# DCTR: Dual-Constraint Subgraph Optimization for Knowledge Graph-based Retrieval-Augmented Generation

**Yukun Cao[1*], Zirui Xu[1*], Dongyang Li[1†], Zhihao Guo[2], Luobin Huang[1], Lisheng Wang[1]**

[1]Shanghai University of Electric Power, China
[2]University of Technology Sydney, Australia
caoyukun@shiep.edu.cn, y24208169@mail.shiep.edu.cn, dongyangli.ldy@shiep.edu.cn

## Abstract

Knowledge Graph (KG)-based Retrieval-Augmented Generation (RAG) shifts the contents of retrieval from narrative text to a relational knowledge network, empowering large language models (LLMs) to harness structured relationships between entities. However, conventional KG-RAG approaches are resource-intensive, requiring either query decomposition with multiple LLM rounds or parameterized static knowledge injection to update the model. Although subgraph reasoning aims to address these issues, most current methods are based on heuristic shortest path and multi-hop graph traversal algorithms. The retrieved subgraphs suffer from incompleteness and semantic drift, and neglect the interaction between subgraph and LLMs in terms of fine-grained structural semantics. We propose a dual-constraint subgraph optimization for KG-RAG (DCTR). It improves subgraph retrieval and generates high-quality subgraphs with structural integrity and information salience for LLMs. Specifically, it formulates subgraph generation as a two-stage graph-theoretic constrained optimization problem to create compact and complete pseudo-labels. Since these pseudo-labels are discrete, a smooth approximation is employed to convert them into a differentiable representation, thereby optimizing the retriever to highlight key information while extracting subgraphs. On two benchmark datasets, DCTR significantly enhances subgraph quality, achieving state-of-the-art performance in LLM reasoning.

## Introduction

Retrieval-augmented generation (RAG) (Quinn et al. 2025; Lewis et al. 2020) improves the reasoning ability of large language models (LLMs) by retrieving relevant context from external knowledge bases. It is widely applied to various tasks, including open-domain question answering (Cahoon et al. 2025; Kim and Lee 2024), dialogue systems (Zhang et al. 2025a; Wang et al. 2024), and multi-hop reasoning (Liu et al. 2025; Zhang et al. 2025b). Among them, text-RAG retrieves unstructured documents to support generation, but may be hindered by the lack of explicit structure required for complex reasoning (Guo et al. 2025; Zhu et al. 2025).

Recent research focuses on knowledge graph-based RAG (KG-RAG) (Edge et al. 2025; Li, Miao, and Li 2025; Mavromatis and Karypis 2024), which leverages structured knowledge through distinct approaches. One paradigm is explicit reasoning-based retrieval. This method employs an LLMs to decompose complex queries into subqueries, which are then executed on the knowledge graph to retrieve relevant entities for final reasoning (Jiang et al. 2024; Liu et al. 2024a; Jin et al. 2024). However, this iterative use of LLMs for retrieval reasoning leads to a lot of computational and time overhead (Tang et al. 2025; Li, Miao, and Li 2025). An alternative strategy involves embedding the knowledge graph directly into the model via parameterized fine-tuning. While this can reduce inference latency, it sacrifices dynamic information access. Whenever the knowledge base changes, the model must be costly retrained, which fundamentally impairs its generalization ability and real-time accuracy (LUO et al. 2024; Mavromatis and Karypis 2024).

To transcend the limitations of the decomposition and parameterization paradigms, a promising approach involves using LLMs to reason about subgraphs. It can be divided into two categories. (1) One category involves multi-hop expansion methods such as breadth-first graph traversal after anchoring entities using pre-trained models (Guo et al. 2024; He et al. 2024; LUO et al. 2024), which tend to retrieve semantically redundant subgraphs. (2) The other category is lightweight model retrieval based on pseudo-label supervision (Li, Miao, and Li 2025; Mavromatis and Karypis 2024), but it uses heuristic supervision such as shortest path, which tends to prioritize structural compactness, thereby sacrificing semantically rich subgraphs. Secondly, neither approach accounts for the inherent positional bias of LLMs (Liu et al. 2024c; Zhang et al. 2024a,b). We have analyzed the underlying causes and found that it's crucial not only to improve subgraph quality but also to optimize the subgraph's fine-grained structure to make it more accessible to LLMs.

To address these limitations, we propose DCTR, a dual-constraint subgraph optimization for KG-RAG. At its core, DCTR optimizes subgraph retrieval by focusing on the structural integrity and information salience of the subgraphs. The internal mechanism comprises a dual optimization process. First, it employs a dual-constrained flux graph retrieval (DCFGR) module. This module algorithmically conceptualizes subgraphs as network flows through maximum flow retrieval and a dual-flux path extractor. By simultaneously considering the highest reliability and den-

*These authors contributed equally.
†Corresponding author.

sity, it selects high-quality subgraphs that are both structurally compact and semantically rich as pseudo-labels for supervised optimization. Second, it introduces a differentiable subgraph ranking supervision (DSRS) mechanism.Recognizing that discrete, combinatorial metrics for ranking are incompatible with gradient-based optimization, DSRS introduces a smooth, differentiable surrogate for pairwise ranking agreement. By replacing the non-differentiable sign function with a continuous approximation, our approach enables the direct optimization of the retriever's ranking function. This supervision aligns the model's attention with the weights of the pseudo-labels, teaching it not only what to retrieve but also how to highlight the saliency of subgraph content. Our method demonstrates stable and consistent superiority over multiple baseline models.

Our main contributions of this paper include:

- We propose the DCTR, which bridges the gap between high-quality subgraph retrieval and subgraph structure optimization through a unified training mechanism.
- To extract compact and complete subgraphs, we design DCFGR, which uses maximum flow retrieval and dual-flux path extractor to generate pseudo-labels.
- A DSRS mechanism is designed to guide the retriever to explicitly retrieve important content in the subgraph.
- Extensive experiments show that DCTR consistently outperforms existing baselines, while ablation studies confirm the crucial role of each of its individual components.

## Related Work

The convergence of KGs with LLMs (Edge et al. 2025; Li, Miao, and Li 2025) has evolved through a series of methodological paradigms, each defining a unique pipeline for knowledge access and exploitation. Initial explorations focused on using LLMs as central reasoning engines, deconstructing complex queries into a sequence of subqueries. Then, information was provided to LLMs to answer by querying the knowledge base via SPARQL (Jiang et al. 2024; Liu et al. 2024a; Jin et al. 2024).

To improve efficiency and reduce the latency inherent in this iterative process, a parallel research stream has emerged that focuses on parameterizing knowledge. This process aims to embed factual knowledge directly into the model's parameters, either through comprehensive fine-tuning (LUO et al. 2024; Mavromatis and Karypis 2024) or through more targeted lightweight adapters (He et al. 2024). This parameterizes knowledge, making it accessible through the model's internal representations rather than explicit external lookups.

These approaches catalyze a shift toward subgraph reasoning, a process that aims to provide targeted knowledge context for LLMs. Under this paradigm, subgraph retrieval is roughly divided into two main directions. The first, often symbolic in nature, is algorithmic subgraph construction. This process typically begins by identifying key entities and then applies graph traversal techniques, such as multi-hop expansion (Guo et al. 2024; Ji et al. 2024), to construct a subgraph. Evolving in parallel is supervised subgraph retrieval, which frames the task as a learning problem. Here, a retriever is trained to learn a direct mapping from a natural language query to its subgraph, guided by a pseudo-labeled dataset typically generated by heuristics such as shortest path (Li, Miao, and Li 2025; Mavromatis and Karypis 2024).

Although existing methods have made progress in the content selection problem, the subgraphs they generate often suffer from incompleteness and semantic drift. In contrast, our DCTR is designed to generate subgraphs that possess both structural integrity and information salience by effectively training the retriever. It not only captures necessary information but also considers key reasoning elements to generate high-quality subgraphs.

## Methodology

An overview of DCTR is depicted in Figure 1. As introduced, DCTR operates through a dual-optimization process. The first component, DCFGR, generates high-quality pseudo-labels by balancing structural compactness and semantic richness of subgraphs. The second component, DSRS, then uses these pseudo-labels to train a retriever through differentiable ranking optimization, ensuring that the final subgraph is easier for LLM to understand. The final subgraph is formatted into a prompt, detailed in the appendix, which is used by the LLM to generate the answer.

### Preliminaries

**KG.** A KG is formally represented as a set of triples, $\mathcal{G} = \{(h, r, t) \mid h, t \in \mathcal{E},\ r \in \mathcal{R}\}$, where $\mathcal{E}$ is a set of entities and $\mathcal{R}$ is a set of relations. Each triple, denoted as $\zeta = (h, r, t)$, asserts that a head entity $h$ is connected to a tail entity $t$ via a directed relation $r$. Our method operates on an induced weighted graph $\mathcal{H}$. This graph augments the KG by assigning a semantic weight to each triple. Formally, $\mathcal{H} = \{(\zeta, w(\zeta)) \mid \zeta \in \mathcal{G},\ w(\zeta) \in \mathcal{W}\}$, where $w(\zeta)$ is the weight associated with triple $\zeta$. The weight $w(\zeta_{\text{cand}})$ for a candidate triple $\zeta_{\text{cand}}$ is dynamically computed by a contextual scoring function, $\mathcal{F}_{\text{LLM}}$: $w(\zeta_{\text{cand}}) = \mathcal{F}_{\text{LLM}}(q,\ \mathcal{P}_{\text{cur}},\ \zeta_{\text{cand}})$, where $\mathcal{F}_{\text{LLM}}$ is an LLM-based function that evaluates semantic relevance, $q$ is user's input question, $\mathcal{P}_{\text{cur}} = [\zeta_1, \zeta_2, \ldots, \zeta_t]$ is current reasoning path, defined as a sequence of previously selected triples, and $\zeta_{\text{cand}}$ is candidate triple being scored. The resulting score $w(\zeta_{\text{cand}})$ represents the information volume of adding $\zeta_{\text{cand}}$ to the path $\mathcal{P}_{\text{cur}}$ in the context of the query. For large KGs, we use a greedy search, weighting only edges on question-answer paths to reduce costs.

**KG-RAG.** Given a user query $q$, the KG-RAG paradigm aims to first retrieve a query-relevant subgraph $\mathcal{H}_q \subseteq \mathcal{H}$. This targeted subgraph then serves as the contextual knowledge base for an LLM, which reasons over $\mathcal{H}_q$ to generate the final response $ans$. In our work, the retrieval of $\mathcal{H}_q$ is framed as a learning problem, rather than relying on heuristics. We employ a retriever model to perform this task. This model is trained in a supervised manner, using high-quality subgraphs as pseudo-labels. The generation of these crucial pseudo-labels will be detailed in the subsequent sections.

### Dual-Constrained Flux Graph Retrieval (DCFGR)

The DCFGR module generates high-quality pseudo-labels through a two-stage retrieval process designed to produce a subgraph that is both globally relevant and locally refined.

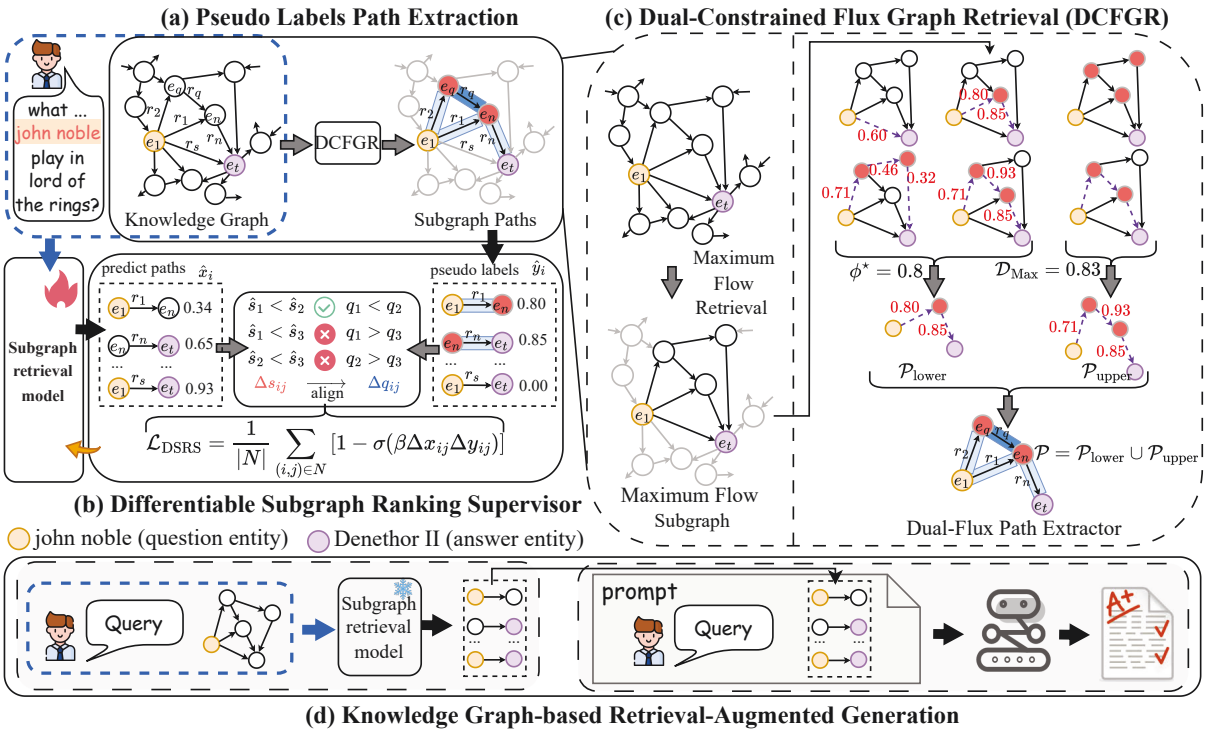


Figure 1: DCTR mainly consists of two stages: subgraph extraction by DCFGR and supervised optimization by DSRS. The actual reasoning path may present a more complex structure than the simple path.

**Maximum Flow Retrieval** First, we identify a globally relevant subgraph $\mathcal{H}_m$, by formulating the extraction task as a maximum flow problem. We construct an information flow network where each triple $\zeta$ in the knowledge graph $\mathcal{H}$ acts as a directed edge with a capacity equal to its weight $w(\zeta)$. The objective is to maximize the total semantic flow, $|f|$, from a designated source node $e_q$ (the query entity) to a sink node $e_a$ (the answer entity). To formalize this, let $f(\zeta)$ denote the flow through a given triple $\zeta$. We define the set of triples originating from $e_q$ as $Z_{\text{out}}(e_q) = \{\zeta \mid h(\zeta) = e_q\}$, and the set of triples terminating at $e_q$ as $Z_{\text{in}}(e_q) = \{\zeta \mid t(\zeta) = e_q\}$, where $h(\zeta)$ and $t(\zeta)$ represent the head and tail entities of $\zeta$, respectively. The objective is then to maximize the total net flow from the source:

$$\text{Maximize } |f| = \sum_{\zeta \in Z_{\text{out}}(e_q)} f(\zeta) - \sum_{\zeta \in Z_{\text{in}}(e_q)} f(\zeta). \quad (1)$$

The detailed algorithmic process for solving this maximum flow problem, including constraints on flow conservation and skew symmetry, is provided in appendix. Unlike local, greedy pathfinding methods (Hou et al. 2021; Zhu et al. 2025), our maximum flow formulation offers a global perspective. It evaluates all potential semantic routes simultaneously, maximizing total information throughput while inherently mitigating bias toward redundant or narrow paths. From an information-theoretic viewpoint, the solution yields a globally saturated subgraph $\mathcal{H}_m$. This subgraph represents a state where no additional semantic flow from $e_q$ to $e_a$ is possible without violating the capacity constraints of the network. Therefore, $\mathcal{H}_m$ serves as a principled and comprehensive foundation for the subsequent stage.

**Dual-Flux Path Extraction** Our dual-flux path extraction method identifies a high-quality subgraph by combining two complementary reasoning paths extracted from the initial semantic subgraph $\mathcal{H}_m$. These paths, a "lower-bound" and an "upper-bound," are designed to collectively balance reliability, density, and information richness.

The process begins by filtering for a set of viable candidate paths $\mathcal{P}_{\text{dense}}$, keeping only those whose density $D(P)$ exceeds a threshold $\eta$:

$$\mathcal{P}_{\text{dense}} = \{P \subseteq \mathcal{H}_m \mid D(P) \geq \eta\}, \quad (2)$$

where the density of a path $P$ is the average triples weight:

$$D(P) = \frac{\sum_{\zeta \in P} w(\zeta)}{|m|}, \quad |m| \text{ is the number of triples in } P. \quad (3)$$

We evaluate each candidate path $P$ in $\mathcal{P}_{\text{dense}}$ based on two key metrics. The bottleneck flux $\phi(P)$ identifies the path's weakest link by taking the minimum edge weight. The total information $\psi(P)$ captures the cumulative value, calculated as the sum of all edge weights.

$$\phi(P) = \min_{\zeta \in P} w(\zeta), \quad \psi(P) = \sum_{\zeta \in P} w(\zeta). \quad (4)$$

*Lower-Bound (MaxMin-InfoPath):* This path $\mathcal{P}_{\text{lower}}$ establishes a baseline of reliability. It is identified through a lexicographical optimization that prioritizes the strongest possible "weakest link." We select the path that maximizes the

bottleneck flux $\phi(P)$ first, and then, among those, the one with the highest total information $\psi(P)$:

$$\mathcal{P}_{\text{lower}} = \arg\max_{P \in \mathcal{P}_{\text{dense}}} (\phi(P), \psi(P)). \quad (5)$$

*Upper-Bound (MaxDensity-InfoPath):* In parallel, this path, $\mathcal{P}_{\text{upper}}$, captures the most concentrated region of information. Its selection is a two-step process. First, we isolate an elite set of paths $\mathcal{P}^*$, that share the absolute maximum density:

$$\mathcal{P}^* = \left\{ P \in \mathcal{P}_{\text{dense}} \mid D(P) = \max_{P' \in \mathcal{P}_{\text{dense}}} D(P') \right\}. \quad (6)$$

From this maximum-density set, we then select the single path with the highest total information $\psi(P)$:

$$\mathcal{P}_{\text{upper}} = \arg\max_{P \in \mathcal{P}^*} \psi(P). \quad (7)$$

*Final Subgraph Synthesis:* The final, optimized subgraph $\mathcal{H}_q$ is formed by the union of these two paths.

$$\mathcal{H}_q = \mathcal{P}_{\text{lower}} \cup \mathcal{P}_{\text{upper}}. \quad (8)$$

Answer entities $e_a$ only create pseudo-labels to guide the retriever during training. Inference remains answer-free.

**Algorithmic Implementation** The detailed pseudocode of the proposed method is summarized in appendix.

## Theoretical Motivation

**Necessity of MaxMin-InfoPath** To motivate the design of *MaxMin-InfoPath*, we first analyze how local errors can compound in a multi-hop reasoning path.

**Definition 1** *Let a multi-hop path be denoted as $p = (e_1, e_2, \ldots, e_n)$, where each edge $e_i$ corresponds to a relation $r_i$ linking entities. We model the semantic transformation along each edge as a latent function $\phi_i : \mathbb{R}^d \to \mathbb{R}^d$. We assume there exists a true, ideal transformation $\Phi_i$, and our practical approximation $\phi_i$ is a perturbed version of it:*

$$\phi_i(x) = \Phi_i(x) + \varepsilon_i(x), \quad (9)$$

*where $\varepsilon_i(x) : \mathbb{R}^d \to \mathbb{R}^d$ is the step-wise error function.*

**Definition 2** *We define the full perturbed path function $f_p(x)$ and the ideal path function $F_p(x)$ as the composition of their respective step-wise functions:*

$$f_p(x) := \phi_n \circ \cdots \circ \phi_1(x), \quad F_p(x) := \Phi_n \circ \cdots \circ \Phi_1(x). \quad (10)$$

*The total path error $\delta_p(x)$ is the divergence between the two:*

$$\delta_p(x) := f_p(x) - F_p(x). \quad (11)$$

In our formulation, the step-wise error $\varepsilon_i(x)$ represents the semantic uncertainty or possibility of loss in each reasoning step. We use the weight of a triple $w(\zeta)$, as a proxy for this abstract error, in an inverse relationship. A high weight $w(\zeta)$ corresponds to a low error $\varepsilon_i$, and vice versa.

**Lemma 1** *Assume each ideal transformation $\Phi_i$ is differentiable and satisfies $\|\nabla \Phi_i(x)\| \leq \alpha_i < 1$, and each perturbation term is bounded by $\|\varepsilon_i(x)\| \leq \epsilon_i$. The total path error is then upper-bounded by:*

$$\|f_p(x) - F_p(x)\| \leq \sum_{i=1}^{n} \left( \prod_{j=i+1}^{n} \alpha_j \right) \cdot \epsilon_i, \quad (12)$$

*where $\epsilon_i$ is the approximation error at hop $i$, and $\alpha_j$ reflects the amplification strength of step $j$ via its Lipschitz constant.*

This formulation highlights a critical vulnerability: even small initial errors $\epsilon_i$ can be recursively amplified by subsequent steps $\alpha_j$, leading to significant degradation over long paths. To suppress this compounding effect, *MaxMin-InfoPath* prioritizes selecting paths whose intermediate steps exhibit consistently low local error, effectively minimizing the terms in the error bound. Assuming uniform bounds $\epsilon_i \leq \epsilon_{\min}$, $\alpha_i \leq \alpha_{\max} < 1$, the bound simplifies to:

$$\|\delta_p(x)\| \leq \epsilon_{\min} \cdot \frac{1 - \alpha_{\max}^n}{1 - \alpha_{\max}}. \quad (13)$$

This expression establishes a direct link between local error quality and global semantic stability, validating our strategy of enforcing high-fidelity transformations at every hop. A formal proof is provided in appendix.

**Necessity of MaxDensity-InfoPath** The theoretical foundation for *MaxDensity-InfoPath* lies in Bayesian inference. We seek the path $P$ that maximizes the a posteriori (MAP) probability given a query $q$. The standard MAP estimation approach is to maximize the sum of the log-likelihood and the log-prior, defining the optimal path $\hat{P}_{\text{MAP}}$ as:

$$\hat{P}_{\text{MAP}} = \arg\max_{P} (\log \Pr(q \mid P) + \log \Pr(P)). \quad (14)$$

Assuming the log-likelihood ($\log \Pr(q \mid P)$) is proportional to the total information and the log-prior ($\log \Pr(P)$) penalizes its length, this optimization objective simplifies. We show that the log-posterior probability ($\log \Pr(P \mid q)$) is a monotonic function of the density $D(P)$:

$$\log \Pr(P \mid q) \propto |m| \cdot (D(P) - \lambda). \quad (15)$$

Here, the parameter $\lambda$ is a positive constant derived from the prior probability that represents the penalty for path complexity. It quantifies the trade-off between the information density of a path and its length. This key relationship reveals that maximizing path density is not a heuristic, but a principled method for identifying the most statistically plausible evidence chain. This framework intrinsically balances the path's information $\psi(P)$ against its length $|m|$, thus establishing $D(P)$ as the optimal metric for this task. A formal proof is provided in the appendix.

## Differentiable Subgraph Optimization

**Motivation** Our goal is to optimize the structure of the subgraph to mitigate the position bias of LLMs (Liu et al. 2024c; Zhang et al. 2024a). As described in the introduction, we theoretically modeled this bias and confirmed that failure to account for it significantly degrades prediction fidelity, with detailed analysis in the appendix. This analysis establishes the need for a mechanism that explicitly optimizes the ranking of content within a subgraph.

**Differentiable Subgraph Ranking Supervision (DSRS)** To achieve the structural optimization motivated above, our strategy is to train the retriever to produce an importance score vector $x$ whose ordinal structure mirrors that of our pseudo-labels, $y$. This frames the problem as one of learning to rank, for which we must first establish a suitable objective. Solving this requires a metric that can robustly measure

the similarity of ordinal structures, independent of the actual score values. The Kendall rank correlation coefficient $\tau$ is an ideal candidate, as it provides a principled, scale-invariant measure of association based on pairwise comparisons.

**Definition 3** *Given two score vectors $x$ and $y$ of length $n$, a pair of distinct indices $(i, j)$ is:*

- ***Concordant** if $(x_i - x_j)(y_i - y_j) > 0$.*
- ***Discordant** if $(x_i - x_j)(y_i - y_j) < 0$.*

**Definition 4** *The coefficient $\tau(\mathbf{x}, \mathbf{y})$ is the normalized difference between the number of concordant pairs $N_{con}$ and discordant pairs $N_{dis}$:*

$$\tau(\mathbf{x}, \mathbf{y}) = \frac{N_{con} - N_{dis}}{N_0}, \tag{16}$$

*where $N_0 = \frac{n(n-1)}{2}$ is the total number of unique pairs.*

While $\tau$ directly measures our ranking goal, its discrete, combinatorial nature makes it non-differentiable, precluding its direct use in gradient-based optimization. To bridge this gap, we introduce a continuous and differentiable surrogate for $\tau$. Our key idea is to replace the non-differentiable sign function with a smooth approximation. For this purpose, the tanh function is a natural choice, as it is a standard method for creating differentiable "soft" versions of sign functions.

**Definition 5** *We introduce a differentiable indicator $\mathcal{I}_\beta(i, j)$ for each pair $(i, j)$ (Definition 3), which takes the form of a scaled* tanh *function:*

$$\mathcal{I}_\beta(i, j) = \frac{e^{\beta(x_i - x_j)(y_i - y_j)} - 1}{e^{\beta(x_i - x_j)(y_i - y_j)} + 1}, \tag{17}$$

*where $\beta > 0$ is a temperature hyperparameter that controls the steepness of the approximation. This function approximates the discrete $\pm 1$ indicator, as established by lemma 2.*

**Lemma 2** *The $\mathcal{I}_\beta(i, j)$ converges to the true discrete values as $\beta \to +\infty$: $\lim_{\beta \to +\infty} \mathcal{I}_\beta(i, j) = +1$ for concordant pairs, and $\lim_{\beta \to +\infty} \mathcal{I}_\beta(i, j) = -1$ for discordant pairs.*

By summing these indicators over all pairs, we obtain a differentiable approximation of Kendall's Tau, denoted as $\tilde{\tau}_\beta(\mathbf{x}, \mathbf{y})$. Following Lemma 2, $\tilde{\tau}_\beta$ provably converges to the true $\tau$ as $\beta \to +\infty$. Finally, we leverage our differentiable surrogate $\tilde{\tau}_\beta$ to construct DSRS. To maximize the differentiable rank correlation, we equivalently minimize its complement. By observing that the pairwise ranking agreement term in $\tilde{\tau}_\beta$ can be re-expressed using the sigmoid function $\sigma$:

$$\mathcal{L}_{\text{DSRS}} = \frac{1}{|N|} \sum_{(i,j) \in N} \left[1 - \sigma(\beta \Delta x_{ij} \Delta y_{ij})\right], \tag{18}$$

where $\Delta x_{ij} = x_i - x_j$, $\Delta y_{ij} = y_i - y_j$, $N$ is the set of unique pairs, and $|N| = N_0$ (Definition 3). Our final objective, formally derived in the appendix, is to minimize this pairwise ranking error. This aligns the retriever's predicted ranking $\mathbf{x}$ with the pseudo-label ranking $\mathbf{y}$, achieving our goal of structural optimization.

# Experiments

We conduct a comprehensive set of experiments to validate the efficacy of DCTR for KG-RAG, by evaluating how it excels at generating high-quality subgraphs, leverages structural information for complex multi-hop reasoning, and demonstrates generalization across diverse tasks.

**Datasets.** We evaluate our method on two challenging multi-hop knowledge graph question answering (KGQA) benchmarks: WebQSP (Yih et al. 2016) and CWQ (Talmor and Berant 2018), both derived from Freebase (Bollacker et al. 2008). To ensure evaluation rigor, we use the cleaned "sub" variants of these datasets, which filter out samples with missing answers (Li, Miao, and Li 2025).

**Experiment Settings.** We initialize triple representations with the GTE encoder (Li et al. 2023b), a strong performer on the MTEB benchmark (Muennighoff et al. 2023). The retriever is trained on DCFGR-generated pseudo-labels using a combined binary and DSRS loss (*hyperparameter* = 0.5). Experiments are run on 8x NVIDIA A100 (80GB) GPUs using Llama3.1-8B/70B-Instruct models (denoted as 8B and 70B), with inference accelerated by vLLM (Kwon et al. 2023). We reproduce baselines like SubgraphRAG from public code or cite original results. All reported metrics are averaged over five runs with different random seeds. For further details, including complete experimental data and setup, as well as experiments on hyperparameters such as temperature $\beta$ and threshold $\eta$, please see the appendix.

## Retrieval Results

**Baseline.** We compare DCTR against diverse KGQA methods, including embedding-based approaches like Cosine Similarity (Li et al. 2023a) and G-Retriever's k-nearest neighbor search (He et al. 2024), constrained path search methods such as SR+NSM w/ E2E (Zhang et al. 2022) and Retrieve-Rewrite-Answer (Wu et al. 2023), as well as several approaches that rely on a shortest path heuristic, like RoG (LUO et al. 2024), GNN-RAG (Mavromatis and Karypis 2024), and SubgraphRAG (Li, Miao, and Li 2025).

**Evaluation Metrics.** We assess retrieval quality using three complementary metrics: path triple recall to measure alignment with the supervised ground-truth paths, GPT-4o triple recall for a broader semantic evaluation against a GPT-4o-generated superset, and answer entity recall to measure if the answer is present in the retrieved subgraph.

**Overall Evaluation** The results in Table 1 demonstrate the superiority of our DCTR. DCTR outperforms all baseline methods in retrieval effectiveness across both datasets. While baselines like SubgraphRAG are competitive on path recall, DCTR establishes a commanding lead when evaluated on the more challenging GPT-4o triples. On this more robust metric, DCTR achieves a relative improvement of 5.1% on WebQSP and 3.2% on CWQ over SubgraphRAG, showcasing its ability to generate more semantically comprehensive subgraphs. As detailed in Table 2, DCTR's performance margin over other methods becomes even more pronounced on queries requiring Multi-hop. On these tasks,

| Method | WebQSP | | | | CWQ | | | | CWQ → WebQSP | | | WebQSP → CWQ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Path | GPT-4o | Answer | Time (s) | Path | GPT-4o | Answer | Time (s) | Path | GPT-4o | Answer | Path | GPT-4o | Answer |
| cosine similarity | 0.714 | 0.719 | 0.708 | **3** | 0.488 | 0.567 | 0.582 | 13 | 0.714 | 0.719 | 0.708 | 0.488 | 0.567 | 0.582 |
| RoG | 0.713 | 0.388 | 0.807 | 948 | 0.623 | 0.298 | 0.841 | 2327 | 0.589 | 0.323 | 0.658 | 0.301 | 0.139 | 0.412 |
| G-Retriever | 0.294 | 0.325 | 0.545 | 672 | 0.183 | 0.217 | 0.375 | 1530 | 0.294 | 0.325 | 0.545 | 0.183 | 0.217 | 0.375 |
| GNN-RAG | 0.522 | 0.405 | 0.818 | 68 | 0.500 | 0.386 | 0.841 | 160 | 0.446 | 0.364 | 0.691 | 0.444 | 0.351 | 0.697 |
| SubgraphRAG | 0.857 | 0.815 | 0.923 | 6 | 0.801 | 0.807 | 0.914 | 12 | 0.802 | 0.789 | 0.897 | 0.624 | 0.626 | 0.793 |
| **Ours (DCTR)** | **0.890** | **0.866** | **0.956** | 5 | **0.829** | **0.839** | **0.928** | **12** | **0.836** | **0.819** | **0.908** | **0.634** | **0.634** | **0.811** |

Table 1: Comparison of retrieval recall performance across four evaluation settings. Best results are in bold.

| Method | Path Triple Recall | | | | | GPT-4o Triple Recall | | | | | Answer Entity Recall | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | WebQSP | | CWQ | | | WebQSP | | CWQ | | | WebQSP | | CWQ | | |
| | 1 | 2 | 1 | 2 | ≥3 | 1 | 2 | 1 | 2 | ≥3 | 1 | 2 | 1 | 2 | ≥3 |
| cosine similarity | 0.874 | 0.405 | 0.629 | 0.442 | 0.333 | 0.847 | 0.483 | 0.629 | 0.511 | 0.464 | 0.943 | 0.253 | 0.903 | 0.472 | 0.289 |
| RoG | 0.869 | 0.415 | 0.766 | 0.597 | 0.253 | 0.446 | 0.271 | 0.347 | 0.293 | 0.122 | 0.874 | 0.677 | 0.920 | 0.827 | 0.628 |
| G-Retriever | 0.335 | 0.216 | 0.134 | 0.205 | 0.168 | 0.345 | 0.284 | 0.159 | 0.240 | 0.226 | 0.596 | 0.446 | 0.377 | 0.384 | 0.269 |
| GNN-RAG | 0.532 | 0.502 | 0.515 | 0.498 | 0.446 | 0.384 | 0.445 | 0.328 | 0.408 | 0.418 | 0.810 | 0.831 | 0.853 | 0.841 | 0.787 |
| SubgraphRAG | 0.951 | 0.745 | 0.828 | 0.819 | 0.623 | 0.905 | 0.811 | 0.820 | 0.858 | 0.749 | 0.967 | 0.877 | 0.942 | 0.915 | 0.738 |
| **Ours (DCTR)** | **0.971** | **0.753** | **0.883** | **0.831** | **0.652** | **0.923** | **0.830** | **0.905** | **0.892** | **0.783** | **0.982** | **0.886** | **0.954** | **0.921** | **0.766** |

Table 2: Multi-hop retrieval recall evaluation. Best results are in bold.

| Method | WebQSP | | | CWQ | | |
|---|---|---|---|---|---|---|
| | Path | GPT-4o | Answer | Path | GPT-4o | Answer |
| DCTR | **0.890** | **0.866** | **0.956** | **0.829** | **0.839** | **0.928** |
| w/o $\mathcal{P}_{\text{upper}}$ | 0.862 | 0.843 | 0.929 | 0.804 | 0.809 | 0.905 |
| w/o $\mathcal{P}_{\text{lower}}$ | 0.851 | 0.835 | 0.913 | 0.799 | 0.812 | 0.907 |
| w/o DSRS | 0.874 | 0.853 | 0.931 | 0.816 | 0.820 | 0.918 |

Table 3: Ablation study of the retrieval stage.

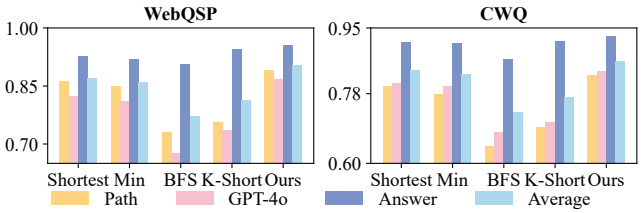


Figure 2: Performance of different path-finding methods.

DCTR achieves a relative improvement of over 8% in GPT-4o triple recall across the two datasets, 4% in answer recall on CWQ, and 5% in path recall, when compared to the strongest baseline. This confirms that DCTR's dual-constraint optimization successfully overcomes the limitations of the simpler heuristics employed by prior work.

**Cross-Dataset Generalization and Ablation Studies.** DCTR's robustness is demonstrated through a cross-dataset generalization experiment ($A \rightarrow B$ in Table 1), where it outperforms all baselines despite domain gaps. We validate our design choices through ablation studies. First, as shown in Table 3, removing lower bound (w/o $\mathcal{P}_{\text{lower}}$) or upper bound (w/o $\mathcal{P}_{\text{upper}}$) paths leads to performance degradation, validating their complementary effects. Removing the DSRS module leads to some performance degradation, demonstrating the positive effect of DSRS on improving subgraph quality. Second, a comparison against standard heuristics, such as Shortest Path, Minimum-Cost Path, BFS, and K-Shortest Paths, shows that our DCFGR is superior across all metrics (Figure 2). This confirms that DCTR's effectiveness is rooted in its superior pseudo-label generation process.

## Generation Results

**Baselines.** We compare DCTR's end-to-end performance against a set of recent reasoning methods, including UniKGQA (Jiang et al. 2023), KD-CoT (Wang et al. 2023), TOG (Sun et al. 2024), ETD (Liu et al. 2024b), RoG (LUO et al. 2024) and SubgraphRAG (Li, Miao, and Li 2025).

**Evaluation Metrics.** Beyond the standard Macro-F1 and Hit@1, we incorporate Micro-F1 to provide a more robust evaluation on instances with multiple reference answers. We also measure Hit, which assesses whether at least one correct answer appears anywhere in the LLM's generated response.To gain deeper insights into factuality, we also adopt the score$_h$ (Yang et al. 2024), which evaluates both the accuracy and the degree of hallucination in generated responses. This metric is particularly valuable as it penalizes hallucinated outputs more heavily than omitted answers.

| Method | WebQSP-sub | | | | | CWQ-sub | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Macro-F1 | Micro-F1 | Hit | Hit@1 | Score$_h$ | Macro-F1 | Micro-F1 | Hit | Hit@1 | Score$_h$ |
| SR+NSM w E2E | 58.79 | 37.04 | 68.63 | 60.62 | 64.44 | – | – | – | – | – |
| G-Retriever | 54.13 | 23.84 | 74.52 | 67.56 | 67.97 | – | – | – | – | – |
| RoG (Llama2-7B tuned) | 67.94 | 43.10 | 84.03 | 77.61 | 72.79 | 57.69 | 52.83 | 64.64 | 60.64 | 54.51 |
| SubgraphRAG (8B) | 71.92 | 45.66 | 88.45 | 84.31 | 81.14 | 54.26 | 50.79 | 65.41 | 59.36 | 62.44 |
| SubgraphRAG (70B) | 75.37 | 51.16 | 87.22 | 85.15 | 85.27 | 60.39 | 59.10 | 68.13 | 64.71 | 67.42 |
| DCTR + 8B | 73.85 | 47.28 | 89.74 | 86.21 | 83.11 | 57.44 | 54.15 | 69.21 | 62.32 | 64.50 |
| DCTR + 8B($\leftrightarrow$) | 69.13 | 43.57 | 87.35 | 82.74 | 80.31 | 43.49 | 44.36 | 56.54 | 48.63 | 57.95 |
| DCTR + 70B | **77.29** | **52.22** | **89.69** | **87.43** | **86.94** | **63.27** | **61.70** | **71.30** | **67.51** | **69.83** |

Table 4: Evaluation results on WebQSP-sub and CWQ-sub datasets across multiple metrics.

| Method | WebQSP | | CWQ | |
|---|---|---|---|---|
| | Macro-F1 | Hit | Macro-F1 | Hit |
| UniKGQA | 70.2 | – | 49.0 | – |
| KD-CoT | 52.5 | 68.6 | – | 55.7 |
| SR+NSM w E2E | 64.1 | – | 46.3 | – |
| ToG (Llama2-70B-chat) | – | 68.9 | – | 57.6 |
| RoG | 66.45 | 82.19 | 53.87 | 60.55 |
| EtD (Llama2-13B-chat) | – | 77.4 | – | 57.7 |
| GNN-RAG | 71.3 | 85.7 | **59.4** | **66.8** |
| SubgraphRAG (8B) | 70.49 | 86.57 | 47.08 | 56.85 |
| SubgraphRAG (70B) | 74.52 | 86.21 | 51.56 | 57.45 |
| DCTR + 8B | 72.13 | **88.05** | 50.12 | 59.72 |
| DCTR + 8B($\leftrightarrow$) | 67.52 | 84.62 | 38.92 | 49.19 |
| DCTR + 70B | **75.72** | 87.89 | 53.17 | 60.17 |

Table 5: Generate performance. GNN-RAG, RoG, KD-CoT, and G-Retriever use 7B fine-tuned Llama2 models.

**Overall Performance.** As detailed in Tables 4 and 5, our DCTR establishes new state-of-the-art results on both the WebQSP benchmark and its cleaned subset. When configured with a 70B model, DCTR solidifies this dominance, outperforming the strongest baseline (SubgraphRAG-70B) by a relative margin of 2% in Macro-F1 on WebQSP-sub. At the 8B scale, DCTR significantly surpasses its direct supervised competitor, SubgraphRAG-8B, with a 2.5% relative gain in Macro-F1. This result showcases DCTR's superior retrieval and ranking strategy, which allows it to achieve top-tier performance even with smaller models. On the more demanding, multi-hop CWQ dataset, DCTR's key advantage is that it is entirely fine-tuning-free. While achieving performance competitive with frameworks like GNN-RAG that require costly LLMs fine-tuning, DCTR's inference optimization enables significant gains of up to 6.4% in Macro-F1 on CWQ-sub compared to strong, non-fine-tuned baselines.

**Cross-Dataset Generalization and Ablation Studies.** DCTR's robustness is demonstrated in cross-dataset ($\leftrightarrow$) generalization experiments (Tables 4 and 5). We validate our design through ablation studies (Table 6). Removing either

| Method | WebQSP | | CWQ | |
|---|---|---|---|---|
| | Macro-F1 | Hit | Macro-F1 | Hit |
| DCTR + 8B | **72.13** | **88.05** | **50.12** | **59.72** |
| w/o $\mathcal{P}_{\text{upper}}$ | 70.33 | 85.96 | 48.14 | 57.31 |
| w/o $\mathcal{P}_{\text{lower}}$ | 69.94 | 85.02 | 46.92 | 56.85 |
| w/o DSRS | 65.73 | 81.59 | 45.10 | 54.91 |

Table 6: Ablation study of the generation stage.

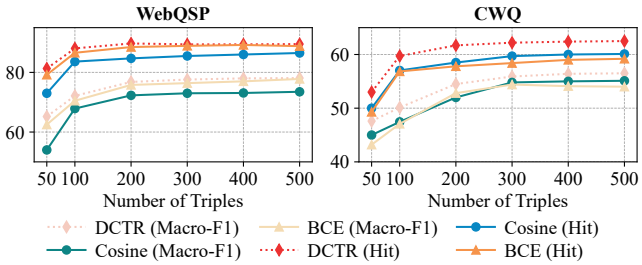


Figure 3: Performance of different loss methods.

the lower-bound (w/o $\mathcal{P}_{\text{lower}}$) or upper-bound (w/o $\mathcal{P}_{\text{upper}}$) path causes performance degradation, confirming their complementary roles. A significant performance drop upon removing the DSRS module underscores its necessity for mitigating positional bias. Furthermore, Figure 3 shows it vastly outperforms alternative objectives like cosine similarity or binary classification loss, confirming that it is the most critical component of our model.

## Conclusion

In this paper, we propose DCTR to improve retrieval by jointly considering subgraph structural integrity and information salience. The DCFGR module generates structurally compact, semantically rich pseudo-labels using a dual method of maximum flow retrieval and dual flux path extraction. The DSRS's differentiable surrogate optimizes the retriever's ranking function by aligning its attention to pseudo-labels, which highlights salient content and overcomes issues with nondifferentiable metrics.

## Acknowledgments

This work was supported by the Shanghai Municipal Education Commission Artificial Intelligence Plan (Z2024-119) and the Innovation Special Fund Project in Shanghai University of Electric Power.

## References

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, 1247–1250. New York, NY, USA: Association for Computing Machinery. ISBN 9781605581026.

Cahoon, J.; Singh, P.; Litombe, N.; Larson, J.; Trinh, H.; Zhu, Y.; Mueller, A.; Psallidas, F.; and Curino, C. 2025. Optimizing open-domain question answering with graph-based retrieval augmented generation. arXiv:2503.02922.

Edge, D.; Trinh, H.; Cheng, N.; Bradley, J.; Chao, A.; Mody, A.; Truitt, S.; Metropolitansky, D.; Ness, R. O.; and Larson, J. 2025. From Local to Global: A Graph RAG Approach to Query-Focused Summarization. arXiv:2404.16130.

Guo, T.; Yang, Q.; Wang, C.; Liu, Y.; Li, P.; Tang, J.; Li, D.; and Wen, Y. 2024. KnowledgeNavigator: leveraging large language models for enhanced reasoning over knowledge graph. *Complex & Intelligent Systems*, 10(5): 7063–7076.

Guo, Z.; Xia, L.; Yu, Y.; Ao, T.; and Huang, C. 2025. LightRAG: Simple and Fast Retrieval-Augmented Generation. arXiv:2410.05779.

He, X.; Tian, Y.; Sun, Y.; Chawla, N. V.; Laurent, T.; LeCun, Y.; Bresson, X.; and Hooi, B. 2024. G-Retriever: Retrieval-Augmented Generation for Textual Graph Understanding and Question Answering. In Globerson, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J.; and Zhang, C., eds., *Advances in Neural Information Processing Systems*, volume 37, 132876–132907. Curran Associates, Inc.

Hou, Z.; Jin, X.; Li, Z.; and Bai, L. 2021. Rule-Aware Reinforcement Learning for Knowledge Graph Reasoning. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 4687–4692. Online: Association for Computational Linguistics.

Ji, Y.; Wu, K.; Li, J.; Chen, W.; Zhong, M.; Jia, X.; and Zhang, M. 2024. Retrieval and Reasoning on KGs: Integrate Knowledge Graphs into Large Language Models for Complex Question Answering. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Findings of the Association for Computational Linguistics: EMNLP 2024*, 7598–7610. Miami, Florida, USA: Association for Computational Linguistics.

Jiang, J.; Zhou, K.; Zhao, W. X.; Song, Y.; Zhu, C.; Zhu, H.; and Wen, J.-R. 2024. KG-Agent: An Efficient Autonomous Agent Framework for Complex Reasoning over Knowledge Graph. arXiv:2402.11163.

Jiang, J.; Zhou, K.; Zhao, X.; and Wen, J.-R. 2023. UniKGQA: Unified Retrieval and Reasoning for Solving Multi-hop Question Answering Over Knowledge Graph. In *The Eleventh International Conference on Learning Representations*.

Jin, B.; Xie, C.; Zhang, J.; Roy, K. K.; Zhang, Y.; Li, Z.; Li, R.; Tang, X.; Wang, S.; Meng, Y.; and Han, J. 2024. Graph Chain-of-Thought: Augmenting Large Language Models by Reasoning on Graphs. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Findings of the Association for Computational Linguistics: ACL 2024*, 163–184. Bangkok, Thailand: Association for Computational Linguistics.

Kim, K.; and Lee, J.-Y. 2024. RE-RAG: Improving Open-Domain QA Performance and Interpretability with Relevance Estimator in Retrieval-Augmented Generation. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 22149–22161. Miami, Florida, USA: Association for Computational Linguistics.

Kwon, W.; Li, Z.; Zhuang, S.; Sheng, Y.; Zheng, L.; Yu, C. H.; Gonzalez, J.; Zhang, H.; and Stoica, I. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, 611–626. New York, NY, USA: Association for Computing Machinery. ISBN 9798400702297.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; Riedel, S.; and Kiela, D. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 9459–9474. Curran Associates, Inc.

Li, M.; Miao, S.; and Li, P. 2025. Simple is Effective: The Roles of Graphs and Large Language Models in Knowledge-Graph-Based Retrieval-Augmented Generation. In *ICLR 2025 Workshop on Foundation Models in the Wild*.

Li, S.; Gao, Y.; Jiang, H.; Yin, Q.; Li, Z.; Yan, X.; Zhang, C.; and Yin, B. 2023a. Graph Reasoning for Question Answering with Triplet Retrieval. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Findings of the Association for Computational Linguistics: ACL 2023*, 3366–3375. Toronto, Canada: Association for Computational Linguistics.

Li, Z.; Zhang, X.; Zhang, Y.; Long, D.; Xie, P.; and Zhang, M. 2023b. Towards General Text Embeddings with Multi-stage Contrastive Learning. arXiv:2308.03281.

Liu, G.; Zhang, Y.; Li, Y.; and Yao, Q. 2024a. Explore then Determine: A GNN-LLM Synergy Framework for Reasoning over Knowledge Graph. *CoRR*, abs/2406.01145.

Liu, G.; Zhang, Y.; Li, Y.; and Yao, Q. 2024b. Explore then Determine: A GNN-LLM Synergy Framework for Reasoning over Knowledge Graph. *CoRR*, abs/2406.01145.

Liu, H.; Wang, Z.; Chen, X.; Li, Z.; Xiong, F.; Yu, Q.; and Zhang, W. 2025. HopRAG: Multi-Hop Reasoning for Logic-Aware Retrieval-Augmented Generation. arXiv:2502.12442.

Liu, N. F.; Lin, K.; Hewitt, J.; Paranjape, A.; Bevilacqua, M.; Petroni, F.; and Liang, P. 2024c. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics*, 12: 157–173.

LUO, L.; Li, Y.-F.; Haffari, G.; and Pan, S. 2024. Reasoning on Graphs: Faithful and Interpretable Large Language Model Reasoning. In *The Twelfth International Conference on Learning Representations*.

Mavromatis, C.; and Karypis, G. 2024. GNN-RAG: Graph Neural Retrieval for Large Language Model Reasoning. arXiv:2405.20139.

Muennighoff, N.; Tazi, N.; Magne, L.; and Reimers, N. 2023. MTEB: Massive Text Embedding Benchmark. In Vlachos, A.; and Augenstein, I., eds., *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, 2014–2037. Dubrovnik, Croatia: Association for Computational Linguistics.

Quinn, D.; Nouri, M.; Patel, N.; Salihu, J.; Salemi, A.; Lee, S.; Zamani, H.; and Alian, M. 2025. Accelerating Retrieval-Augmented Generation. In *Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1*, ASPLOS '25, 15–32. New York, NY, USA: Association for Computing Machinery. ISBN 9798400706981.

Sun, J.; Xu, C.; Tang, L.; Wang, S.; Lin, C.; Gong, Y.; Ni, L.; Shum, H.-Y.; and Guo, J. 2024. Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph. In *The Twelfth International Conference on Learning Representations*.

Talmor, A.; and Berant, J. 2018. The Web as a Knowledge-Base for Answering Complex Questions. In Walker, M.; Ji, H.; and Stent, A., eds., *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 641–651. New Orleans, Louisiana: Association for Computational Linguistics.

Tang, X.; Li, J.; Du, N.; and Xie, S. 2025. Adapting to Non-Stationary Environments: Multi-Armed Bandit Enhanced Retrieval-Augmented Generation on Knowledge Graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(12): 12658–12666.

Wang, H.; Huang, W.; Deng, Y.; Wang, R.; Wang, Z.; Wang, Y.; Mi, F.; Pan, J. Z.; and Wong, K.-F. 2024. UniMS-RAG: A Unified Multi-source Retrieval-Augmented Generation for Personalized Dialogue Systems. *ArXiv*, abs/2401.13256.

Wang, K.; Duan, F.; Wang, S.; Li, P.; Xian, Y.; Yin, C.; Rong, W.; and Xiong, Z. 2023. Knowledge-Driven CoT: Exploring Faithful Reasoning in LLMs for Knowledge-intensive Question Answering. arXiv:2308.13259.

Wu, Y.; Hu, N.; Bi, S.; Qi, G.; Ren, J.; Xie, A.; and Song, W. 2023. Retrieve-Rewrite-Answer: A KG-to-Text Enhanced LLMs Framework for Knowledge Graph Question Answering. arXiv:2309.11206.

Yang, X.; Sun, K.; Xin, H.; Sun, Y.; Bhalla, N.; Chen, X.; Choudhary, S.; Gui, R. D.; Jiang, Z. W.; Jiang, Z.; Kong, L.; Moran, B.; Wang, J.; Xu, Y. E.; Yan, A.; Yang, C.; Yuan, E.; Zha, H.; Tang, N.; Chen, L.; Scheffer, N.; Liu, Y.; Shah, N.; Wanga, R.; Kumar, A.; Yih, W.-t.; and Dong, X. L. 2024. CRAG - Comprehensive RAG Benchmark. In Globerson, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J.; and Zhang, C., eds., *Advances in Neural Information Processing Systems*, volume 37, 10470–10490. Curran Associates, Inc.

Yih, W.-t.; Richardson, M.; Meek, C.; Chang, M.-W.; and Suh, J. 2016. The Value of Semantic Parse Labeling for Knowledge Base Question Answering. In Erk, K.; and Smith, N. A., eds., *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 201–206. Berlin, Germany: Association for Computational Linguistics.

Zhang, F.; Zhu, D.; Ming, J.; Jin, Y.; Chai, D.; Yang, L.; Tian, H.; Fan, Z.; and Chen, K. 2025a. DH-RAG: A Dynamic Historical Context-Powered Retrieval-Augmented Generation Method for Multi-Turn Dialogue. *ArXiv*, abs/2502.13847.

Zhang, J.; Zhang, X.; Yu, J.; Tang, J.; Tang, J.; Li, C.; and Chen, H. 2022. Subgraph Retrieval Enhanced Model for Multi-hop Knowledge Base Question Answering. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 5773–5784. Dublin, Ireland: Association for Computational Linguistics.

Zhang, N.; Zhang, C.; Tan, Z.; Yang, X.; Deng, W.; and Wang, W. 2025b. Credible plan-driven RAG method for Multi-hop Question Answering. arXiv:2504.16787.

Zhang, T.; Li, D.; Chen, Q.; Wang, C.; Huang, L.; Xue, H.; He, X.; and Huang, J. 2024a. R4: Reinforced Retriever-Reorder-Responder for Retrieval-Augmented Large Language Models. arXiv:2405.02659.

Zhang, Z.; Chen, R.; Liu, S.; Yao, Z.; Ruwase, O.; Chen, B.; Wu, X.; and Wang, Z. 2024b. Found in the Middle: How Language Models Use Long Contexts Better via Plug-and-Play Positional Encoding. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Zhu, X.; Xie, Y.; Liu, Y.; Li, Y.; and Hu, W. 2025. Knowledge Graph-Guided Retrieval Augmented Generation. In Chiruzzo, L.; Ritter, A.; and Wang, L., eds., *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)*, 8912–8924. Albuquerque, New Mexico: Association for Computational Linguistics. ISBN 979-8-89176-189-6.