performs graph search over this static graph to retrieve relevant evidence. Given the budget b, graph search contributes $\mathcal{O}(bC_{\mathrm{exp}})$, while final answer generation contributes $\mathcal{O}(C_{\mathrm{LLM}})$. Thus, its complexity is

$$\boxed{\text{HIPPORAG}: \quad \mathcal{O}(bC_{\mathrm{exp}} + C_{\mathrm{LLM}})\,.}$$

RELINK dynamically constructs missing links by reasoning over a latent relation pool. During inference, it jointly uses explicit graph structure and latent embedding-based relations to instantiate and rank query-useful candidates. Therefore, with the budget b, candidate construction costs $\mathcal{O}(bC_{\mathrm{exp}})$, LLM-based candidate evaluation costs $\mathcal{O}(bC_{\mathrm{LLM}})$, and candidate ranking costs $\mathcal{O}(b \log b)$. Therefore, its inference-time complexity is

$$\boxed{\text{RELINK}: \quad \mathcal{O}(b(C_{\mathrm{exp}} + C_{\mathrm{LLM}}) + b\log b)\,.}$$

HOPRAG first retrieves seed passages and then explores multi-hop logical neighbours guided by pseudo-queries. With budget b, retrieving candidate edges costs $\mathcal{O}(bC_{\mathrm{exp}})$. Since the method evaluates pairwise frontier–neighbour interactions during multi-hop reasoning, the LLM-guided reasoning cost can grow as $\mathcal{O}(b^2 C_{\mathrm{LLM}})$. Thus, its inference-time complexity is

$$\boxed{\text{HOPRAG}: \quad \mathcal{O}\big(bC_{\mathrm{exp}} + b^2 C_{\mathrm{LLM}}\big)\,.}$$

TOG-3 employs multiple agents to iteratively perform retrieval, response generation, sufficiency reflection, query evolution, and subgraph evolution. If the process runs for at most R rounds and each round considers at most b candidates, candidate retrieval contributes $\mathcal{O}(RbC_{\mathrm{exp}})$, while multi-agent LLM operations contribute $\mathcal{O}(RbC_{\mathrm{LLM}})$. Thus, its time complexity becomes

$$\boxed{\text{TOG-3}: \quad \mathcal{O}(Rb(C_{\mathrm{exp}} + C_{\mathrm{LLM}}))\,.}$$

Finally, CONSIST follows an exploration–editing procedure. First, it expands at most b new structural or semantic edges from the current frontier, so the expansion cost is $\mathcal{O}(bC_{\mathrm{exp}})$. Second, Gumbel-based editing is applied to the whole candidate edge set $\mathcal{E}_{t+1}^{(\mathrm{can})}$. Since $\left|\mathcal{E}_{t+1}^{(\mathrm{can})}\right| \le |\mathcal{E}_t| + b$, the editing cost is $\mathcal{O}\left(\left|\mathcal{E}_{t+1}^{(\mathrm{can})}\right|\right) = \mathcal{O}(|\mathcal{E}_t| + b)$. Third, for H look-ahead steps, the rollout cost is $\mathcal{O}(HbC_{\mathrm{exp}})$. Finally, utility evaluation over the current and future graphs requires $\mathcal{O}(2C_{\mathrm{LLM}})$. Overall, the complexity of CONSIST is

$$\boxed{\texttt{CONSIST}: \quad \mathcal{O}(bC_{\mathrm{exp}} + (|\mathcal{E}_t| + b) + HbC_{\mathrm{exp}} + 2C_{\mathrm{LLM}})\,.}$$

Since each iteration introduces at most b new edges, we have $|\mathcal{E}_t| \leq |\mathcal{E}_0| + tb$. When the initial graph size is treated as constant, this gives $|\mathcal{E}_t| = \mathcal{O}(tb)$, and the complexity can be simplied as

$$\boxed{\texttt{CONSIST}: \quad \mathcal{O}((1+H)bC_{\text{exp}} + tb + C_{\text{LLM}})\,.}$$

Thus, the additional editing cost grows linearly with the accumulated expansion budget, while the expensive expansion and LLM-based operations remain comparable to existing GraphRAG methods.

Table 13: Comparison of computational cost (time in second) on three multi-hop QA datasets.

Method	HotpotQA	2Wiki	MuSiQue	Avg.
GRAPHRAG	8.91	7.45	9.37	8.58
HIPPORAG	4.85	4.12	4.92	4.63
RELINK	14.23	12.54	11.09	12.62
HOPRAG	18.25	16.73	17.59	17.52
TOG-3	17.13	15.07	13.34	15.18
CONSIST	14.45	12.70	11.25	12.80

Table 13 empirically validates the above complexity analysis. As observed, CONSIST achieves a runtime comparable to RELINK and remains more efficient than HOPRAG and TOG-3 on three multi-hop QA datasets. This result confirms that the additional Nash-guided editing step introduces only moderate overhead while preserving practical inference efficiency.

To evaluate the trade-off between algorithmic cost and answer quality, Table 14 reports results under different expansion budgets $b \in \{3, 6, 9\}$, rollout horizons $H \in \{2, 3\}$, and stopping thresholds $\tau_{\text{stop}} \in \{0.7, 0.9\}$ on 2Wiki using GPT-4o.

Table 14: Cost-quality trade-offs on 2Wiki using GPT-4o.

b	H	τ_{stop}	Time(s)	LLMCalls	EdgeEdits	EM
3	2	0.7	9.2	6.1	4.2	66.5
3	2	0.9	9.8	7.0	4.7	66.6
3	3	0.7	11.7	8.6	5.1	64.0
3	3	0.9	12.2	8.8	5.4	64.1
6	**2**	**0.7**	**12.7**	**8.3**	**4.9**	**67.2**
6	2	0.9	13.4	8.5	5.1	67.2
6	3	0.7	15.3	10.3	6.7	64.2
6	3	0.9	15.8	10.8	6.9	64.1
9	2	0.7	14.6	10.4	6.1	67.1
9	2	0.9	15.4	10.9	6.4	66.9
9	3	0.7	17.9	12.3	7.3	65.4
9	3	0.9	18.5	12.6	7.4	65.6

Consistent with the sensitivity analysis and the runtime comparison above, we find that $b = 6$, $H = 2$, and $\tau_{\text{stop}} = 0.7$ provide the best cost-quality trade-off: a smaller b may miss query-critical links, whereas a larger b introduces more candidates that are later removed by Nash editing. Similarly, a longer H accumulates more noisy evidence, and a higher stopping threshold triggers unnecessary edits to the graph. Consequently, larger settings increase runtime, LLM calls, and edge edits without consistent improvements in answer quality.

K Discussion on Nash-guided optimization

Standard Nash Bargaining Problem (NBP) requires axioms of scale invariance, symmetry, efficiency, and independence of irrelevant alternatives to be satisfied [27, 28]. Recall that our utility function is calculated as

$$u(G,q) = \mathbb{E}\Big(\mathrm{LLM}_{\mathrm{sup}}(G,q) + \lambda\Big(1 - \mathrm{Avg}_{(e_i,e_j)\in\mathcal{E}(G)}\,\mathrm{LLM}_{\mathrm{pair}}(q,(e_i,e_j))\Big) + (1-\lambda)\Big(|\mathcal{E}(G')|/|\mathcal{E}(G)|\Big).\Big) \tag{16}$$

We first prove it is continuous *w.r.t* $\boldsymbol{\theta}$, *i.e.*the Gumbel softmax model parameter.

Proposition 1. *Let $\mathcal{G}$ be a finite set of graphs and $h : \mathcal{G} \mapsto \mathbb{R}$ be any function that maps a graph to a real number. For each parameter vector $\boldsymbol{\theta} \subset \mathbb{R}^m$, let g be a random graph generated by Gumbel-softmax edge-retention probabilities. Define*

$$u(\boldsymbol{\theta}) = \mathbb{E}[h(g) \mid \boldsymbol{\theta}].$$

Assume the temperature satisfies $\Delta > 0$. Then $u : A \to \mathbb{R}$ is continuous in $\boldsymbol{\theta}$, so are u_{cur} and u_{fut}.

Proof. Let the candidate edge set be finite: $\mathcal{E} = \{e_1, \ldots, e_m\}$. Every graph $g \in \mathcal{G}$ corresponds to a binary edge-retention vector $\boldsymbol{x} = (x_1, \ldots, x_m) \in \{0,1\}^m$, where 0 for deletion and 1 otherwise. Thus $\mathcal{G}$ is finite, with at most 2^m possible graphs.

The Gumbel-softmax parameterization defines edge-retention probabilities

$$p_i(\boldsymbol{\theta}, \epsilon) = \sigma\left(\frac{\log\epsilon - \log(1-\epsilon) + \boldsymbol{\theta}}{\Delta}\right),$$

where $\epsilon \sim \mathcal{U}(0,1)$. Refer to Eq. (10) for details. For each fixed ϵ, the map $\boldsymbol{\theta} \mapsto p_i(\boldsymbol{\theta}, \epsilon)$ is continuous, because it is a quotient of continuous functions and the denominator is strictly positive.

Now consider a particular graph g_x represented by $\boldsymbol{x} \in \{0,1\}^m$. Under independent Bernoulli edge-retention decisions conditional on $\boldsymbol{\theta}$, its probability is

$$P(g = g_x \mid \boldsymbol{\theta}) = \prod_{i=1}^{m} p_i(\boldsymbol{\theta})^{x_i}(1 - p_i(\boldsymbol{\theta}))^{1-x_i}.$$

Each $p_i(\boldsymbol{\theta})$ is continuous. Therefore each factor $p_i(\boldsymbol{\theta})^{x_i}(1 - p_i(\boldsymbol{\theta}))^{1-x_i}$ is continuous. A finite product of continuous functions is continuous, hence $\boldsymbol{\theta} \mapsto P(g = g_x \mid \boldsymbol{\theta})$ is continuous.

Now write the expected utility as a finite sum over all possible graphs:

$$u(\boldsymbol{\theta}) = \mathbb{E}[h(g) \mid \boldsymbol{\theta}] = \sum_{g\in\mathcal{G}} h(g)P(g \mid \boldsymbol{\theta}). \tag{17}$$

Since $\mathcal{G}$ is finite, this is a finite sum. Each coefficient $h(g)$ is a fixed real number, and each probability term $P(g \mid \boldsymbol{\theta})$ is continuous in $\boldsymbol{\theta}$. Therefore $u(\boldsymbol{\theta})$ is a finite linear combination of continuous functions and $u(\boldsymbol{\theta})$ is continuous in $\boldsymbol{\theta}$.

An immediate conclusion is that $u_{\mathrm{cur}} = \mathbb{E}[U(G_{t+1}^{(edit)})]$ is continuous. Since $G_{t+1+H}^{(edit)}$ is the rolling out of graph expansion after $G_{t+1}^{(edit)}$, it can be regarded as a mapping f between finite graphs $\mathcal{G}$ on g, leading to replacing $h(g)$ by $h(f(g))$ in Eq. (17). Therefore $u_{\mathrm{fut}} = \mathbb{E}[U(G_{t+1+H}^{(edit)})]$ is also continuous in $\boldsymbol{\theta}$. □

The proof shows that no constraint to the graph mapping $h(g)$ whose role is a scaling factor in the utility computation. This circumvents the problem of the composition from $\boldsymbol{\theta}$ to $\mathbb{R}$ (*i.e.* $h(g)|\theta$) being discontinuous, *yielding a continuous surrogate objective that is amenable to gradient-based optimization*.

In addition, we need to point out that $u(\boldsymbol{\theta})$ may be non-convex due to the Gumbel-Softmax/Sigmoid relaxation. However, we do not rely on the uniqueness guarantee of the classical Nash Bargaining Problem. Instead, the proposed solution (see Eq. (14)) uses the Nash product as a Nash-style

coordination criterion for comparing feasible relaxed edits. By maximizing the joint improvement over the disagreement point, the objective discourages one-sided edits that improve only the current graph or only its look-ahead rollout. As a result, the selected parameter $\boldsymbol{\theta}$ balances the immediate quality of G_{t+1} and the future stability of G_{t+1+H}. This is sufficient for our setting, since `CONSIST` only requires a principled edit-selection rule rather than a closed-form unique bargaining solution.

L Limitation and Future Work

CONSIST is designed as an evidence-regulation framework for GraphRAG, and our experiments focus on multi-hop QA and conflict-oriented benchmarks. While these settings capture typical graph reasoning and conflict scenarios, the framework relies on manually specified hyperparameters, which may limit adaptability across diverse tasks; developing more data-driven and adaptive tuning mechanisms is an important direction. In addition, although we adopt a unified utility for sufficiency, consistency, and minimality, more task-specific designs could further improve performance, as shown in the ablation study. Moreover, the relaxed graph-editing objective is generally non-convex, classical axiomatic guarantees such as uniqueness or independence of irrelevant alternatives may not hold in full. Instead, our use of the Nash product is therefore not intended as a strict axiomatic instantiation of Nash problem, but as a principled selection criterion for balancing competing graph-editing utilities. As such, future work may also explore more fine-grained evidence regulation and conflict modeling to meet axiomatic guarantees. Finally, integrating minimal sufficient graph learning with interactive or agentic retrieval is a promising direction, enabling GraphRAG systems to actively and accurately retrieve evidence under uncertainty.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We clearly state the background, motivation, and main contributions of CONSIST in the abstract and introduction, including the GraphRAG evidence-control problem, the conflict-aware minimal sufficient graph formulation, the Nash-guided optimization framework, and the corresponding empirical evaluation, all of which are aligned with the core sections of the paper.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: A detailed discussion of the limitations and future directions is provided in Appendix L.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: The assumptions are formally specified in Section 3.4, with the full proof provided in Appendix K.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: The overall experimental setup is presented in Section 4. Dataset statistics are reported in Appendix D. Implementation details for seed initialization and graph expansion are provided in Appendices G and H, respectively. Prompt templates for support scoring, conflict scoring, and final answer generation are provided in Appendix I.

Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: The code is available at `https://anonymous.4open.science/r/CONSIST-DF0F`.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The experimental setup is described in the main paper, with full details provided in the appendix.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We report results averaged over 10 runs in the main experimental tables, with standard deviations explicitly provided in the corresponding "Std." rows. In addition, the ablation and hyperparameter sensitivity analyses are reported in Appendix E to show that the observed improvements are stable across different design choices and parameter settings.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The paper provides the required compute-resource information in Appendix J.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: This work adheres to the NeurIPS Code of Ethics in all relevant aspects.

   Guidelines:

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

10. **Broader impacts**

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

Answer: [Yes]

Justification: We discuss broader impacts in Section 5.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: Our work does not release any high-risk models or datasets. The method is evaluated on existing public benchmarks, with no known misuse-sensitive data or model artifacts introduced.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: All third-party assets used in this paper, including datasets, baselines, and models, are properly cited, and their licenses and terms of use are explicitly acknowledged.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.

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

13. **New assets**

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

    Answer: [N/A]

    Justification: This work does not introduce new assets. All experiments are conducted on existing publicly available datasets and models, which are properly cited and used in accordance with their respective licenses.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This work does not include crowdsourcing activities or experiments involving human subjects.

    Guidelines:

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

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

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

Answer: [N/A]

Justification: This work does not involve human subjects.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [Yes]

Justification: he paper describes the usage of LLMs in the core method, with detailed explanations provided in Appendice G and I.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.