Think Parallax: Solving Multi-Hop Problems via Multi-View Knowledge-Graph-Based Retrieval-Augmented Generation

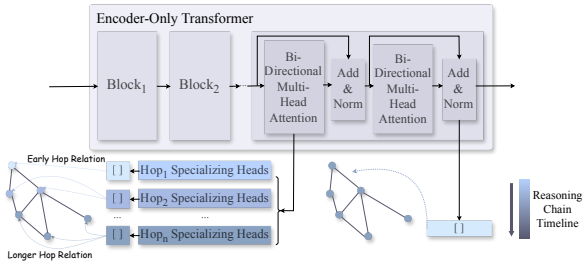
Jinliang Liu[1,2], **Jiale Bai**[2], **Shaoning Zeng**[1,2*]
[1]Yangtze Delta Region Institute (Huzhou), University of Electronic Science and Technology of China, Huzhou, Zhejiang, China
[2]School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan, China
lucaliu510@gmail.com, syxb02@gmail.com, zsn@outlook.com

Abstract

Large language models (LLMs) still struggle with multi-hop reasoning over knowledge-graphs (KGs), and we identify a previously overlooked structural reason for this difficulty: Transformer attention heads naturally specialize in distinct semantic relations across reasoning stages, forming a hop-aligned relay pattern. This key finding suggests that multi-hop reasoning is inherently multi-view, yet existing KG-based retrieval-augmented generation (KG-RAG) systems collapse all reasoning hops into a single representation, flat embedding space, suppressing this implicit structure and causing noisy or drifted path exploration. We introduce ParallaxRAG, a symmetric multi-view framework that decouples queries and KGs into aligned, head-specific semantic spaces. By enforcing relational diversity across multiple heads while constraining weakly related paths, ParallaxRAG constructs more accurate, cleaner subgraphs and guides LLMs through grounded, hop-wise reasoning. On WebQSP and CWQ, it achieves state-of-the-art retrieval and QA performance, substantially reduces hallucination, and generalizes strongly to the biomedical BioASQ benchmark. Our implementation is available at https://github.com/LucaLiu1313/ParallaxRAG.

1 Introduction

Multi-hop reasoning over KGs remains a challenge for LLMs. Although RAG improves factual grounding (Lewis et al., 2020), existing KG-RAG systems still struggle with long reasoning chains (Luo et al., 2023), compounding errors (Mavromatis and Karypis, 2024), and the rapid expansion of irrelevant graph paths (He et al., 2024). These issues limit both the accuracy and robustness of downstream question answering.



(a) ParallaxRAG: Hop-Specialized Multi-Head Embeddings **(b) Standard KG-RAG: Single Embedding Space**

Figure 1: Comparison of ParallaxRAG and standard KG-RAG embedding strategies. (a) In ParallaxRAG, attention heads specialize in distinct semantic relations across reasoning stages, where early heads capture shallow relational patterns while later heads handle longer-hop dependencies. (b) In standard KG-RAG, all reasoning hops are collapsed into a single embedding space, suppressing hop-specific semantic structure.

Our analysis identifies an architectural property of Transformers that is directly relevant to this problem, where different attention heads consistently specialize in distinct semantic relations across reasoning stages, forming a hop-aligned pattern. Some attention heads primarily contribute to the early stages of reasoning, while others specialize in later-stage reasoning and make dominant contributions as the reasoning process deepens. This observation suggests that multi-hop reasoning is inherently multi-view, with different hops requiring distinct representational subspaces.

Current KG-RAG approaches do not leverage this structure. Most methods encode queries and graph triples into a single embedding space (Fu et al., 2020; Ji et al., 2021), which conflates hop-specific semantics and hinders step-wise reasoning. This monolithic representation causes early-hop signals to interfere with deeper relational composition, increasing noise and reducing retrieval

*Corresponding author

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 26637–26655

precision (Liu et al., 2023). Iterative LLM-based reasoning (Sun et al., 2023) offers partial improvements but introduces high latency and lacks an explicit mechanism for hop decomposition over KGs. These limitations indicate a structural mismatch: multi-hop reasoning requires hop-separated semantics, yet existing KG-RAG systems enforce a single shared embedding basis.

To instantiate this principle, we introduce ParallaxRAG, a framework based on symmetric multi-head decoupling. We leverage transformer multi-head activations to decompose queries into multi-view representations (Vaswani et al., 2017; Besta et al., 2024), while projecting the KG triples into aligned multi-faceted latent spaces (Mavromatis and Karypis, 2024; Li et al., 2024). Two mechanisms operationalize this framework: (1) a Pairwise Similarity Regularization (PSR) module integrated into the Distance Encoding (DE) stage, which enforces head-level specialization and prevents representational collapse; and (2) a lightweight retrieval component that consolidates head-specific information to reduce noise and improve alignment between queries and graph structure (Khattab and Zaharia, 2020).

Our main contributions are as follows:

- **Multi-head decoupling architecture for KG-RAG.** We propose the first KG-RAG framework that decouples queries and KGs into head-specific views, allowing attention heads to capture complementary relational cues at different reasoning depths. We also introduce a joint exploration–exploitation strategy, namely PSR and weakly supervised gating which enhances robustness and promotes specialization.
- **Head specialization in multi-hop reasoning.** We provide the first empirical evidence of a relay effect, where distinct head groups dominate different stages of multi-hop reasoning. New metrics quantify head-level contribution and effectiveness.
- **State-of-the-Art Performance and Cross-Domain Generalization.** ParallaxRAG achieves state-of-the-art results on WebQSP and CWQ, and generalizes to the biomedical BioASQ benchmark. Under zero-shot transfer, it surpasses the previous SOTA by 7.32 Macro-F1, demonstrating transferable reasoning capabilities across domains.

2 Related Work

2.1 Retrieval-Augmented Generation (RAG)

RAG was proposed to ground LLMs in external knowledge to mitigate hallucination (Lewis et al., 2020). Dense retrievers showed strong results (Karpukhin et al., 2020; Izacard et al., 2022; Gao et al., 2023), but struggled with unstructured or redundant data (Izacard and Grave, 2020; Petroni et al., 2020). KG-RAG leveraged structured KGs (Peng et al., 2024) to enable more precise evidence retrieval. GNN-RAG (Mavromatis and Karypis, 2024) propagated information via graph neural networks, encoding queries and triples into a shared embedding space. SubgraphRAG (Li et al., 2024) improved subgraph retrieval quality through lightweight structural scoring. Both approaches, however, operated in a single, monolithic embedding space and did not model relational structure at the level of individual reasoning hops. We instead focus on query–graph representations that explicitly decompose reasoning by hops.

2.2 Multi-Head Embeddings

Multi-head attention (MHA) variants improved embeddings across modalities, e.g., visual semantic alignment via self-attention (Park et al., 2020), redundancy reduction (Bhojanapalli et al., 2021; Bian et al., 2021), and low-resource multilingual tasks (Vashisht et al., 2025). Recent interpretability work has shown that attention heads in LLMs tend to specialize in distinct semantic roles, exhibiting consistent patterns across inputs and architectures (Zheng et al., 2025; Basile et al., 2025). In RAG, MRAG (Besta et al., 2024) leveraged multiple attention heads as parallel retrieval channels to capture diverse semantic aspects of a query, working well for unstructured textual corpora. However, MRAG was designed around independent, non-aligned head channels, where query representations and context embeddings need to be projected into consistent semantic subspaces across reasoning hops. We address this gap by symmetrically decomposing both queries and KG triples into shared head-specific subspaces.

2.3 Multi-Hop QA

Prior methodologies generally bifurcated into enhancing LLM reasoning via decomposition or prompting (Perez et al., 2020; Fu et al., 2021; Wei et al., 2022), and improving retrieval via iterative or adaptive methods (Trivedi et al., 2022; Asai et al.,

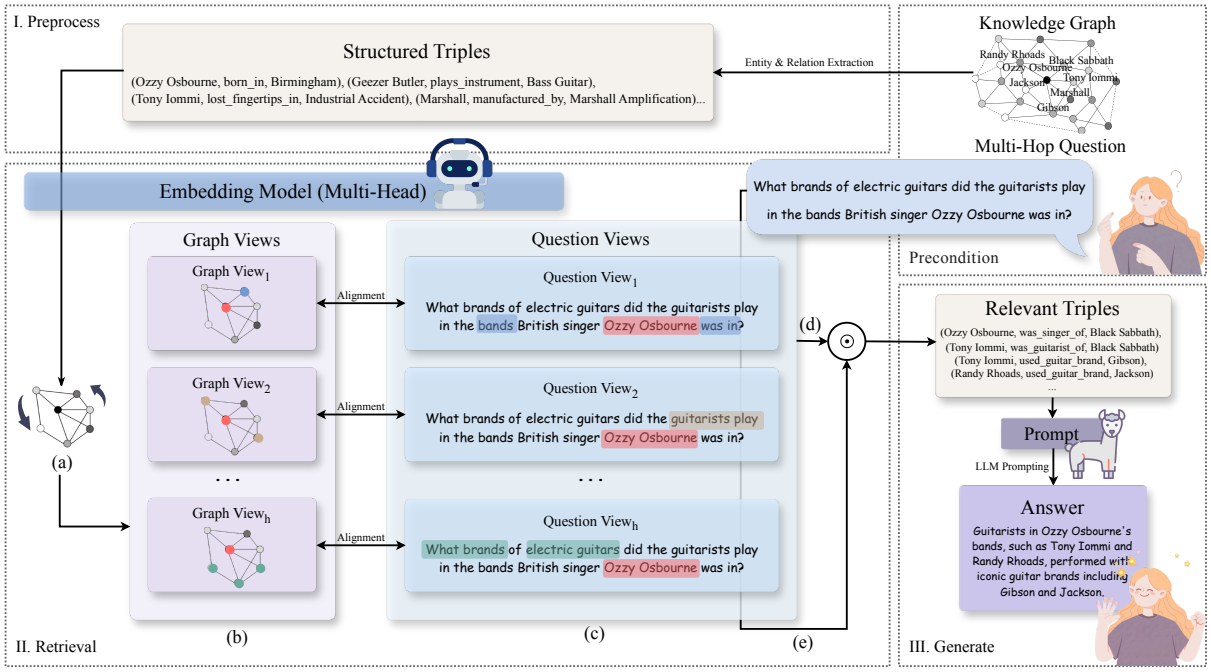


Figure 2: The architecture of ParallaxRAG, illustrating its three main stages: Preprocess, Retrieval, and Generate. The core retrieval process begins by creating specialized multi-view representations for both the graph (b) and the question (c), a process guided by directional distance encoding and regularized by pairwise similarity (a). Next, candidate triples are scored in parallel across these diverse views (d). Finally, a query-aware gating mechanism computes head importance (e) to produce a weighted aggregation of the scores, identifying the most relevant triples for the final generation stage.

2024). LLM-prompted methods such as StructGPT (Jiang et al., 2023) and ToG (Sun et al., 2023) enabled step-wise KG traversal but required multiple LLM calls and incurred high latency. EtD (Liu et al., 2024) reduced cost with a lightweight GNN retriever, but operated within a single-view embedding space. Path-based methods such as RoG (Luo et al., 2023) planned relation paths with full supervision, treating retrieval and planning as disjoint stages. RoE (Han et al., 2026) unified exploration and generation through Reinforcement Learning (RL) but incurred high trajectory training costs. Deep GraphRAG (Li et al., 2026) introduced hierarchical global-local retrieval via RL, yet required complex hierarchy maintenance. Across these methods, hop-level representation alignment between the query and the graph remains largely unaddressed. We approach this with an encoder-only Transformer trained under a weakly supervised objective, where multi-head attention is organized to reflect the hop structure of multi-hop reasoning.

3 The ParallaxRAG Framework

Multi-hop reasoning over knowledge graphs requires connecting multiple relational steps in a consistent manner. However, most KG-RAG systems collapse this process into a single embedding space, which obscures step-wise structure. ParallaxRAG builds on the intuition that each reasoning hop should be represented from a distinct perspective. By decomposing both the query and the graph into multiple, aligned views, and regulating their interaction through a balanced exploration–exploitation process, ParallaxRAG retrieves compact, interpretable subgraphs that better support multi-hop reasoning.

3.1 Symmetric Decoupling of Query and Graph Views

In standard KG-RAG, a complex query is encoded as a single vector $\mathbf{q} \in \mathbb{R}^d$, conflating semantics required for different reasoning stages. We instead leverage the internal multi-head structure of a Transformer to obtain H specialized query views:

$$Q^{\text{views}} = \{\mathbf{q}_k \in \mathbb{R}^{d_h}\}_{k=1}^{H}. \quad (1)$$

Concretely, we extract $\mathbf{q}_k$ from the `[CLS]` representation of the final Transformer layer: the H attention heads naturally partition the hidden state into H complementary subspaces of dimension

$d_h = d/H$, each capturing a distinct semantic facet of the query. A shared linear projection W_{proj} maps each head slice to the same d_h-dimensional space and is applied symmetrically to entity and relation texts, yielding $\mathbf{e}_k, \mathbf{r}_k \in \mathbb{R}^{d_h}$. This design aligns the query and the graph across multiple semantic dimensions, allowing each head to focus on a particular aspect such as entity grounding or relational chaining. Unlike MRAG (Besta et al., 2024), which primarily leverages multiple heads as parallel embedding channels for multi-aspect retrieval, our symmetric decomposition explicitly enforces alignment between query and graphs within each head. This design encourages each head to specialize in a coherent semantic role shared by both the query and the graph, facilitating more structured hop-wise reasoning.

3.2 Balancing Exploration and Exploitation

Creating multiple semantic views introduces flexibility but also tension: the model must encourage heads to explore distinct reasoning cues, yet prevent them from drifting toward irrelevant or redundant information. ParallaxRAG balances them through *exploration* for head diversity and *exploitation* for query relevance.

Exploration: Pairwise Similarity Regulation. To encode structural context, we adopt Distance Encoding (DE) (details in A.1), which propagates signals from the topic entity via L_f forward and L_r reverse message-passing layers, yielding per-head structural features δ_k. At each DE layer ℓ, head k integrates neighborhood context via:

$$\tilde{\mathbf{F}}_k^{(\ell+1)}(v_i) = \sum_{v_j \in \mathcal{N}(v_i)} \frac{1}{|\mathcal{N}(v_i)|} \mathbf{F}_k^{(\ell)}(v_j) W^{(\ell)}. \quad (2)$$

While this process spreads information across entities, multiple heads could converge to similar activation patterns. To maintain diversity, PSR measures the overlap among heads via their activation summaries:

$$\mathbf{s}_k^{(\ell)} = \frac{\tilde{\mathbf{F}}_k^{(\ell+1)} \mathbf{1}_d}{\|\tilde{\mathbf{F}}_k^{(\ell+1)} \mathbf{1}_d\|_2}, \qquad r_k^{(\ell)} = \sum_{j \neq k} \langle \mathbf{s}_k^{(\ell)}, \mathbf{s}_j^{(\ell)} \rangle. \quad (3)$$

Each head's update is then adaptively scaled by a diversity coefficient:

$$\mathbf{F}_k^{(\ell+1)} = e^{-\beta r_k^{(\ell)}} \tilde{\mathbf{F}}_k^{(\ell+1)}, \quad (4)$$

where β controls the penalty strength. This keeps the retrieval process exploratory, where heads remain distinct, each probing a different facet of the graph.

Head-Specific Triple Scoring. Under each head k, a candidate triple $\tau = (h, r, t)$ is scored by a shared MLP over the concatenation of the query view, augmented entity, and relation representations, where $\tilde{\mathbf{e}}_k = [\mathbf{e}_k;\, \delta_k]$ denotes the entity embedding concatenated with its DE structural features, and superscripts h and t refer to the head and tail entity of τ respectively:

$$z_k(\tau) = \text{MLP}\big([\mathbf{q}_k;\, \tilde{\mathbf{e}}_k^h;\, \mathbf{r}_k;\, \tilde{\mathbf{e}}_k^t]\big). \quad (5)$$

Stacking scores across all triples and heads yields $Z \in \mathbb{R}^{|\mathcal{E}| \times H}$, which the query-aware gate then aggregates into a final ranking.

Exploitation: Query-Aware Gating with Weak Supervision. To focus retrieval on the most relevant heads for a given question, a lightweight gate maps the global query embedding $\mathbf{q}$ to head-importance weights

$$\boldsymbol{\alpha} = \text{softmax}(W_g \mathbf{q}) \in \mathbb{R}^H. \quad (6)$$

Each head's shared MLP scores candidate triples via Eq. equation 5, yielding $Z \in \mathbb{R}^{|\mathcal{E}| \times H}$; gated aggregation produces

$$\mathbf{s} = Z\boldsymbol{\alpha}, \qquad P_{\text{pred}} = \text{softmax}(\mathbf{s}). \quad (7)$$

For each question, we extract the shortest paths linking its topic and answer entities in the KG. Triples on these paths ($\mathcal{T}_{\text{sp}}$) form the weak supervision signal, defining a normalized target distribution:

$$P_{\text{true}}(\tau) = \frac{\mathbb{I}[\tau \in \mathcal{T}_{\text{sp}}]\, w_\tau}{\sum_{\tau' \in \mathcal{E}} \mathbb{I}[\tau' \in \mathcal{T}_{\text{sp}}]\, w_{\tau'}}. \quad (8)$$

The retriever is trained end-to-end by minimizing

$$\mathcal{L} = \text{KL}(P_{\text{true}} \| P_{\text{pred}}), \quad (9)$$

which guides the gate and heads to emphasize triples composing the shortest topic–answer reasoning chains.

3.3 Grounded Reasoning with LLMs

The top-k retrieved triples are linearized and concatenated with the question, together with a one-shot demonstration, to construct an evidence-grounded prompt for the LLM. This prompt formulation provides explicit relational context for each

Table 1: Main retrieval recall evaluation results for different models on WebQSP and CWQ datasets.

Model	Shortest Path Triple Recall					GPT-4o Triple Recall					Answer Entity Recall				
	WebQSP		CWQ			WebQSP		CWQ			WebQSP		CWQ		
	1 (65.8%)	2 (34.2%)	1 (28.0%)	2 (65.9%)	≥ 3 (6.1%)	1 (65.8%)	2 (34.2%)	1 (28.0%)	2 (65.9%)	≥ 3 (6.1%)	1 (65.8%)	2 (34.2%)	1 (28.0%)	2 (65.9%)	≥ 3 (6.1%)
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
SR+NSM w E2E	0.565	0.324	-	-	-	0.580	0.376	-	-	-	0.916	0.301	-	-	-
Retrieve-Rewrite-Answer	0.064	0.046	-	-	-	0.062	0.061	-	-	-	0.745	0.729	-	-	-
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	0.954	0.720	0.845	**0.826**	**0.609**	0.895	0.768	0.787	0.810	0.725	0.979	0.844	0.937	0.918	0.683
ParallaxRAG (Decoupled + Gated)	0.963	0.756	0.891	0.809	0.568	0.916	0.820	0.829	0.842	0.759	0.976	0.884	0.958	0.928	0.753
ParallaxRAG	**0.966**	**0.761**	**0.916**	0.818	0.578	**0.923**	**0.825**	**0.847**	**0.845**	**0.760**	**0.986**	**0.899**	**0.962**	**0.935**	0.771

reasoning step and ensures that the model's generation remains aligned with hop-specific evidence derived from the multi-view retriever. Details shows in Appendix B.3.

4 Experiments

We design a series of experiments to examine whether multi-view decoupling leads to more reliable and interpretable multi-hop reasoning. Specifically, we investigate three key questions: **RQ1:** Does ParallaxRAG retrieve cleaner and more relevant subgraphs efficiently? **RQ2:** Do multi-head views specialize in different reasoning stages and improve retrieval quality? **RQ3:** Can ParallaxRAG enhance answer grounding and reduce hallucination in end-to-end KGQA? Sections 4.2, 4.4 and 4.5 address RQ1, Section 4.3 analyzes RQ2, and Section 4.4 further studies RQ3.

4.1 Experiment Setup

Datasets. We use WebQSP (Yih et al., 2016) and CWQ (Talmor and Berant, 2018) as benchmarks and further test cross-domain generalization on BioASQ (Tsatsaronis et al., 2015), which requires biomedical factual reasoning.

Baselines.

Retrieval Baselines. We compare against representative retrieval paradigms, structure-free (Cosine Similarity (Li et al., 2024)), path-based (SR+NSM (Zhang et al., 2022), Retrieve-Rewrite-Answer (Wu et al., 2023), RoG (Luo et al., 2023)), hybrid (G-Retriever (He et al., 2024)), and GNN-based (GNN-RAG (Mavromatis and Karypis, 2024)), spanning a broad spectrum from flat vector matching to graph-structured reasoning.

End-to-End KGQA Baselines. We further evaluate against state-of-the-art KGQA models—UniKGQA (Jiang et al., 2022), KD-CoT (Zhao et al., 2024), StructGPT (Jiang et al., 2023), ToG (Sun et al., 2023), EtD (Liu et al., 2024), SubgraphRAG(Zhang et al., 2022), REL-RAG(Yao et al., 2025) and GraphRAG-FI(Guo et al., 2025) Following (Zhang et al., 2022), we adopt the RoG-Sep variant to avoid label leakage in RoG-Joint training.

Implementation Details. All retrievers use the `BGE-M3`[1] encoder for consistency. Baselines are reproduced from official implementations. Key hyperparameters and training details are listed in Appendix B. Results using alternative backbones are shown in Appendix C.3.

Metrics. We report both retrieval and end-to-end performance. *Retrieval:* (i) shortest-path triple recall, (ii) GPT-4o–verified triple recall (validating the correctness of recalled triples), and (iii) answer-entity recall, averaged per query and broken down by hop length, we report two ParallaxRAG configurations, one with the decoupled-and-gated architecture alone and one with additional PSR, to disentangle the structural contribution from regularization effects. *Efficiency:* wall-clock inference time on a 48GB RTX 6000 Ada GPU (excluding KG I/O). *End-to-end QA:* Macro-F1 and Hit scores on WebQSP/CWQ, plus domain metrics on BioASQ.

4.2 Retrieval Performance (RQ1)

Notably, even without PSR, the decoupled-and-gated ParallaxRAG configuration already outperforms almost all baseline retrievers across both datasets, especially on Answer Entity Recall, indicating that the core multi-view architecture itself provides a strong inductive bias for multi-hop retrieval. Pairwise Similarity Regulation (PSR) further strengthens this effect, particularly on longer-hop queries, by preventing head collapse and im-

[1] https://huggingface.co/BAAI/bge-m3

proving robustness under combinatorial expansion.

In terms of efficiency, ParallaxRAG remains computationally lightweight, requiring 40 seconds on WebQSP and 84 seconds on CWQ (excluding KG I/O), compared to 948 and 2327 seconds for RoG, 672 and 1530 seconds for G-Retriever.

Table 2: Question-answering performance on WebQSP and CWQ. The Hallucination (Hallu.) score is evaluated on a subset that excludes samples where the answer entity is not in the KG, following (Li et al., 2024). Our generaters use the top 100 retrieved triples by default; results for 200 and 500 (indicated in parentheses) are also shown. Best results are in **bold**. Results with (↔) evaluate retriever generalizability, where the retriever is trained on one dataset and applied to the other.

	WebQSP			CWQ		
Model	**Macro-F1**	**Hit**	**Hallu.**	**Macro-F1**	**Hit**	**Hallu.**
(A) Neural Methods						
UniKGQA	72.2	-	-	49.0	-	-
SR+NSM w E2E	64.1	-	64.44	46.3	-	-
(B) Multi-turn LLM Reasoning Methods						
KD-CoT	52.5	68.6	-	-	55.7	-
ToG (GPT-4)	-	82.6	-	-	67.6	-
StructGPT	-	74.69	-	-	-	-
Retrieve-Rewrite-Answer	-	79.36	-	-	-	-
RoG-Joint	70.26	86.67	76.13	54.63	61.94	55.15
RoG-Sep	66.45	82.19	72.79	53.87	60.55	54.51
RoG + GraphRAG-FI	73.86	89.25	-	55.12	64.82	-
(C) RAG-based Methods						
G-Retriever	53.41	73.46	67.97	-	-	-
EtD	-	82.5	-	-	62.0	-
GNN-RAG	71.3	85.7	-	59.4	66.8	-
SubgraphRAG + GPT-4o	76.46	89.80	81.85	59.08	66.69	66.57
REL-RAG + GPT-4o-mini	78.7	92.5	-	58.6	68.3	-
(D) Ours Methods						
ParallaxRAG + Llama3.1-8B	71.73	86.85	**83.64**	48.33	58.41	66.12
ParallaxRAG + Qwen3-30B	75.24	91.60	75.45	59.25	65.30	57.18
ParallaxRAG + Qwen3-30B (200)	76.07	92.48	76.23	59.31	66.21	58.06
ParallaxRAG + Qwen3-30B (500)	76.11	93.26	77.86	59.18	64.52	60.07
ParallaxRAG + GPT-4o (200)	**78.80**	**93.53**	82.94	**62.31**	**70.74**	**66.69**
(E) Cross-dataset Generalization (↔)						
SubgraphRAG + Llama3.1-8B (↔)	66.42	83.42	80.09	37.96	48.57	56.78
ParallaxRAG + Llama3.1-8B (↔)	**69.22**	**89.51**	82.64	**45.28**	**54.82**	60.04

Table 3: Answer Entity Recall under controlled disruption on CWQ, broken down by hop depth. Long-hop-associated heads are functionally non-substitutable.

	Answer Entity Recall			
Condition	**1-hop**	**2-hop**	**≥3-hop**	**Overall**
ParallaxRAG Baseline	0.959	0.926	0.763	0.883
Drop Long-Hop Heads	0.958	0.777	0.501	0.745
Shuffle Heads	0.734	0.532	0.468	0.578
All	0.734	0.446	0.421	0.534

4.3 Attention Head Specialization (RQ2)

Figure 3 reveals a clear division of labor among ParallaxRAG's attention heads that adapts to query complexity. For short-hop questions in WebQSP, early heads dominate across reasoning steps. As the reasoning chain extends in CWQ, later heads become increasingly active at deeper reasoning steps. This dynamic shift forms a relay pattern, where distinct head groups take over successively as reasoning deepens.

We quantify this specialization by training a linear probe to predict reasoning depth from head activations, achieving 52.3% accuracy compared to a 25% random baseline, demonstrating that head activations encode distinct reasoning stages. To examine the functional role of specialist heads, we perform a Difference-in-Differences-in-Differences (DDD) analysis comparing performance drops after ablating specialist versus random heads across short and long queries:

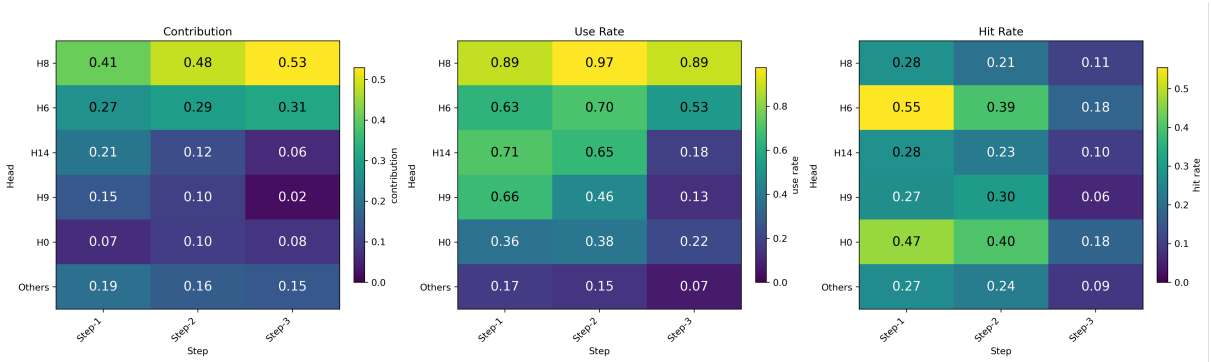
$$\text{DDD} = [(\Delta^{\text{Specialist}}_{\text{Long}} - \Delta^{\text{Specialist}}_{\text{Short}})] - [(\Delta^{\text{Random}}_{\text{Long}} - \Delta^{\text{Random}}_{\text{Short}})]$$

where Δ denotes the post-ablation performance difference. A significant negative DDD value of -0.0184 (95% CI: [-0.0248, -0.0045], p=0.0055) provides functional evidence that specialist heads are non-substitutable, playing distinct roles in multi-hop reasoning.

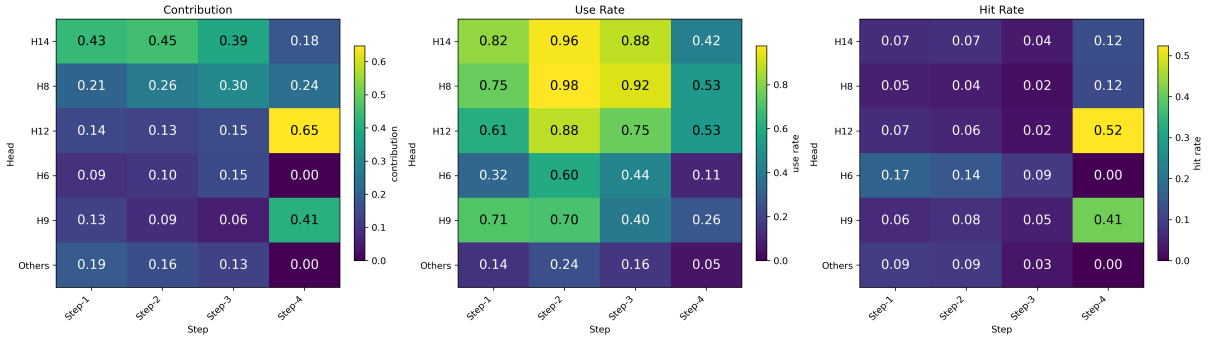
To further confirm this, we conduct controlled disruption experiments on CWQ under three conditions: **Drop Long-Hop Heads** removes the long-hop-associated head groups (8,9,12) identified from Figure 3; **Shuffle Heads** randomly permutes the head-to-query-view assignment; and **All** applies both simultaneously. As Table 3 shows, dropping long-hop heads has negligible effect on 1-hop queries but substantially degrades ≥3-hop performance by 0.262 absolute. Shuffling head alignment impairs all hop depths, with degradation scaling with query complexity. These asymmetric patterns confirm that the identified head groups play coordinated, non-interchangeable roles in multi-hop reasoning over knowledge graphs.

4.4 End-to-End KGQA with RAG (RQ1 & RQ3)

We next examine whether the retrieval and head-specialization advantages of ParallaxRAG lead to improvements in downstream question answering. As shown in Table 2, when paired with Llama3.1-8B and 100 retrieved triples, ParallaxRAG attains a Macro-F1 of 71.73 and Hit of 86.85 on WebQSP, and 48.33 Macro-F1 and 58.41 Hit on CWQ. Combined with GPT-4o and 200 triples, it establishes new state-of-the-art results, reaching 78.80 Macro-F1 and 93.53 Hit on WebQSP, and 62.31 Macro-F1 and 70.74 Hit on CWQ. The gain from a stronger generator is disproportionately larger on CWQ, suggesting that for complex multi-hop chains the generator's capacity to integrate multi-relational evi-



(a) WebQSP: Early-Step Specialization for Short-Chain Reasoning



(b) CWQ: Dynamic Head Specialization Switching for Long-Chain Reasoning

Figure 3: Visualization of attention head specialization, revealing a task-adaptive division of labor. The heatmaps display three metrics (Contribution, Use Rate, Hit Rate) for the top 5 specialist heads, selected from our BGE-based model's 16 heads according to their overall contribution. The Others row aggregates the remaining 11 heads: for Contribution, this value is the sum of their contributions, while for Use Rate and Hit Rate, it is their average, see more explanation in (Appendix C). We define *steps* as the BFS expansion depth along reasoning paths, rather than the conventional hop (shortest-path length), in order to capture head behaviors at each layer of candidate exploration. **(a)** On WebQSP (short chains). **(b)** On CWQ (long chains).

dence becomes the binding constraint once retrieval quality is sufficient. Expanding the retrieval budget from 100 to 200 triples yields moderate gains on both datasets, while further expansion to 500 triples provides negligible improvement on WebQSP and slightly hurts CWQ, indicating that retrieval precision rather than coverage is the key driver of downstream accuracy.

Beyond overall accuracy, ParallaxRAG improves reliability by reducing hallucinated generations. On hallucination-controlled subsets (Li et al., 2024), hallucination scores decrease by 1.22% on WebQSP and 3.23% on CWQ, indicating that the retrieved subgraphs are not only sufficient but also cleaner and more grounded. Qualitative examples in Appendix E further illustrate this effect: for long-chain CWQ questions, ParallaxRAG retrieves fewer but more relevant triples, enabling coherent reasoning chains that accurately support the final answers. To assess reproducibility, we conducted five independent runs and observed stable performance (standard deviation below 1.4 Macro-F1 points). Improvements over strong baselines are statistically significant via paired bootstrap resampling ($p < 0.05$); full details are reported in Appendix D.1.

4.5 Domain Generalization (RQ1)

To evaluate cross-domain robustness, we test ParallaxRAG on the biomedical BioASQ benchmark (Task B), which covers Yes/No, Factoid, and List questions. The retriever is trained on CWQ and transferred without retraining. We use Qwen3-Plus as the generator and report official metrics for each question type.

As shown in Table 4, ParallaxRAG achieves the best Yes/No results, with an accuracy of 0.9351 and a Macro-F1 of 0.9252, outperforming strong general-purpose LLMs. On Factoid questions, it attains a lenient accuracy of 0.8222 and an MRR of 0.8667, reflecting effective multi-view reasoning even under domain shift. For List questions,

Table 4: Generalization on BioASQ Task B, where the retriever is trained on CWQ and transferred without fine-tuning. We compare the performance of Qwen3-Plus, Single-View and ParallaxRAG (Multi-View), isolating the contribution of the retrieval design

Model	Yes/No Questions				Factoid Questions			List Questions		
	Acc.	F1 Yes	F1 No	Macro F1	Strict	Lenient	MRR	Prec.	Recall	F1
Deepseek-R1-8B	0.9077	0.9312	0.8558	0.8935	0.2620	0.2733	0.2677	0.3982	0.3427	0.3517
GPT4-Turbo	0.9129	0.9357	0.8616	0.8986	**0.5505**	0.6515	0.6010	0.5788	**0.4857**	**0.5051**
GPT-4o	0.9140	0.9347	0.8670	0.9009	0.3462	0.3462	0.3462	0.5102	0.4025	0.4330
Qwen3-Plus	0.9012	0.9255	0.8402	0.8829	0.3219	0.6213	0.4132	0.5682	0.3989	0.4563
Single-View + Qwen3-Plus	0.9112	0.9267	0.8374	0.8821	0.3773	0.7214	0.8462	0.6833	0.4280	0.4464
ParallaxRAG + Qwen3-Plus	**0.9351**	**0.9524**	**0.8980**	**0.9252**	0.4210	**0.8222**	**0.8667**	**0.7116**	0.4569	0.4737

Table 5: Ablation study on WebQSP and CWQ, analyzing the impact of ParallaxRAG's core multi-head architecture and its synergistic mechanisms. Values in parentheses (↓) indicate the performance drop compared to the full model (Llama3.1-8B as generator).

	Configuration	WebQSP		CWQ	
		Macro-F1	Hit	Macro-F1	Hit
	ParallaxRAG (Full Model)	**71.73**	**86.85**	**48.33**	**58.41**
	Multi-Head Architecture Validation				
(a)	Split Vector Baseline	69.42 (↓2.31)	85.02 (↓1.83)	42.27 (↓6.06)	51.54 (↓6.87)
(b)	Single Vector Baseline	70.60 (↓1.13)	86.11 (↓0.74)	44.82 (↓3.51)	52.63 (↓5.78)
	Synergistic Mechanism Validation				
(c)	Without PSR	69.73 (↓0.73)	85.47 (↓0.35)	47.12 (↓1.21)	56.32 (↓2.09)
(d)	Without Query-Aware Gating	56.25 (↓15.48)	74.69 (↓12.16)	29.14 (↓19.19)	38.26 (↓20.15)

ParallaxRAG favors precision over recall, yielding a slightly lower F1 than GPT-4-Turbo, which suggests a bias toward concise, high-confidence retrieval.

We further compares single-view and multi-view configurations under the same generator, prompting format, and context budget, confirming that the gains stem from the multi-view retrieval design rather than from the choice of generator.

4.6 Ablation Study

First, to verify the role of the multi-head design, we compare two simplified baselines. The Split Vector variant divides a single query embedding into pseudo-heads without true head specialization, causing a Macro-F1 drop of 2.31 on WebQSP and 6.06 on CWQ. Similarly, the Single Vector variant using a flat embedding, shows performance drops of 1.13 and 3.51 points, confirming that learned multi-head representations are crucial for capturing distinct relational cues and mitigating representational collapse.

Next, we evaluate the synergistic mechanisms. Removing Pairwise Similarity Regularization (PSR) yields redundant head behavior and reduces Macro-F1 by 0.73 and 1.21 points, demonstrating PSR's role in maintaining head diversity. Eliminating the query-aware gating and replacing it with simple averaging leads to severe performance degradation, with Hit rate drops of 12.16 and 20.15 points, underscoring its necessity for dynamic head weighting and precise retrieval.

Together, removing either component degrades performance substantially, with gating removal causing the sharpest drop (Hit $-12.16/-20.15$), indicating it is the more critical of the two.

5 Conclusion

We propose ParallaxRAG, a multi-view KG-RAG framework that explicitly leverages attention head specialization for multi-hop reasoning. By decoupling queries and knowledge graphs into aligned head-specific semantic spaces and combining diversity regularization with query-aware aggregation, ParallaxRAG retrieves accurate and clean subgraphs while remaining efficient. Experiments demonstrate state-of-the-art retrieval and QA performance on WebQSP and CWQ, reduced hallucination, and zero-shot generalization to the biomedical BioASQ benchmark. Beyond performance gains, our analysis provides functional evidence suggesting that attention heads tend to specialize in

different stages of multi-hop reasoning and play coordinated, non-substitutable roles within the structured multi-view framework. These results suggest that modeling head-level semantic specialization is a practical and generalizable direction for retrieval-augmented multi-hop reasoning over KGs.

6 Limitations and Future Work

While ParallaxRAG demonstrates strong retrieval performance and effectively captures head-level specialization, we identify several limitations that suggest directions for future work.

First, our method relies on the quality of the underlying knowledge graph and embedding model; errors in these components may propagate to retrieval performance. In addition, the current implementation focuses on static knowledge graphs and does not address dynamic or continuously updated settings. We leave extensions to noisier and larger-scale knowledge sources as future work.

Second, our training relies on shortest-path triples as weak supervision signals, following established practice in prior work (Luo et al., 2023). While shortest paths are a practical and scalable source of supervision, they may not always represent the optimal reasoning route, and this choice may introduce a bias toward shortest-path evidence. That said, our primary evaluation metric, Answer Entity Recall, requires only that the gold answer entity appear in the retrieved subgraph regardless of the specific path taken, which partially mitigates this concern. Developing retrieval supervision that incorporates diverse, multi-path annotations remains an important direction for future work.

Third, as illustrated by our case study, the benefits of multi-view retrieval are most pronounced for complex multi-hop queries, while they may be less critical for shallow queries (e.g., 1-hop transitive relations). In such cases, retrieving a richer set of evidence across multiple views can introduce contextual signals that are not strictly required for answering the question. This may increase the burden on downstream LLMs to reconcile partially redundant information, occasionally resulting in conservative predictions. This observation aligns with the observation proposed by (Guo et al., 2025). A direction for future work is to develop adaptive retrieval strategies that dynamically modulate retrieval breadth based on estimated query complexity.

Acknowledgments

We thank the anonymous reviewers for their insightful comments and suggestions. This work was supported by the National Natural Science Foundation of China (Grant NO. 62576292), the Zhejiang Province Leading Geese Plan (2025C02025), the Science and Technology Program of Huzhou (Grant NOs. 2023GZ42 and 2024GZ09), and in part by the Yangtze Delta Region Institute (Huzhou) Guidance Fund of University of Electronic Science and Technology of China (Grant NO. U03210054).

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection.

Lorenzo Basile, Valentino Maiorca, Diego Doimo, Francesco Locatello, and Alberto Cazzaniga. 2025. Head pursuit: Probing attention specialization in multimodal transformers. In *Advances in Neural Information Processing Systems*, volume 38. Spotlight.

Maciej Besta, Ales Kubicek, Robert Gerstenberger, Marcin Chrapek, Roman Niggli, Patrik Okanovic, Yi Zhu, Patrick Iff, Michal Podstawski, Lucas Weitzendorf, and 1 others. 2024. Multi-head rag: Solving multi-aspect problems with llms. *arXiv preprint arXiv:2406.05085*.

Srinadh Bhojanapalli, Ayan Chakrabarti, Andreas Veit, Michal Lukasik, Himanshu Jain, Frederick Liu, Yin-Wen Chang, and Sanjiv Kumar. 2021. Leveraging redundancy in attention with reuse transformers. *arXiv preprint arXiv:2110.06821*.

Yuchen Bian, Jiaji Huang, Xingyu Cai, Jiahong Yuan, and Kenneth Church. 2021. On attention redundancy: A comprehensive study. In *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: human language technologies*, pages 930–945.

Bin Fu, Yunqi Qiu, Chengguang Tang, Yang Li, Haiyang Yu, and Jian Sun. 2020. A survey on complex question answering over knowledge base: Recent advances and challenges. *arXiv preprint arXiv:2007.13069*.

Ruiliu Fu, Han Wang, Xuejun Zhang, Jun Zhou, and Yonghong Yan. 2021. Decomposing complex questions makes multi-hop qa easier and more interpretable. *arXiv preprint arXiv:2110.13472*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1).

Kai Guo, Harry Shomer, Shenglai Zeng, Haoyu Han, Yu Wang, and Jiliang Tang. 2025. Empowering GraphRAG with knowledge filtering and integration. In *Proceedings of EMNLP 2025*.

Haoyu Han, Kai Guo, Harry Shomer, Yu Wang, Yucheng Chu, Hang Li, Li Ma, and Jiliang Tang. 2026. Reasoning by exploration: A unified approach to retrieval and generation over graphs. In *Proceedings of the ACM Web Conference 2026*, WWW '26.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *Advances in Neural Information Processing Systems*, 37:132876–132907.

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*, 1(2):4.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S Yu. 2021. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*, 33(2):494–514.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Structgpt: A general framework for large language model to reason over structured data. *arXiv preprint arXiv:2305.09645*.

Jinhao Jiang, Kun Zhou, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Unikgqa: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph. *arXiv preprint arXiv:2212.00959*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Mufei Li, Siqi Miao, and Pan Li. 2024. Simple is effective: The roles of graphs and large language models in knowledge-graph-based retrieval-augmented generation. *arXiv preprint arXiv:2410.20724*.

Yuejie Li, Ke Yang, Tao Wang, Bolin Chen, Bowen Li, and Chengjun Mao. 2026. Deep GraphRAG: A balanced approach to hierarchical retrieval and adaptive integration. *arXiv preprint arXiv:2601.11144*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988.

Guangyi Liu, Yongqi Zhang, Yong Li, and Quanming Yao. 2024. Explore then determine: A gnn-llm synergy framework for reasoning over knowledge graph. *arXiv e-prints*, pages arXiv–2406.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2023. Reasoning on graphs: Faithful and interpretable large language model reasoning. *arXiv preprint arXiv:2310.01061*.

Costas Mavromatis and George Karypis. 2024. Gnn-rag: Graph neural retrieval for large language model reasoning. *arXiv preprint arXiv:2405.20139*.

Geondo Park, Chihye Han, Wonjun Yoon, and Daeshik Kim. 2020. Mhsan: Multi-head self-attention network for visual semantic embedding. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1518–1526.

Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. 2024. Graph retrieval-augmented generation: A survey. *arXiv preprint arXiv:2408.08921*.

Ethan Perez, Patrick Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. 2020. Unsupervised question decomposition for question answering. *arXiv preprint arXiv:2002.09758*.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, and 1 others. 2020. Kilt: a benchmark for knowledge intensive language tasks. *arXiv preprint arXiv:2009.02252*.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M Ni, Heung-Yeung Shum, and Jian Guo. 2023. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. *arXiv preprint arXiv:2307.07697*.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, and 1 others. 2015. An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 16(1):138.

Varun Vashisht, Samar Singh, Mihir Konduskar, Jaskaran Singh Walia, and Vukosi Marivate. 2025. Mage: Multi-head attention guided embeddings for low resource sentiment classification. *arXiv preprint arXiv:2502.17987*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yike Wu, Nan Hu, Sheng Bi, Guilin Qi, Jie Ren, Anhuan Xie, and Wei Song. 2023. Retrieve-rewrite-answer: A kg-to-text enhanced llms framework for knowledge graph question answering. *arXiv preprint arXiv:2309.11206*.

Tianjun Yao, Haoxuan Li, Zhiqiang Shen, Pan Li, Tongliang Liu, and Kun Zhang. 2025. Learning efficient and generalizable graph retriever for knowledge-graph question answering. *Preprint*, arXiv:2506.09645.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2016. Value-based neighborhood expansion: A survey. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 1780–1786.

Jing Zhang, Xiaokang Zhang, Jifan Yu, Jian Tang, Jie Tang, Cuiping Li, and Hong Chen. 2022. Subgraph retrieval enhanced model for multi-hop knowledge base question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5773–5784.

Ruilin Zhao, Feng Zhao, Long Wang, Xianzhi Wang, and Guandong Xu. 2024. Kg-cot: Chain-of-thought prompting of large language models over knowledge graphs for knowledge-aware question answering. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24)*, pages 6642–6650. International Joint Conferences on Artificial Intelligence.

Zifan Zheng, Yezhaohui Wang, Yuxin Huang, Shichao Song, Mingchuan Yang, Bo Tang, Feiyu Xiong, and Zhiyu Li. 2025. Attention heads of large language models: a survey. *Patterns*, 6(2):101176.

A Implementation Details of the ParallaxRAG Framework

A.1 Detailed Derivation of Pairwise Similarity Regulation (PSR)

The Pairwise Similarity Regulation (PSR) mechanism is integrated into each layer of the Distance Encoding (DE) to foster representational diversity.

1. DE Propagation. Given the initial one-hot topic entity features $\mathbf{X}_0 \in \mathbb{R}^{N\times 2}$, the model performs L_f forward and L_r reverse propagation layers. Each layer ℓ computes a preliminary update $\tilde{\mathbf{F}}_k^{(\ell+1)}$ for each head k via message passing (MP):

$$\tilde{\mathbf{F}}_k^{(\ell+1)} = \mathrm{MP}(\mathbf{E}, \mathbf{F}_k^{(\ell)}) \quad \text{(forward pass, using edge index } \mathbf{E}\text{)} \tag{10}$$

The reverse pass is analogous, using the transposed edge index $\mathbf{E}^T$.

2. PSR Computation and Application. After each propagation step, the preliminary updates $\tilde{\mathbf{H}}$ are modulated by PSR. The process begins by computing a node intensity vector $\mathbf{s}_k^{(\ell)}$ for each head to capture its activation distribution:

$$\mathbf{s}_k^{(\ell)} = \mathrm{L2Norm}\left(\sum_d \tilde{\mathbf{F}}_k^{(\ell)}[:, d]\right) \tag{11}$$

These vectors are then used to derive a redundancy score $r_k^{(\ell)}$ from pairwise similarities (cf. Eq. equation 3), which in turn defines the regulation coefficient $\alpha_k^{(\ell)}$:

$$\begin{aligned} r_k^{(\ell)} &= \sum_{j\neq k} \langle \mathbf{s}_i^{(\ell)}, \mathbf{s}_j^{(\ell)} \rangle \\ \alpha_k^{(\ell)} &= \exp(-\beta \cdot r_k^{(\ell)}) \end{aligned} \tag{12}$$

Finally, this coefficient performs the final update by scaling the preliminary update to produce the final layer output:

$$\mathbf{F}_k^{(\ell+1)} = \alpha_k^{(\ell)} \cdot \tilde{\mathbf{F}}_k^{(\ell+1)} \tag{13}$$

3. Final Representation. The final structural representations $\mathcal{F}_k$ for each head are formed by collecting the outputs $\{\mathbf{F}_k^{(\ell)}\}$ from all layers, which are then used for triple scoring.

A.2 Weighted Listwise Training Objective

To handle the severe class imbalance in retrieval, we convert the binary weak supervision signal $\mathbf{y} \in \{0,1\}^{|\mathcal{E}|}$ into a weighted target distribution for our listwise objective.

1. Positive Reweighting. The binary vector $\mathbf{y}$ is first normalized. Then, a weight factor w_τ is applied to construct the final weighted distribution. In our implementation, we use $w_\tau = 10$ for positive triples ($y_\tau = 1$) and $w_\tau = 1$ otherwise. This is conceptually similar to the α-balancing in Focal Loss (Lin et al., 2017).

$$y_\tau^{\text{norm}} = \frac{y_\tau}{\sum_{\tau'} y_{\tau'}}, \quad y_\tau^{\text{weighted}} = \frac{y_\tau \cdot w_\tau}{\sum_{\tau'} (y_{\tau'} \cdot w_{\tau'})} \tag{14}$$

2. Final Loss Formulation. The model is trained by minimizing the weighted listwise cross-entropy (equivalent to Eq. equation 9) between the predicted distribution P_{pred} and the target y^{weighted}:

$$\mathcal{L}_{\text{listwise}} = -\sum_{\tau \in \mathcal{E}} y_\tau^{\text{weighted}} \log \left(P_{\text{pred}}(\tau) + \epsilon\right) \tag{15}$$

B Additional Experiment Setting

B.1 Implementation Details for ParallaxRAG

Our model is implemented in PyTorch and PyTorch Geometric. The ParallaxRAG retriever, which consists of a `BGE-M3` text encoder, a PSR-enhanced DE module, and a scoring MLP, is trained end-to-end for up to 100 epochs with an early stopping patience of 20. The forward and reverse propagation round is set to 2 rounds, with PSR strength of 0.5. We use the AdamW optimizer with a half-cycle cosine annealing with warmup learning rate, during a warmup phase for the first few epochs, the learning rate linearly increases from 0 to a peak value of 1×10^{-3}. Following the warmup, the learning rate is smoothly decayed along a cosine curve to a minimum value of 1×10^{-5}. to enable finer parameter adjustments in the later stages of training. An effective batch size of 2 (via gradient accumulation), and employ a weighted listwise ranking loss.

For the end-to-end KGQA task, the top-100 triples retrieved are linearized into a natural text format and prepended to the question as context for a LLM generator. We generate final answers using nucleus sampling with $p = 0.95$ and a temperature of 0.7.

B.2 GPT-4o Triple Recall Verification Protocol

In the main retrieval evaluation, GPT-4o is used solely as a structured extraction tool to identify which retrieved triples are necessary for answering a given question (GPT-4o-verified Triple Recall, $\mathcal{R}_{\text{gpt}}$). It is *not* used as an answer generator or for any open-ended evaluation. The exact prompt used for supporting-triple extraction is as follows.

Prompt format used for GPT-4o triple-recall verification.

```
System Prompt: Based on the triplets retrieved from a knowledge graph, please select the relevant triplets necessary for answering the question. Return the selected triplets as a list, each prefixed with "evidence:".

User: [Retrieved triples] [Question from the dataset]
```

After extraction, triple recall is computed deterministically via scripted string matching against annotated ground truth, so the final metric does not depend on GPT-4o's generative tendencies. To assess extraction stability, we repeated the extraction with three different models (GPT-4o, Qwen3-30B, GLM-4.7) and observed negligible variance (std $\leq 1.5\%$). We also manually inspected a random 10% subset and found high consistency between automated extraction and human judgment.

B.3 Prompt for Answer Generation

The detailed prompt template used in our experiments for answer generation is as follows.

B.4 Hyperparameter Sensitivity Analysis

In this section, we analyze the sensitivity of our model to the key hyperparameter β, which controls the strength of the Pairwise Similarity Regulation (PSR). We argue that a challenging zero-shot transfer task is the most effective setting to demonstrate PSR's true diversity impact, as it enables the model to maximally leverage its learned head specialization. Therefore, we evaluate a model trained on WebQSP on the CWQ test set, varying β within the range of $\{0.2, 0.5, 2.0\}$, with $\beta = 0$ serving as the baseline without diversity regulation.

Prompt format used for QA.

System Prompt

```
Based on the triplets from a
knowledge graph, please answer the
given question.  Please keep the
answers as simple as possible and
return all the possible answers as
a list, each with a prefix "ans:".
```

In-Context Learning (Few-shot) Examples

```
Triplets:
(Lou Seal, sports.mascot.team, San
Francisco Giants)
(San Francisco Giants,
sports.sports team.championships,
2012 World Series)
(San Francisco Giants,
sports.sports championship
event.champion, 2014 World Series)
(San Francisco Giants,
time.participant.event, 2014 Major
League Baseball season)
...
Question:
What year did the team with mascot
named Lou Seal win the World
Series?
To find the year ......  Therefore,
the formatted answers are:
ans:  2014 (2014 World Series)
ans:  2012 (2012 World Series)
ans:  2010 (2010 World Series)
```

User Prompt

```
Triplets:
(e_a, r_ab, e_b),
(e_c, r_cd, e_d),
...
Question:
What ...?
```

As shown in Figure 4, the results under this stringent evaluation setting are revealing. A moderate PSR strength of $\beta = 0.5$ achieves the peak performance, boosting the Macro-F1 from a baseline of 50.54 to 50.93 and the Hit rate from 61.87 to 62.51. This performance gain suggests that the diverse representations fostered by PSR are indeed learning more generalizable reasoning mechanisms. The degradation at a high penalty ($\beta = 2.0$) indicates the limit of this effect, where excessive suppression can hinder signal propagation. These findings validate our choice of $\beta = 0.5$ and confirm that the generalization setting is an effective testbed for evaluating the impact of PSR.

C Attention Head Analysis Details

This section provides the detailed methodology for the attention head analysis presented in Section 4.3.

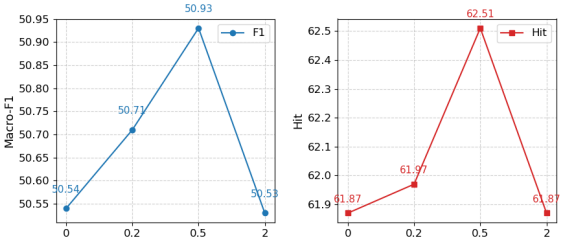


Figure 4: Sensitivity analysis of the PSR strength parameter β on the generalization task, where the retriever was trained on WebQSP and tested on CWQ-sub, using Llama-3.1-8B as the generator.

We formally define the concepts of reasoning *steps* and the three metrics used to evaluate head-level performance.

C.1 Defining Reasoning Steps

To formally quantify head performance during multi-hop reasoning, we define *steps* as the Breadth-First Search (BFS) expansion depth along reasoning paths. This is distinct from the conventional definition of a *hop* (i.e., the final shortest-path length) and allows us to capture head behaviors at each layer of candidate exploration. For example, if the topic entity is "Barack Obama," Step-1 includes triples directly connected to him, such as $(\text{Obama}, \text{born\_in}, \text{Honolulu})$. Step-2 then expands from the entities retrieved in Step-1, considering candidates like $(\text{Honolulu}, \text{located\_in}, \text{Hawaii})$.

C.2 Head Performance Metrics

We define three complementary metrics to evaluate the role and effectiveness of each attention head during the reasoning process.

Contribution Measures a head's share of credit for retrieving correct triples at a given step. It is defined as the proportion of correctly retrieved triples for which this head provided the highest score.

$$\text{Contribution}(h,t) = \frac{|C(h,t) \cap A_t \cap G_t|}{|A_t \cap G_t|} \tag{16}$$

Use Rate Measures the model's overall reliance on a head. It is the proportion of samples where a triple scored highest by head h was ultimately selected by the model's gating mechanism at step t.

$$\text{Use Rate}(h,t) = \frac{|\mathcal{S}(h,t)|}{|\mathcal{S}_t|} \tag{17}$$

Hit Rate Measures the precision of a head's top-scoring suggestions. It is the percentage of a head's suggestions selected by the gate that are actually correct.

$$\text{Hit Rate}(h,t) = \frac{|C(h,t) \cap A_t \cap G_t|}{|C(h,t) \cap A_t|} \quad (18)$$

where, for any given head h and reasoning step t, G_t is the set of ground-truth triples, and A_t is the set of triples actually retrieved by the gated model. The term $C(h,t)$ represents the set of candidate triples for which head h provided the highest score among all heads. Finally, $\mathcal{S}_t$ is the set of all samples that require reasoning at step t, while $\mathcal{S}(h,t)$ is the subset of those samples where a triple from $C(h,t)$ was selected by the final gating mechanism.

C.3 Head Specialization with Alternative Backbone Encoders

To confirm the generalizability of the head specialization phenomenon, we tested the ParallaxRAG framework using two alternative, high-performance backbone encoders: `intfloat/e5-large-v2` and `thenlper/gte-large`. Our analysis confirms that the core phenomenon of head specialization persists across all tested encoders, though the specific cooperative patterns among heads vary, reflecting different emergent strategies for multi-hop reasoning.

C.3.1 `intfloat/e5-large-v2` Analysis (Figure 5a)

The `e5-large-v2` encoder consistently validates the functional division of labor. Head 10 (Contribution: 0.60, Use Rate: 0.96) and Head 2 (Contribution: 0.52, Use Rate: 0.94) emerged as the dominant initial-step specialists (peaking at Step-1 for entity localization). Conversely, Head 3 demonstrated the highest precision for terminal reasoning, achieving a Hit Rate of 0.50 at Step-4. This confirms the learned segregation between high-activation front-end heads and high-precision terminal heads. However, the magnitude of the dynamic switching effect was marginally less pronounced compared to BGE-M3, suggesting a more continuous contribution profile.

C.3.2 `thenlper/gte-large` Analysis (Figure 5b)

The `gte-large` results offer a clearer and more compelling validation of dynamic head specialization switching. Head 10 served as the definitive initial-step specialist (Contribution: 0.43, Use Rate: 0.85 at Step-1). Crucially, Head 6 and Head 12 collectively assumed the role of long-range dependency specialists in later stages. Head 6's contribution increased significantly at Step-4 (from 0.13 to 0.31), marking a clear functional transition. Most notably, Head 12 achieved a maximum Hit Rate of 0.50 at Step-4, despite minimal involvement at Step-1 (Contribution: 0.06). This sharp contrast in activation profiles unequivocally demonstrates the learned non-substitutability of specialized heads, confirming that ParallaxRAG successfully engineers a task-adaptive retrieval architecture irrespective of the foundational text encoder.

Table 6: End-to-End QA Performance with Alternative Backbone Encoders using Llama3.1-8B as KGQA generator

Encoder	WebQSP		CWQ	
	Macro-F1	Hit	Macro-F1	Hit
E5	69.23	85.63	44.59	53.95
GTE	69.75	85.92	47.12	57.48

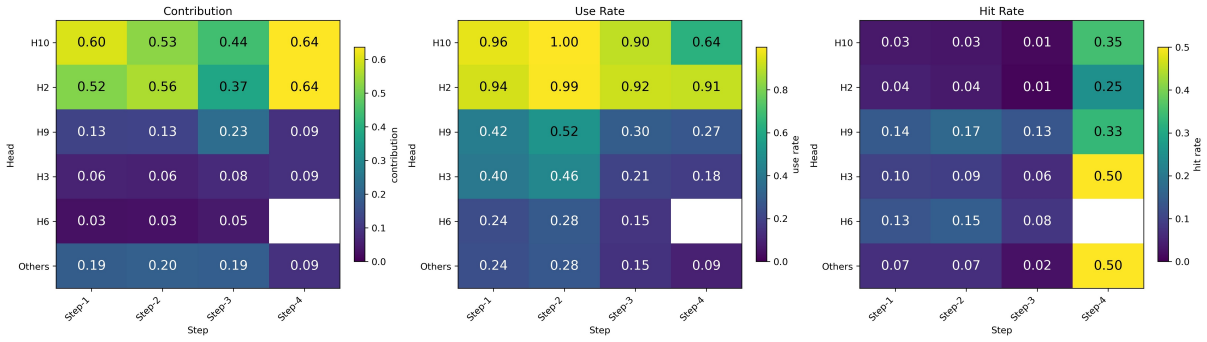
C.3.3 Correlation with Performance

The observed specialization patterns strongly correlate with end-to-end performance, particularly on the long-chain reasoning CWQ benchmark (Table 6). The GTE model, which exhibited a more pronounced dynamic specialization, significantly outperforms the E5 model on CWQ (Macro-F1: 47.12 vs. 44.59; Hit Rate: 57.48 vs. 53.95). Performance on the simpler WebQSP task is comparable (Macro-F1 difference is 0.52). This evidence supports the conclusion that the degree of functional specialization learned by the backbone encoder is a critical factor for robustness in multi-hop KGQA.

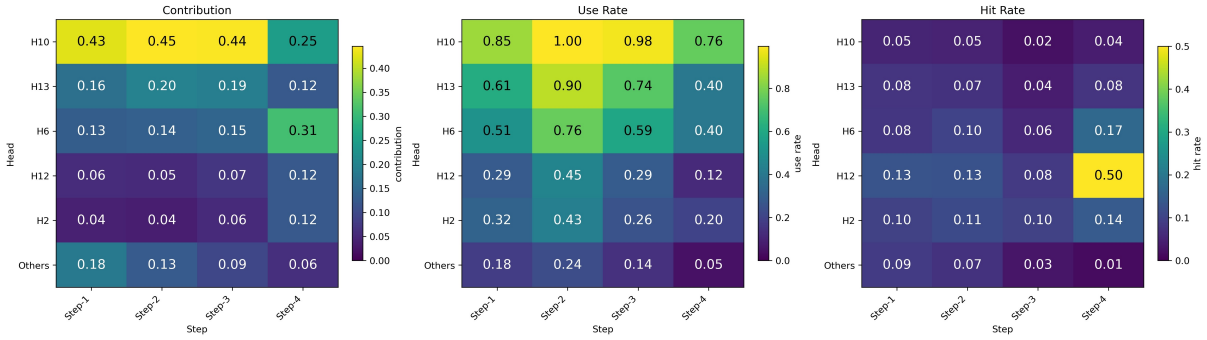
C.4 Head Specialization Verification Details

C.4.1 Linear Probing

To quantitatively verify that different attention heads specialize in distinct reasoning stages, we conducted a linear probing experiment. For each reasoning step, we aggregated the output scores from all attention heads to form a feature vector. A logistic regression classifier was then trained on these features to decode the current step number (from 1 to 4). Evaluated under 5-fold cross-validation grouped by sample, the classifier achieved a decoding accuracy that significantly outperformed a random baseline. This result confirms that the activation patterns of the attention



(a) E5-large-v2 on CWQ dataset



(b) GTE-large on CWQ dataset

Figure 5: Attention Head Specialization Analysis on CWQ Dataset with Alternative Backbone Encoders. (a) Results using the E5-large-v2 model. (b) Results using the GTE-large model.

heads contain sufficient information to distinguish between different stages of the reasoning process, supporting the hypothesis of functional specialization.

C.4.2 Difference in Difference in Difference

To causally validate the functional importance of specialist heads, we designed a Triple-Difference (DDD) intervention. This analysis isolates the additional performance degradation from ablating specialist heads versus random heads on long-hop ($>$ 1-hop) questions, relative to short-hop (1-hop) questions. We first compute the Difference-in-Differences (DID) for both the specialist-ablation (DID_{spec}) and random-ablation (DID_{rand}) scenarios. The final causal estimate is then given by the Triple-Difference: $DDD = \text{DID}_{\text{spec}} - \text{DID}_{\text{rand}}$. Our experiment yielded a statistically significant DDD estimate, confirmed via bootstrap confidence intervals and permutation tests. This provides functional evidence consistent with the view that specialist heads are non-substitutable and disproportionately important for complex, multi-hop reasoning.

C.4.3 Head Disruption Experiments

Full disruption results and analysis are presented in Section 4.3 (Table 3).

D Additional Experiment Results

D.1 Statistical Stability Analysis

To verify that the reported improvements are not due to random variation, we conducted five independent training runs and evaluated on the CWQ test set using Qwen3-30B as the generator. Table 7 reports the mean and standard deviation across runs. Performance is stable across all metrics, with standard deviations below 2 points, confirming that the gains are consistent and reproducible.

Table 7: Performance stability across 5 independent runs on WebQSP and CWQ (Qwen3-30B generator). Standard deviations are all below 2.0 points.

	WebQSP			CWQ		
Model	**Macro-F1**	**Hit**	**Hallu.**	**Macro-F1**	**Hit**	**Hallu.**
ParallaxRAG	75.97 ± 0.88	92.36 ± 1.60	75.85 ± 0.63	59.15 ± 1.39	66.08 ± 1.92	57.72 ± 1.01

Additionally, we performed paired bootstrap resampling (1k samples) on Hit@1 over CWQ using Llama3.1-8B as generator. Improvements over SubgraphRAG ($p = 0.022$) and GNN-RAG

($p = 0.010$) are statistically significant, substantially reducing the likelihood that the observed gains arise from random fluctuations.

D.2 End-to-End Latency Breakdown

Table 8 and Table 9 report a full per-stage latency breakdown (mean $\pm$ std, in seconds per query) on WebQSP and CWQ respectively, measured on a single NVIDIA RTX 6000 Ada GPU. Retrieval latency increases with hop count due to larger subgraph sizes, but is largely invariant to the retrieval budget k because the multi-view scoring and gating are implemented as fully vectorized matrix operations over the entire candidate triple set. Consequently, increasing k enriches the LLM context without incurring additional retrieval overhead. The dominant cost in the pipeline is the LLM generation stage; the multi-view retrieval mechanism itself adds only modest overhead.

Table 8: Per-stage latency (seconds/query, mean $\pm$ std) on WebQSP. "Retrieval (mean)" is averaged over all queries; hop-specific values reflect subgraph complexity.

Method	1-hop	2-hop	≥3-hop	Retrieval	KG Access	Generation
ParallaxRAG	22.10 ± 7.80	70.47 ± 14.28	–	38.78	0.023	3.124 ± 0.232

Table 9: Per-stage latency (seconds/query, mean $\pm$ std) on CWQ.

Method	1-hop	2-hop	≥3-hop	Retrieval	KG Access	Generation
ParallaxRAG	25.58 ± 8.20	87.95 ± 17.17	183.63 ± 63.01	82.35	0.026	3.784 ± 0.227

After extraction, triple recall is computed deterministically via scripted string matching against annotated ground truth, so the final metric does not depend on GPT-4o's generative tendencies. To assess extraction stability, we repeated the extraction with three different models (GPT-4o, Qwen3-30B, GLM-4.7) and observed negligible variance (std $\leq 1.5\%$). We also manually inspected a random 10% subset and found high consistency between automated extraction and human judgment.

E Case Study

We present several representative cases from the CWQ dataset to demonstrate ParallaxRAG's efficacy in multi-hop question answering, emphasizing its refined subgraph construction, reduced hallucinations, and robust handling of constraints through multi-view decoupling and head specialization, while also revealing scenarios where increased retrieval breadth may be unnecessary for simpler queries.

Case 1 (Figure 6): Single-Vector RAG retrieves a noisy subgraph, incorrectly following the containment path *(Nijmegen, ..., Netherlands)* while ignoring the adjacency constraint to France, thus outputting `Netherlands`. This failure highlights how its undifferentiated embeddings struggle to enforce multiple constraints. In contrast, ParallaxRAG's multi-view retrieval isolates the correct evidence path: *(Nijmegen, nearby_airports, Weeze Airport)*, *(Weeze Airport, containedby, Germany)*, and *(Germany, adjoins, France)*. By decoupling and prioritizing geographic constraints, it correctly answers `Germany`.

Case 2 (Figure 7): Single-Vector RAG's flat representation fails to manage two distinct constraints (symbol and bisection). It retrieves a diffuse subgraph, conflates relations, and incorrectly links the Ring-necked Pheasant to Missouri, thus outputting `Missouri`. ParallaxRAG, however, uses constraint-specific heads to retrieve precise and separate facts: *(Ring-necked Pheasant, official_symbol_of, South Dakota)* and *(South Dakota, partially_contains, Missouri River)*. This decomposition allows it to satisfy both constraints and correctly identify `South Dakota`.

Case 3 (Figure 8): This query requires only simple transitive reasoning, where a compact set of facts is sufficient to infer the correct answers. As shown in Figure 8, Single-Vector RAG retrieves a concise collection of relevant triples, enabling the downstream LLM to directly enumerate all championship years. In contrast, ParallaxRAG retrieves a richer set of team- and season-related triples from multiple views. While all retrieved evidence is factually correct, the additional contextual information is not strictly necessary for this shallow inference. Consequently, the downstream LLM adopts a conservative interpretation strategy and outputs an inconclusive answer. This case highlights the scope of multi-view retrieval. While richer contextual coverage is crucial for robustness in complex multi-hop reasoning, ParallaxRAG is not explicitly optimized for minimal-context inference in shallow queries, where simpler retrieval strategies may already suffice.

WebQTrn-241_dfb6c97ac9bf2f0ac07f27dd80f9edc2

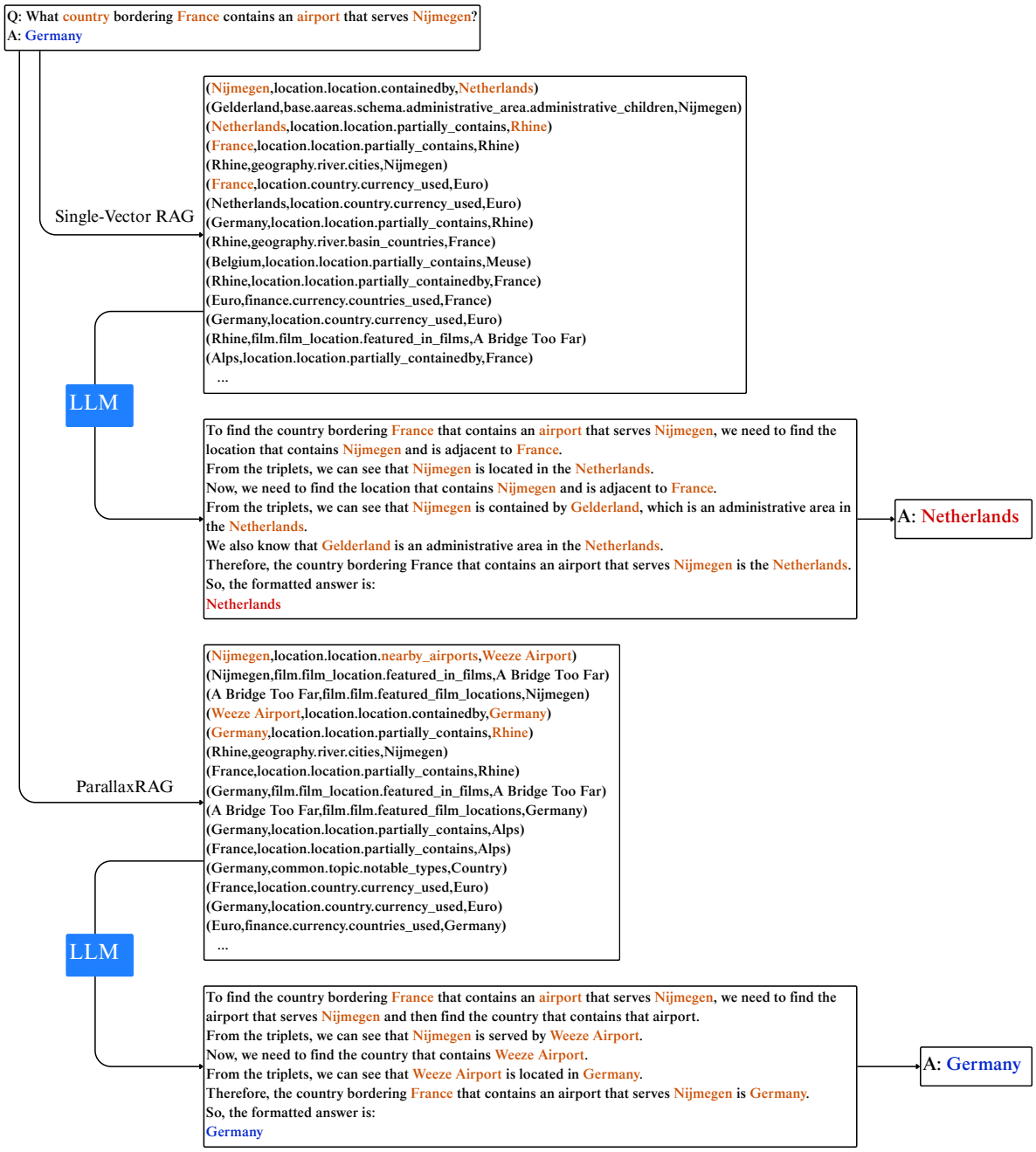


Figure 6: Comparison of retrieved subgraphs and reasoning chains for case WebQTrn-241

WebQTest-626_40c4288896adffe97901ff78149912e5

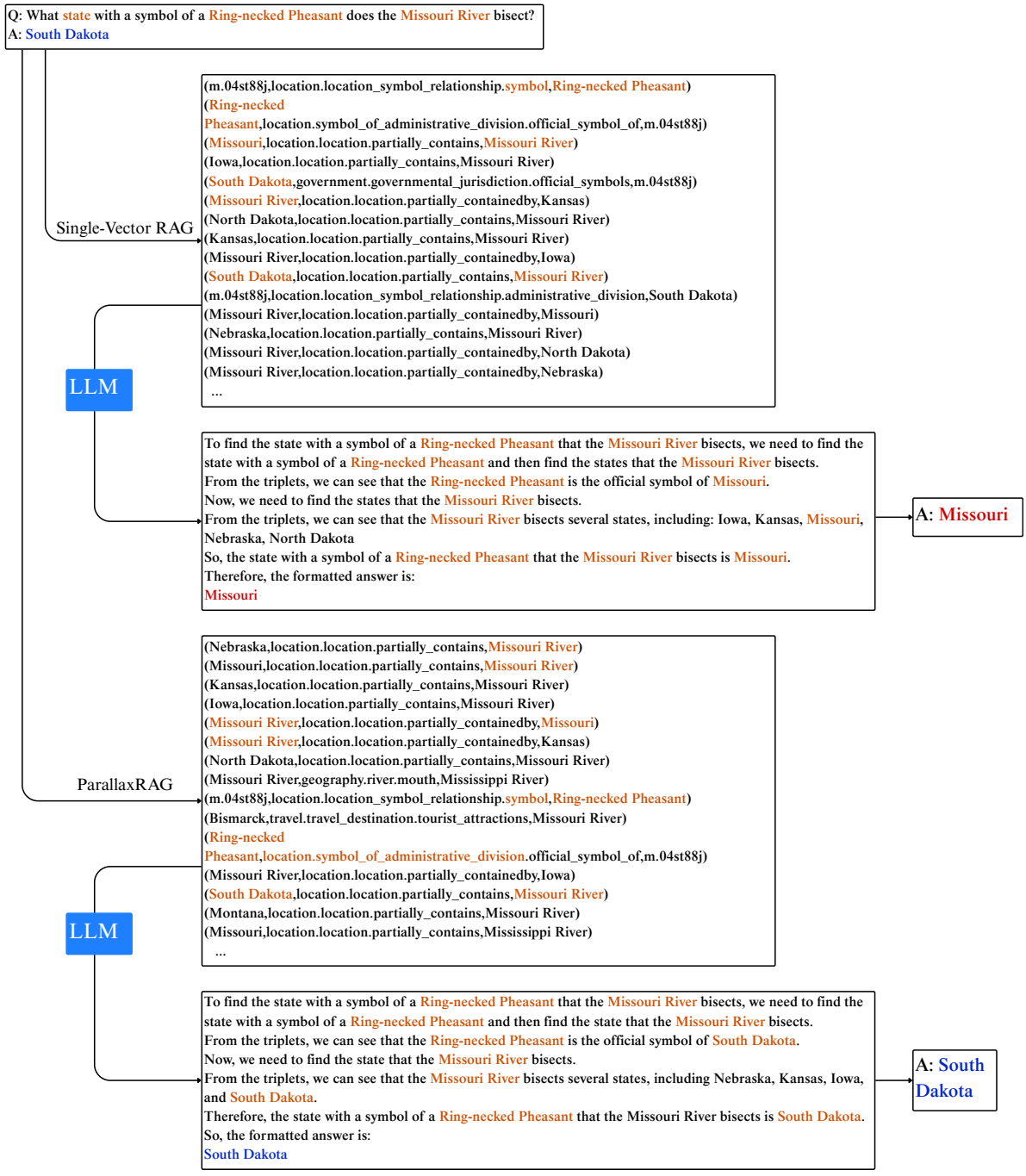


Figure 7: Comparison of retrieved subgraphs and reasoning chains for case WebQTest-626

WebQTrn-810_c334509bb5e02cacae1ba2e80c176499

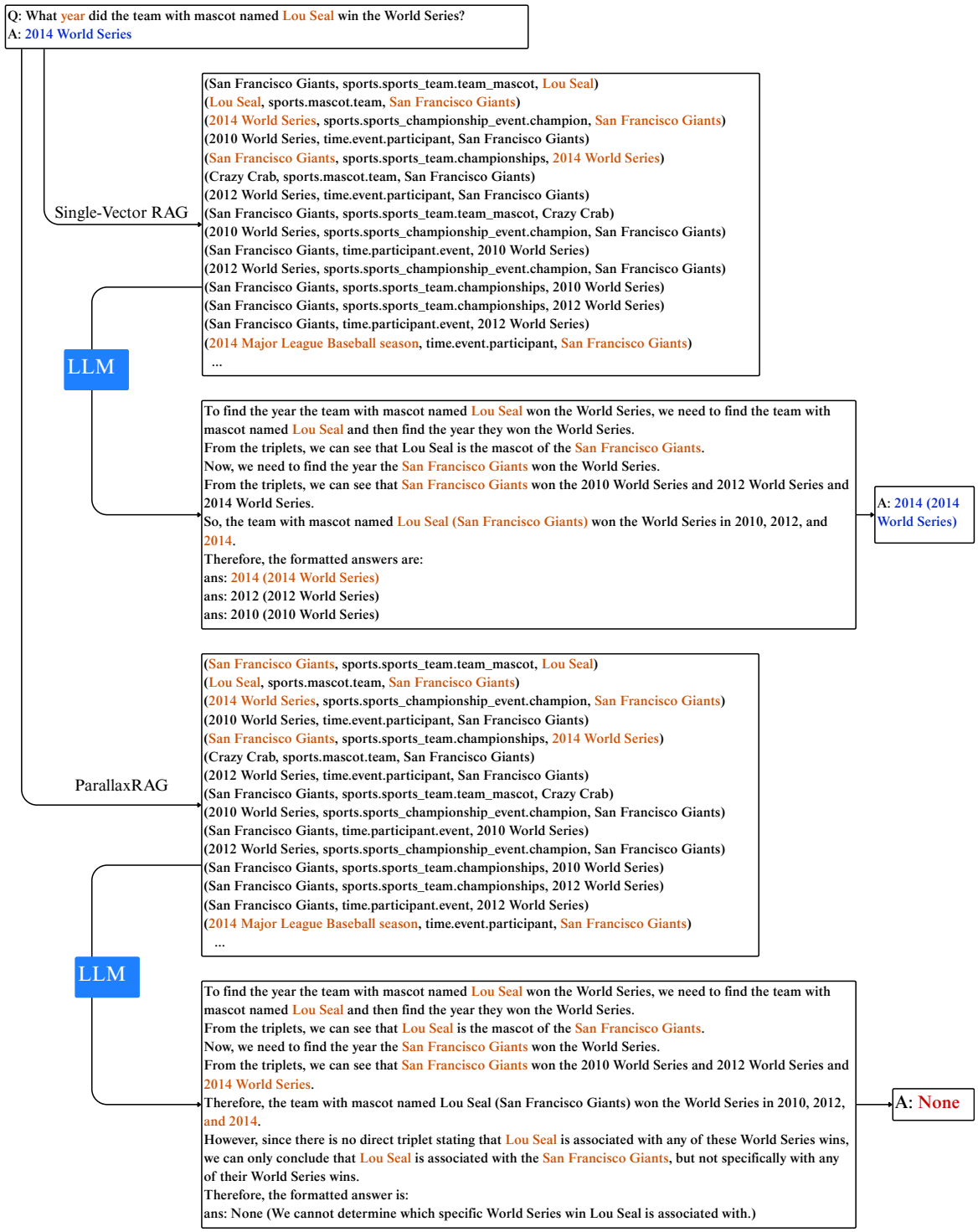


Figure 8: Comparison of retrieved subgraphs and reasoning chains for case WebQTest-810