

Intrinsic Attention as Navigator: A Non-Generative Retrieval Method for Multi-Hop Queries

Linwei Song^a, Weijie Zhang^a, WeiDi Chen^a, Huixia Lai^{*,a}, Shi Zhang^{*,a,b}, Ruliang Xiao^{a,b}

[a] College of Computer and Cyber Security, Fujian Normal University, Fuzhou 350117, China

[b] Digit Fujian Internet-of-Things Laboratory of Environmental Monitoring, Fuzhou 350117, China

*Corresponding authors: lhxfjnu.edu.cn, shifjnu.edu.cn

Abstract

For multi-hop queries in retrieval-augmented generation (RAG), it is crucial to accurately obtain evidence pieces. In this study, we propose a training-free method for multi-hop query retrieval, named Intrinsic Navigating-based Non-Generative Retrieval (IN-Retriever). Instead of generating intermediate reasoning text, the method extracts the bridging clue from the internal attention of the embedding model based on the current context and retrieved candidate documents. During iterative clue extraction and document retrieval, IN-Retriever strips away revealed clues from the embedded query vector and then reranks candidate documents in a targeted manner, thereby steering retrieval toward hidden next-hop evidence. In the experiments, we confirm the feasibility and effectiveness of using intrinsic attention signals to guide external multi-hop retrieval. Experimental results show that our method precisely collects sufficient documents to support multi-hop queries and significantly outperforms sparse and dense retrieval methods.

1 Introduction

Retrieval-augmented generation (RAG) mitigates the limitations of large language models (LLMs) in knowledge timeliness and domain-specific expertise by retrieving information from external data sources (Lewis et al., 2020; Gao et al., 2024). In RAG, retrieving sufficient evidence is crucial for LLMs to generate the final answer. As to multi-hop question answering (QA), RAG systems need to progressively gather evidence across multiple documents. Existing single-pass retrieval methods rely directly on the user query, which can not guarantee the retrieved documents jointly forming a complete evidence chain (Xiong et al., 2021b).

To address the above issue, some studies explicitly generate intermediate reasoning steps and then design retrieval paths (Trivedi et al., 2023; Yao

et al., 2023; Asai et al., 2024). In sequential generation, LLMs may over-commit to early mistakes, a phenomenon known as hallucination snowballing (Zhang et al., 2023). Other studies improve dense dual-encoder retrieval through supervised training and hard-negative selection (Karpukhin et al., 2020; Xiong et al., 2021a). Single-vector dual encoders represent each query and document with one fixed-dimensional vector and score relevance with a single dot product (Ni et al., 2022). ColBERTv2 retains token-level representations and applies late interaction for finer-grained query-document matching (Santhanam et al., 2022).

Recently, Yu et al. (2023) identified individual attention heads in Pythia and GPT-2 that promote either memorized or in-context answers in a counterfactual world-capitals task. By rescaling the heads' value vectors, they changed the relative likelihood of the two answers at runtime. Gould et al. (2024) identified successor heads that increment tokens in natural ordinal sequences and found a weak form of universality across model architectures and scales. These results indicate that individual attention heads can play predictable roles in model behavior. We therefore examine whether attention signals in an embedding model can identify bridging clues and develop Intrinsic Navigating-based Non-Generative Retrieval (IN-Retriever), a training-free method for multi-hop queries. This method guides document retrieval by leveraging the internal attention signals of embedding models, without explicitly generating reasoning text. Specifically, it first activates specific attention heads in the embedding model through instruction-guided prompting, and then extracts bridging clues from the current context based on the model's internal attention to retrieve candidate documents for the next hop. It subsequently strips away revealed clues from the embedded query vector and reranks the candidate documents, thereby steering retrieval toward hidden next-hop evidence.

Overall, the main contributions of this paper are summarized as follows.

(1) We propose a non-generative retrieval method for multi-hop queries, named the Intrinsic Navigating-based Non-Generative Retrieval method (IN-Retriever). The method exploits the internal attention heads of embedding models to discover bridging clues for multi-hop queries.

(2) We verify the implicit relationship between the internal attention of embedding models and bridging clues, and propose an instruction-driven method for locating elite attention heads for obtaining bridging clues with no explicit reasoning text.

(3) We design a residual-guided reranking module, which gradually removes the semantic component in the query vector that has already been expressed by the current clue, and reranks candidate documents using the remaining information, thereby steering retrieval toward next-hop evidence.

(4) Experiments show that IN-Retriever outperforms the feasibility and effectiveness in multi-hop queries, demonstrating a feasible way to leverage the internal attention signals of large models to locate bridging clues.

2 Methodology

Problem Definition: Given a query Q and a large-scale unstructured corpus C , the aim of retrieval is to extract and construct an evidence document, such that these documents can jointly support the reasoning toward the answer A for query Q .

2.1 Framework overview

The structure of IN-Retriever is illustrated in Figure 1. The retrieval process can be divided into three stages: initialization, stepwise reasoning, and final evidence set construction.

Initialization stage. First, the potential named entities and noun phrases contained in the query Q are extracted as an initial entity set E_{init} . Then, the most similar J documents are retrieved from the Vector Datastore (VD) based on semantic retrieval, forming D_{base} . Subsequently, the retrieval-intent residual RQ^{embed} is computed based on the user query embedding Q^{embed} and E_{init} , where the residual computation is defined in Section 2.3. Finally, RQ^{embed} , the document $q_i \in D_{base}$, and score s are combined into J triples $([q_i], RQ^{\text{embed}}, s)$ and inserted into the pri-

ority queue PQ , where s denotes the cumulative score of the evidence chain $[q_i]$ and serves as the priority metric in the priority queue, and the scoring mechanism is defined in Section 2.5.

Stepwise reasoning stage. Elements are repeatedly popped from PQ and processed. For each element $(D', RQ^{\text{embed}}, s')$, the process consists of three steps as follows.

(1) The instruction I , the last document q' in the document list D' , and the original query Q are fed into the embedding model. Next, a bridging clue e' is extracted from q' using the instruction-driven bridging clue probing method (Section 2.2). Subsequently, e' is used to retrieve a candidate document set D_{cand} from the VD.

(2) Based on RQ^{embed} , the semantic residual-based reranking method (Section 2.4) is applied to D_{cand} to obtain the top- D documents $\{q_1, \dots, q_D\}$.

(3) The D documents are individually appended to D' , forming D document lists $\{D'_1, \dots, D'_D\}$. Next, the retrieval-intent residual IRQ^{embed} is calculated based on RQ^{embed} and e' . Based on s' , the cumulative score s''_i of each new evidence chain D'_i is calculated. If s''_i is greater than the threshold θ , the evidence chain D'_i is added to D_{final} as a candidate evidence path; otherwise, a new triple $(D'_i, IRQ^{\text{embed}}, s''_i)$ is pushed into PQ .

The stepwise reasoning expands the reasoning evidence chain. To avoid unlimited expansion, IN-Retriever also limits the expansion path length.

Final evidence set construction stage. Evidence paths are selected from D_{final} in descending order of score. The documents are merged as final retrieval results.

IN-Retriever adopts the beam search mechanism (Zhao et al., 2021), which allows it to maintain multiple potential reasoning trajectories (trajectory hypotheses) simultaneously, thereby enabling the model to tolerate local uncertainty.

In the remainder of this section, we will introduce the instruction-driven bridging clue probing method (Section 2.2), retrieval-intent residual computation (Section 2.3), semantic residual-based reranking (Section 2.4), and the cumulative scoring mechanism for evidence chains (Section 2.5) in detail.

2.2 Instruction-Driven Bridging Clue Probing

The instruction-driven bridging clue probing method aims to identify bridging clues supporting next-hop expansion. It consists of two steps:

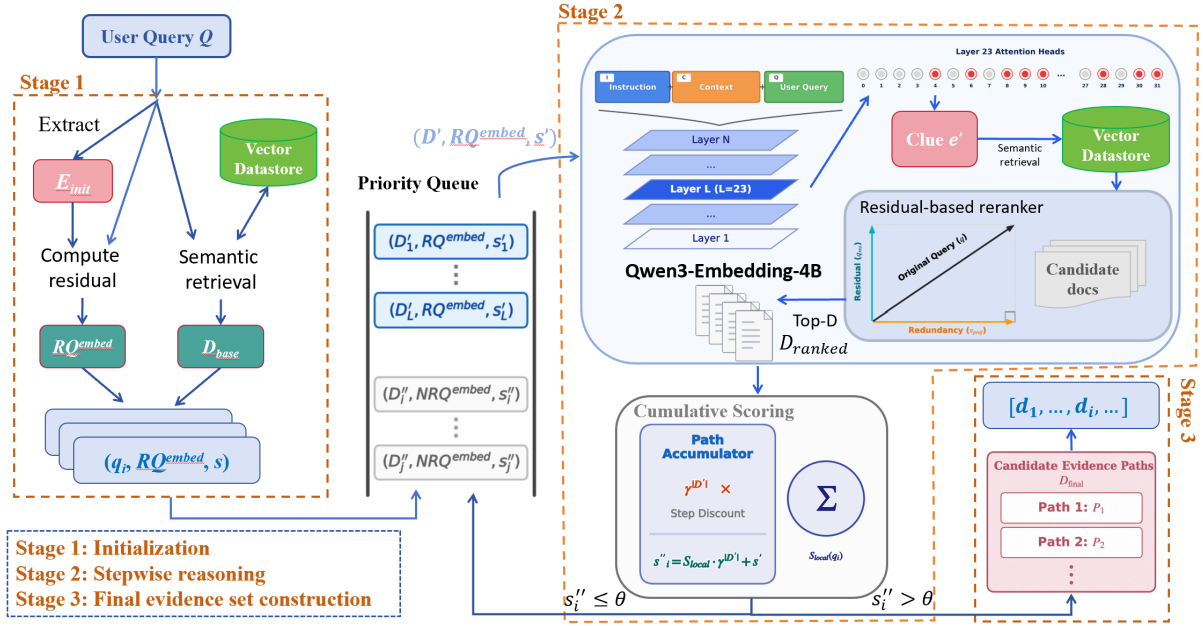


Figure 1: Framework of the Intrinsic Navigating-based Retrieval Method (IN-Retriever). The overall process is delineated into three stages: initialization, stepwise reasoning, and final evidence set construction. The initialization stage retrieves documents grounded on semantic similarity. The stepwise reasoning stage employs a beam search mechanism to progressively expand the document evidence chains. Ultimately, the trajectory attaining the highest cumulative score is selected to yield the final evidence set.

task-oriented attention head selection and bridging clue extraction. Among them, the first step requires only a one-time calibration for a specific embedding model to identify the attention heads. Once the calibration and selection are completed, no further adjustment is required during the subsequent inference process.

2.2.1 Task-Oriented Attention Head Selection

To steer the model’s internal attention toward clues relevant to the next hop, we propose a task-oriented method for selecting key attention heads, termed *elite attention heads*. Specifically, task orientation is implemented through instruction guidance by prepending a task-related instruction I to the input, together with the context C and the user query Q . The instruction is defined as:

Identify the key evidence in the context that helps find the next context to finally answer the question. Focus specifically on the entity type requested in the query.

The probing input is organized as a single sequence $[I; C; Q]$, corresponding to $[\text{Instruct}; \text{Context}; \text{Query}]$, where the instruction appears once at the beginning. Clue extraction is restricted to the token span of C , while the instruction and query tokens only guide bridge clue probing and are never decoded.

This instruction encourages the embedding model to focus on local segments in the context that are relevant to the query, thereby activating specific attention heads and providing more stable clue signals for key attention head selection and subsequent bridging clue probing.

For elite attention head selection, we first construct a calibration dataset $\mathcal{D}_{\text{calib}}$ based on the 2WikiMultihopQA (2Wiki) (Ho et al., 2020), where each sample contains a context C , a query Q , and a bridging clue E extracted via a lexical matching strategy. Then, for each attention head, we compute the Bridge Focus Score (BFS), as defined in Formula 1, to measure the degree of attention allocated to the bridging clue.

$$\text{BFS}(h, l) = \frac{1}{|\mathcal{D}_{\text{calib}}|} \sum_{(C, Q, E) \in \mathcal{D}_{\text{calib}}} s_{Q, E}^{(h, l)} \quad (1)$$

$$s_{Q, E}^{(h, l)} = \frac{1}{|T_Q|} \sum_{i \in T_Q} \sum_{j \in T_E} A_{i, j}^{(h, l)}$$

where T_Q and T_E denote the token-index sets of the query and the annotated bridging clue. $A_{i, j}^{(h, l)}$ is the attention weight from position i to position j by the h -th head in the l -th layer.

Finally, based on the layer-wise bridge focus score (LBFS) as defined in Formula 2, we select the L -th layer as the key attention layer, and then

choose the top- H attention heads with the highest BFS values in that layer as the elite head set $\mathcal{H}_{\text{elite}}$.

$$\text{LBFS}(l) = \sum_{h \in \arg\text{Top}_H(\text{BFS}(h,l))} \text{BFS}(h,l) \quad (2)$$

where H denotes the number of selected attention heads. We set $H = 8$.

According to our empirical analysis, **for the Qwen embedding model, attention heads in $\mathcal{H}_{\text{elite}} = \{10, 30, 8, 28, 31, 6, 4, 9\}$ at layer $L = 23$ are set as elite attention heads**. They are the basis for bridging clue extraction. All layer and head indices in this paper follow a zero-indexed convention; therefore, the 32 heads of Qwen3-Embedding-4B are indexed from 0 to 31.

2.2.2 Bridging Clue Extraction

Bridging clue extraction aims to extract text fragments that bridge hops. After feeding $[I; C; Q]$ into the embedding model, we first pinpoint local peaks in $\mathcal{H}_{\text{elite}}$, and then extend them into complete semantic units through locally aware boundary expansion. The extraction method seeks to isolate clues with reasoning value from long documents. It consists of the following steps.

(1) Given the query Q and the document $C = \{s_1, \dots, s_i, \dots\}$, where s_i is a sentence in C , we define the average attention weight of each sentence s_i on the elite attention heads according Formula 3, and then retain the top- S sentences with the highest weights for the subsequent process. In our experiments, we set $S = 3$.

$$W(s_i) = \frac{1}{|s_i|} \sum_{t \in s_i} \sum_{h \in \mathcal{H}_{\text{elite}}} \text{Att}(t, h) \quad (3)$$

where $|s_i|$ is the size of tokens in s_i , and $\text{Att}(t, h)$ denotes the attention weight assigned to token t by attention head h .

(2) For the selected S sentences, we accumulate attention weights of each token over the elite attention heads. Let $t_{j,i}$ denote the i -th token in sentence s_j . Its cumulative attention weight on $\mathcal{H}_{\text{elite}}$ is defined as Formula 4.

$$\text{CW}(t_{j,i}) = \sum_{h \in \mathcal{H}_{\text{elite}}} \alpha_{h,i} \quad (4)$$

where $\alpha_{h,i}$ denotes the attention weight assigned to $t_{n,i}$ by the h -th elite head.

(3) For each selected sentence, we select the top- A token anchors with the highest cumulative attention scores on $\mathcal{H}_{\text{elite}}$ as peak anchors. In our experiments, $A = 20$.

(4) For each selected sentence, we take the peak anchors as centers, and extend them bidirectionally to absorb adjacent words until a predefined syntactic-semantic blocking token is reached. The clue is then obtained by decoding the extracted tokens. By extracting one clue from each sentence, we finally acquire a total of S clues.

Overall, based on the internal attention weights of the embedding model and the syntactic structure of the text, the probing method provides a schema for conversing diffuse attention signals to discrete reasoning clues at a low computational cost.

2.3 Retrieval-Intent Residual Computation

In multi-hop retrieval, relying solely on the original query Q often traps the retriever within a semantic neighborhood highly similar to Q . Consequently, even after multiple retrieval rounds, the model still repeatedly recalls documents that contribute little new evidence (as shown in Figure 2(a)). To enable the retriever to discover new evidences for next-hop reasoning, we propose a retrieval-intent residual computation method. This method strips away the semantic components of the query embedding that have been referred by the current clue, so that the residual can better guide the retriever to capture the clue required for the next hop (as shown in Figure 2(b)).

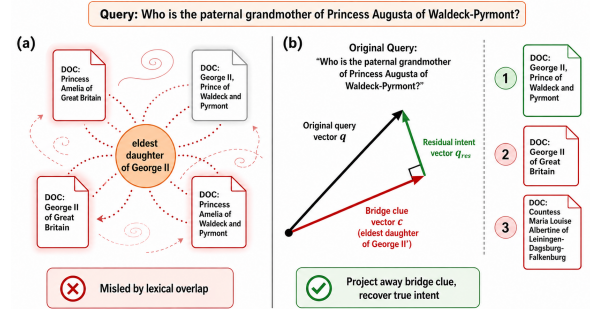


Figure 2: Reasoning for the multi-hop query "Who is the paternal grandmother of Princess Augusta of Waldeck-Pyrmont?". (a) Misled by the high lexical frequency of "George II" and "Princess", the dense retriever always recalls documents concerning "King George II of Great Britain" and "Princess Amelia", failing to pinpoint the gold-standard document (George II, Prince of Waldeck). (b) Upon successfully identifying the bridging clue "eldest daughter of George II", the query is projected onto the orthogonal complement of the clue. By isolating the residual intent vector and maximizing its alignment score, the retriever explicitly upweights the target document (George II, Prince of Waldeck and Pyrmont).

Let Q^{embed} denote the current query embedding, let e denote the textual bridging clue, and let e^{embed}

denote its embedding. We compute the retrieval-intent residual by subtracting from Q^{embed} its component along e^{embed} as:

$$Q_{\text{res}}^{\text{embed}} = Q^{\text{embed}} - \frac{Q^{\text{embed}} \cdot e^{\text{embed}}}{\|e^{\text{embed}}\|^2} e^{\text{embed}} \quad (5)$$

The resulting vector $Q_{\text{res}}^{\text{embed}}$ captures the components of Q^{embed} that are orthogonal to the embedding of bridging-clue. This projection has a clear geometric interpretation and serves as a practical reranking signal rather than a strict discriminator of natural-language semantics.

2.4 Semantic Residual-based Reranking

After retrieving candidate documents based on the bridging clue, the documents that are most relevant to the clue need to be identified, avoiding arbitrary drifting of evidence chains. To this end, a semantic residual-based reranking method is proposed to score documents using Formula 6.

$$S_{\text{rerank}}(c^{\text{embed}}) = \lambda \text{sim}(Q^{\text{embed}}, c^{\text{embed}}) + (1 - \lambda) \text{sim}(Q_{\text{res}}^{\text{embed}}, c^{\text{embed}}) \quad (6)$$

where $\text{sim}(\cdot, \cdot)$ computes similarity, Q^{embed} is the original query embedding, $Q_{\text{res}}^{\text{embed}}$ is the retrieval-intent residual, c^{embed} is the candidate-document embedding, and $\lambda \in [0, 1]$ is the mixing coefficient between the two similarity terms.

This scoring method jointly captures the global semantic relevance to the original query and the targeted relevance to the retrieval-intent residual, thereby upweighting candidate documents that address the unresolved semantic components along the current retrieval path. Finally, the top- D most relevant documents are retained. Each of the retained documents is subsequently appended to the current evidence chain in parallel, yielding D newly expanded reasoning paths.

2.5 Scoring Mechanism for Evidence Chains

In the stepwise reasoning stage, it is essential to estimate the reliability of expanded paths. We define a unified scoring function composed of a local node score (Definition 1) and a cumulative path score (Definition 2). The scoring function jointly considers the global relevance to the original query and the guidance provided by the retrieval residual for the next hop.

Definition 1 (Local Node Score). Given the original query embedding Q^{embed} , the retrieval-intent

residual RQ^{embed} , a bridging clue e , and a candidate document q , the local node score of q is defined as:

$$S_{\text{local}}(q) = \lambda_{\text{vec}} \cos(Q^{\text{embed}}, q^{\text{embed}}) + \lambda_{\text{intent}} \text{ReLU}(RQ^{\text{embed}} \cdot q^{\text{embed}}) + \lambda_{\text{sym}} S_{\text{sym}}(q, e). \quad (7)$$

where $\text{Tok}(\cdot)$ denotes the token set obtained after lowercasing and punctuation stripping, and $S_{\text{sym}}(q, e) = \frac{|\text{Tok}(e) \cap \text{Tok}(q)|}{|\text{Tok}(e)|}$ measures the explicit token coverage of the textual bridging clue e in document q . q^{embed} and e^{embed} are the embeddings of q and e .

The three terms keep the path aligned with the original query, favor documents matching the unresolved retrieval intent, and require explicit support for the current bridging clue.

Definition 2 (Cumulative Score). Let the current evidence chain is D' with cumulative score s' . When a candidate document q is appended to D' to form a new path $D'' = D'. \text{append}(q)$, the cumulative score of D'' is defined as:

$$S_{\text{path}}(D'') = s' + \gamma^{|D'|} \cdot S_{\text{local}}(q) \quad (8)$$

where $\gamma \in (0, 1)$ is the step discount factor, and $|D'|$ denotes the path depth.

The cumulative scoring mechanism enables the search process to preferentially preserve reasoning trajectories that are continuously supported by high-quality local evidence. The discount term penalizes excessively long paths and mitigates the accumulation of uncertainty in deeper expansions. Ultimately, the cumulative scoring mechanism jointly considers the historical score of the path and the importance of the current document.

3 Experiments and Analysis

In this section, we first introduce the experimental setup (Section 3.1), and then analyze the effectiveness of the internal-attention-guided mechanism (Section 3.2). Subsequently, comparisons are conducted to evaluate the overall performance of our method (Section 3.3).

3.1 Experimental Setup

Datasets: Three representative multi-hop question answering datasets, 2Wiki (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and HotpotQA (Yang et al., 2018), are selected as benchmark datasets.

Baseline Methods: Three categories of retrieval method are selected as baselines. They are the sparse retrieval method BM25 (Robertson and Zaragoza, 2009), the standard dense retrieval model GTR (T5-Large) (Ni et al., 2022), and strong large-scale embedding models, including Qwen3-Embedding-8B (Zhang et al., 2025) and SFR-Embedding-Mistral-7B (Meng et al., 2024). Our method is build on Qwen3-Embedding-4B.

Implementation Details: By default, all experiments use Qwen3-Embedding-4B with the same probing configuration. The maximum reasoning depth is fixed to $T = 3$ for all three datasets. The weights in local node score are $\lambda_{\text{vec}} = 8$, $\lambda_{\text{intent}} = 7$, and $\lambda_{\text{sym}} = 20$. The reranking coefficient in Eq. 6 is $\lambda = 8/(8 + 7) = 0.533$, and the step discount is $\gamma = 0.90$. These settings are not retuned for individual datasets. In Appendix B.5, we analyze the hyperparameters in detail.

Evaluation Metric: For each query and each $F \in \{5, 10, 15, 20\}$, IN-Retriever retains the top- F reasoning paths and deduplicates their documents. Let D_F denote the resulting document count. Each baseline then returns its top- D_F documents for the same query. Recall@ F therefore measures supporting-document coverage under F paths. Appendix B.4 reports the complementary document-level F1 analysis under the same alignment protocol.

3.2 Attention Analysis

The effectiveness of IN-Retriever is grounded in a fundamental hypothesis: the elite attention heads of a pre-trained embedding model imply the clues required for reasoning. In this subsection, we analyze the model’s layer-wise attention weights, verify the stability of the elite attention heads that play the dominant role in locating reasoning clues, and visualize two cases for understanding.

3.2.1 Layer-wise Sensitivity

To investigate the layer-wise pattern of the attention mechanism, we randomly sampled 400 query-context pairs from 2Wiki, fed them into the model, and output attention weights. The heat map of bridge focus score (BFS) across attention heads is visualized as Figure 3a. The figure shows that attention heads with high BFS values are mainly concentrated in the middle-to-late layers of the model, revealing a sparse and clearly non-uniform spatial distribution. Similar results (as shown in Figure 3b and Figure 3c) are also supported in the other

datasets.

Following the setting, we extracted the top-8 attention heads with the highest BFS values in each layer and evaluated the ability to retrieve the correct next-hop documents via the bridging clues. The results are shown in Figure 4a.

Figure 4a shows that the recall exhibit a consistent trend across all three datasets. The recall first decreases in layers 0 to 3, then remains relatively low and stable from layers 4 to 12. It subsequently rises gradually from layers 13 to 23 with fluctuations. Finally, the recall reaches its peak at layer 23, followed by fluctuations across layers 24–34. In the last layer (layer 35), the recall exhibits a distinct decline. This layer-wise trend is broadly consistent with earlier BERT probing studies that reported lower-layer surface or lexical information and higher-layer semantic information (Jawahar et al., 2019; Tenney et al., 2019). Niu et al. (2022) later found that a simple layer-wise pipeline lacks conclusive empirical support and that layer depth alone does not fully explain BERT’s internal organization.

3.2.2 Analysis of Elite Attention Heads

Furthermore, taking layer 23 as the basis, we retrieve the next-hop document using the top- H heads with higher BFS scores. Figure 4(b) shows the relationship between H and the recall of next-hop document. Across all three datasets, the figure shows that the head exhibiting the highest BFS encapsulate the vast majority of the model’s reasoning capacity. As H increases, the highest recall is achieved when $H = 8$. On MuSiQue and 2Wiki, recall increases noticeably as the number of attention heads grows when $H < 8$. When $H > 8$, recall on 2Wiki largely plateaus, whereas recall on MuSiQue and HotpotQA shows a certain degree of decline. These results suggest that it is sufficient to focus on attention heads with the higher BFS values, while incorporating too many heads may instead introduce noise and degrade the performance. Taking both performance and stability into account, we finally select the attention heads specified by $\mathcal{H}_{\text{elite}} = \{10, 30, 8, 28, 31, 6, 4, 9\}$ in layer $L = 23$ as the elite attention heads for the Qwen3-Embedding-4B.

3.2.3 Case Analysis

Focusing on head 10 at layer 23, we visualize token-level attention patterns of two representative cases. The weight matrix between the query and the con-

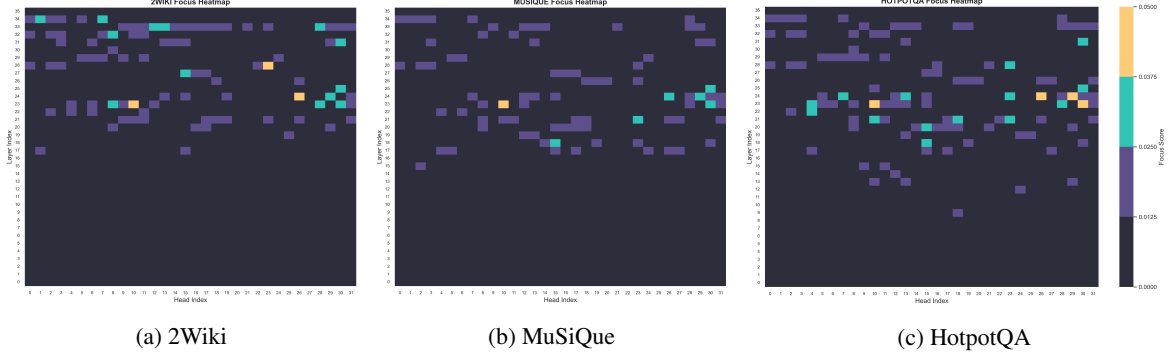


Figure 3: Heatmaps of Bridge Focus Score (BFS) on 2Wiki, MuSiQue, and HotpotQA.

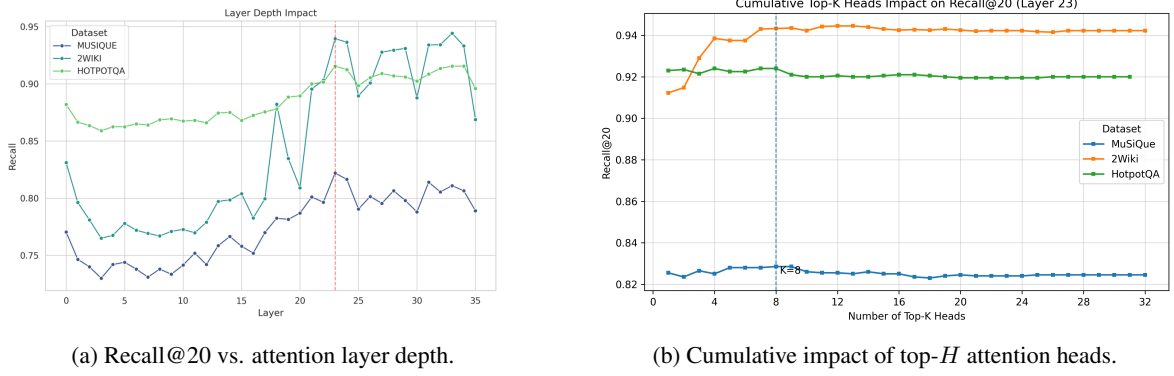


Figure 4: Layer-wise and head-wise analysis for selecting elite attention heads on 2Wiki, MuSiQue, and HotpotQA.

text is shown in Figure 5.

In Figure 5a, it exhibits distinct and persistent vertical bands. This geometric pattern indicates that the attention head consistently focuses on the same bridging clue across multiple query tokens, rather than distributing attentions diffusely over the broader context. In Figure 5b, it show a case with dense semantic interference, where the context contains multiple homogeneous named entities. Even in the presence of several same-type distractors, the attention remains strongly discriminative and aligns more concentratedly with the target clue. The visualization of the two cases further verifies that the selected attention head is not merely performing surface-level lexical matching, but instead provides a stable and interpretable signal for clue localization.

3.3 Comparative Analysis

The comparison results between IN-Retriever and retrieval baselines on the three multi-hop datasets are shown in Table 1. The quantitative evaluation focuses on the retriever itself: whether it recalls supporting evidence under the retrieval defined above.

As shown in Table 1, IN-Retriever with Qwen3-

Embedding-4B as the backbone achieves the best results on R@5, R@10, R@15, and R@20 for the more challenging MuSiQue and 2Wiki datasets. Meanwhile, on HotpotQA, it outperforms direct retrieval baselines of similar scale, including BM25, GTR, and Qwen3-Embedding-4B, while remaining competitive with larger embedding models such as SFR-Embedding-Mistral-7B and Qwen3-Embedding-8B.

Taking R@20 as an example, comparing with simple baselines, BM25 and GTR-Large, our method improves R@20 by +27.80 and +20.05 on MuSiQue, by +24.60 and +23.82 on 2Wiki, and by +4.10 and +8.60 on HotpotQA. These results reveal the following two key observations: (1) explicitly leveraging internal attention signals to guide retrieval is effective for multi-hop evidence search; and (2) for multi-hop queries, relying solely on embedding-based document retrieval is insufficient.

We further examine the recall of two larger embedding models, SFR-Embedding-Mistral-7B and Qwen3-Embedding-8B. First, despite their larger parameter scales, their performances do not apparently exceed that of Qwen3-Embedding-4B (Direct). This suggests that although larger embed-

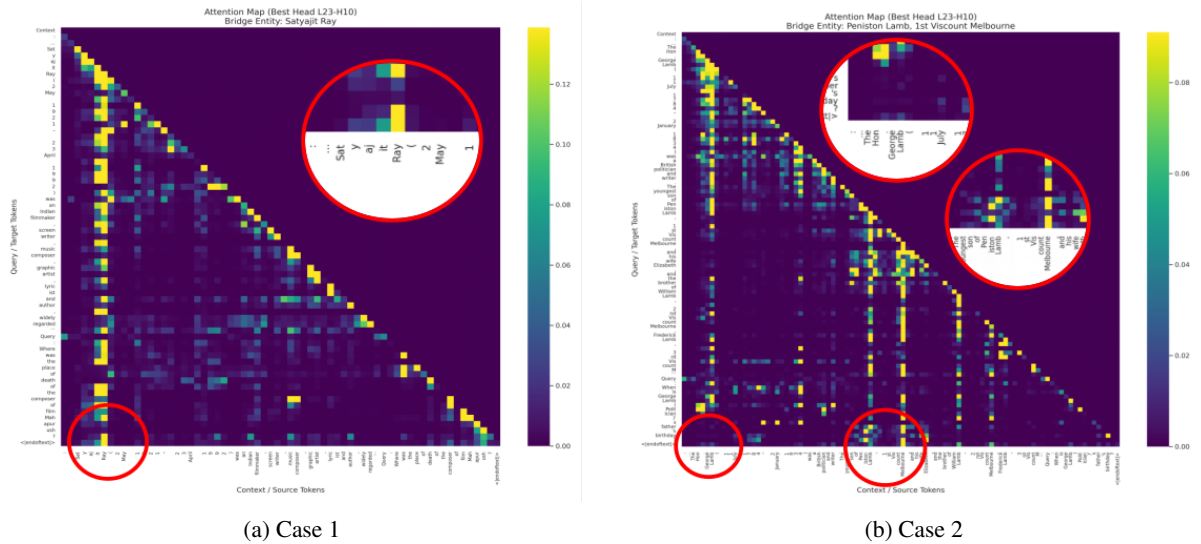


Figure 5: Case analysis of token-to-token in head 10, layer 23. (a) Case 1 shows persistent vertical attention bands. (b) Case 2 shows a dense semantic interference from homogeneous entities

Table 1: Comparison of retrieval methods on MuSiQue, 2Wiki, and HotpotQA. F denotes the number of retained IN-Retriever paths. Baseline methods select the corresponding D_F deduplicated documents for each query comparisons. The best performances are **bolded**, and the second-best performances are underlined.

Retrieval Methods	MuSiQue				2Wiki				HotpotQA			
	R@5	R@10	R@15	R@20	R@5	R@10	R@15	R@20	R@5	R@10	R@15	R@20
<i>Simple Baselines</i>												
BM25	48.80	53.30	57.40	59.30	64.50	68.43	70.17	70.97	76.35	87.70	91.15	92.60
GTR (T5-Large)	53.50	59.65	64.15	67.05	65.87	69.05	70.63	71.75	78.50	84.15	86.70	88.10
<i>Large Embedding Models</i>												
SFR-Embedding-Mistral-7B	<u>65.65</u>	<u>72.55</u>	<u>77.90</u>	<u>79.75</u>	73.95	<u>78.17</u>	<u>79.73</u>	<u>80.77</u>	90.60	95.10	96.30	96.90
Qwen3-Embedding-8B	65.40	72.45	77.85	79.30	<u>74.02</u>	77.32	79.43	80.55	<u>88.75</u>	<u>93.30</u>	<u>95.10</u>	95.85
<i>Baseline RAG</i>												
Qwen3-Embedding-4B (Direct)	63.80	70.20	75.35	77.95	72.82	76.28	77.85	78.90	86.10	92.15	93.90	94.80
IN-Retriever	76.45	82.80	85.40	87.10	89.10	94.33	95.27	95.57	86.35	92.10	94.75	<u>96.70</u>

ding models are generally more favorable for document retrieval and can improve recall, such gains remain limited in multi-hop settings. Moreover, as shown in Table 1, except on HotpotQA, our method achieves the best recall on the other two datasets, which demonstrates that even when using Qwen3-Embedding-4B as the backbone, IN-Retriever can in most cases match or even surpass larger embedding model baselines. This further confirms the necessity of targeted multi-hop clue discovery and stepwise retrieval guided by such clues.

Finally, a comparison across datasets shows that the recall improvements of our method are more pronounced on MuSiQue and 2Wiki, while the gains on HotpotQA are relatively limited. This is consistent with the analysis in Min et al. (2019), which suggests that a substantial fraction of HotpotQA queries can be resolved through reasoning shortcuts, with answers obtainable from a single

document. In contrast, 2Wiki and MuSiQue rely more on bridging evidence across documents. In such tasks, the system must not only retrieve documents that are superficially relevant to the original question, but also progressively identify the bridging clues that connect different pieces of evidence. Compared with methods that rely on a single query vector alone, IN-Retriever demonstrates clearer advantages in multi-hop retrieval and is better suited to retrieval tasks that require progressively locating multi-hop evidence.

4 Conclusion

Establishing a coherent and complete evidence chain remains a core challenge for multi-hop QA tasks. In this study, we propose a training-free Intrinsic Navigating-based non-generative Retrieval method (IN-Retriever). The method directly extracts bridging clues from the internal attention of

the embedding model and uses them to retrieve candidate documents.

Based on Qwen3-Embedding-4B, we first analyze the BFSs across attention heads, revealing clear layer-wise sensitivity and stable distribution patterns among high-BFS heads. Then, the experiments verify the consistency of the elite attention heads across datasets. Comparative experiments indicate that IN-Retriever is effective at locating bridging clues and retrieving supporting evidence, and that attention weights can be directly leveraged to guide multi-hop retrieval.

Overall, our findings suggest that pre-trained embedding models contain internal attention signals that support multi-hop retrieval in a non-generative manner. This study offers a practical idea for bridging internal model signals with external retrieval in complex reasoning tasks.

Limitations

Although IN-Retriever shows strong gains on MuSiQue and 2Wiki and remains competitive on HotpotQA, the conclusions of this work should be interpreted within a limited scope. Our results show that intrinsic attention in embedding models can provide a useful signal for bridge-clue discovery in multi-hop retrieval, but they do not provide a causal explanation of how embedding models perform multi-hop reasoning. Attention is used here as a retrieval signal rather than as an internal reasoning process.

The orthogonal projection used in retrieval-intent residual computation has a clear geometric interpretation and is empirically supported by ablation studies, yet noise in bridging-clue representations and the diversity of entity expressions may reduce the reliability of residual guidance.

IN-Retriever is training-free. However, it requires white-box access to internal attention weights and a one-time calibration step to identify elite attention heads for a given backbone. Although we observe similar sparse BFS patterns in other architectures, comprehensive retrieval evaluations across diverse embedding backbones, closed-source retrieval APIs, multilingual scenarios and domain-specific models have not yet been conducted.

The current clue extraction procedure also relies on several lightweight hand-designed steps, including sentence filtering, peak-token selection, bound-

ary expansion, lexical support scoring, and fixed search budgets. These choices make the method simple and interpretable, but they may be less robust when bridging clues are weakly lexicalized, highly paraphrastic, distributed across multiple sentences, or expressed through temporal, numerical, comparative, or implicit relational constraints. The method is therefore most naturally suited to entity- or noun-phrase-centric multi-hop questions, and its behavior on longer or more highly branching evidence chains remains to be studied.

Finally, our evaluation focuses on retrieval quality rather than end-to-end generation. This matches the main contribution of the paper, but higher Recall@F does not necessarily imply proportional improvements for every downstream generator. In addition, IN-Retriever returns a dynamic number of deduplicated documents and requires iterative attention probing, multi-step retrieval, and reranking, which introduce more inference overhead than single-shot dense retrieval. Future work should study fixed-budget evaluation, end-to-end RAG performance, and latency-quality trade-offs in realistic deployments.

References

- Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. [Self-RAG: Learning to retrieve, generate, and critique through self-reflection](#). In *International Conference on Learning Representations*.
- Shiwei Chen, Bin Liang, Yue Yu, Kam-Fai Wong, Hui Wang, and Ruifeng Xu. 2026a. [Retrieving on a topic graph for long document question answering](#). *Neuro-computing*, 669:132447.
- Xiaojin Chen, Sai Zhang, Xinxing Li, Ruiqin Ma, and Qinan Zhao. 2026b. [DairyGoatQA: A knowledge graph enhanced large language models approach for question answering in the dairy goat domain](#). *Expert Systems with Applications*, 299:130146.
- Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. [From local to global: A graph RAG approach to query-focused summarization](#). *arXiv preprint arXiv:2404.16130*.
- Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. [Retrieval-augmented generation for large language models: A survey](#). *arXiv preprint arXiv:2312.10997*. Submitted in 2023; revised in 2024.

- Rhys Gould, Euan Ong, George Ogden, and Arthur Conmy. 2024. [Successor heads: Recurring, interpretable attention heads in the wild](#). In *International Conference on Learning Representations*.
- Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. [Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. [What does BERT learn about the structure of language?](#) In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.
- Vladimir Karpukhin, Barlas Öguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. [Dense passage retrieval for open-domain question answering](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. [Retrieval-augmented generation for knowledge-intensive NLP tasks](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.
- Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. [Locating and editing factual associations in GPT](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372. Curran Associates, Inc.
- Rui Meng, Ye Liu, Shafiq Joty, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. 2024. [SFR-Embedding-Mistral: Enhance text retrieval with transfer learning](#). Salesforce Blog. Published on October 28, 2024.
- Sewon Min, Eric Wallace, Sameer Singh, Matt Gardner, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. [Compositional questions do not necessitate multi-hop reasoning](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4249–4257, Florence, Italy. Association for Computational Linguistics.
- Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. 2022. [Large dual encoders are generalizable retrievers](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Jingcheng Niu, Wenjie Lu, and Gerald Penn. 2022. [Does BERT rediscover a classical NLP pipeline?](#) In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3143–3153, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, and 7 others. 2022. [In-context learning and induction heads](#). *arXiv preprint arXiv:2209.11895*. Transformer Circuits Thread.
- Stephen E. Robertson and Hugo Zaragoza. 2009. [The probabilistic relevance framework: BM25 and beyond](#). *Foundations and Trends in Information Retrieval*, 3(4):333–389.
- Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. [ColBERTv2: Effective and efficient retrieval via lightweight late interaction](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734, Seattle, United States. Association for Computational Linguistics.
- Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. [BERT rediscovers the classical NLP pipeline](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. [MuSiQue: Multi-hop questions via single-hop question composition](#). *Transactions of the Association for Computational Linguistics*, 10:539–554.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. [Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037, Toronto, Canada. Association for Computational Linguistics.
- Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021a. [Approximate nearest neighbor negative contrastive learning for dense text retrieval](#). In *International Conference on Learning Representations*.
- Wenhan Xiong, Xiang Lorraine Li, Srinivasan Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Wen tau Yih, Sebastian Riedel, Douwe Kiela, and Barlas Öguz. 2021b. [Answering complex open-domain questions with multi-hop](#)

dense retrieval. In *International Conference on Learning Representations*.

Wan-Chi Yang, Xuan Li, Chih-Yung Chang, and Diptendu Sinha Roy. 2026. [Hierarchical knowledge graph-based QA systems with retrieval-augmented generation](#). *Enterprise Information Systems*, 20(1):2580477. Published online: 19 November 2025.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. [HotpotQA: A dataset for diverse, explainable multi-hop question answering](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. [ReAct: Synergizing reasoning and acting in language models](#). In *International Conference on Learning Representations*.

Qinan Yu, Jack Merullo, and Ellie Pavlick. 2023. [Characterizing mechanisms for factual recall in language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9924–9959, Singapore. Association for Computational Linguistics.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. [Qwen3 embedding: Advancing text embedding and reranking through foundation models](#). *arXiv preprint arXiv:2506.05176*.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Chen Xu, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. [Siren’s song in the AI ocean: A survey on hallucination in large language models](#). *arXiv preprint arXiv:2309.01219*.

Chen Zhao, Chenyan Xiong, Jordan Boyd-Graber, and Hal Daumé III. 2021. [Multi-step reasoning over unstructured text with beam dense retrieval](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4635–4641, Online. Association for Computational Linguistics.

Appendix

A Related Work

In this section, we review researches closely related to our study. They are document retrieval methods for multi-hop queries and some studies about attention heads in embedding method.

Document Retrieval for Multi-Hop Queries.

RAG helps alleviate issues such as hallucination and outdated information in LLMs. However, in practical scenarios, complex multi-hop queries often fail to require sufficient evidence documents for generation. Existing document retrieval methods for multi-hop queries within the RAG framework are broadly classified into three paradigms: (1) using dense retrieval to improve the recall of multi-hop evidence; (2) explicitly generating intermediate reasoning steps to plan subsequent retrieval paths; and (3) leveraging external explicit structures to guide reasoning.

To improve the recall, early research mainly focused on dense retrieval-based methods. Classical architectures, such as DPR (Karpukhin et al., 2020) and ANCE (Xiong et al., 2021a), employ dual encoders to map queries and documents into a shared latent space, attempting to capture semantic relevance through similarity matching. GTR shows that scaling a single-vector dual encoder can substantially improve retrieval performance (Ni et al., 2022). ColBERTv2 retains token-level representations and applies late interaction for finer-grained query-document matching (Santhanam et al., 2022).

To bridge the logical gap, recent works tend to explicitly generate intermediate reasoning steps and then plan subsequent retrieval paths. For example, IRCOT (Trivedi et al., 2023) interleaves chain-of-thought (CoT) reasoning with retrieval steps and generates reasoning plans using LLM. ReAct (Yao et al., 2023) introduces an "action-observation" loop to dynamically adjust search strategies. Self-RAG (Asai et al., 2024) further incorporates a self-reflection mechanism to assess retrieval quality. LLMs may over-commit to early mistakes during sequential generation, allowing hallucinations to snowball (Zhang et al., 2023).

The third category of retrieval methods leverages *external explicit structures* to provide reasoning constraints. For example, GraphRAG (Edge et al., 2024) builds an LLM-derived entity graph and hierarchical community summaries during in-

dexing, then generates partial and final responses from those summaries for corpus-level questions. Following this direction, recent studies have explored more sophisticated graph structures to improve multi-hop reasoning. Yang et al. (2026) combine hierarchical knowledge graphs with RAG through graph convolutions that aggregate local and global relations for complex queries. Chen et al. (2026a) propose a Topic-Graph-based method for long-document, enhancing cross-paragraph evidence association by activating relevant sentences and their neighborhoods along the graph structure. Chen et al. (2026b) use relational paths as the basic unit and exploit explicit path constraints to guide multi-hop evidence selection in domain-specific QA. Overall, these methods provide clear logical and structural constraints for multi-hop retrieval, but their practical use is often limited by high construction and maintenance costs. They are not only highly sensitive to the quality of upstream text pre-processing, but also incur substantial update costs when external knowledge evolves dynamically.

The comparisons of the above methods are summarized in Table 2. The comparisons provides a qualitative comparison of paradigm-level design properties. We do not compare them in our experiments, since their final performance is influenced by multiple factors including generation, prompting, and graph construction.

Attention Heads of Embedding Models. Recently, studies in mechanistic interpretability try to open the black box of Transformer models. For example, study on induction heads Olsson et al. (2022) reveals that some attention heads can automatically locate previously observed patterns in the context and copy the information that follows, thereby supporting sequence-level pattern continuation and information transfer. By contrast, ROME (Meng et al., 2022) uses causal interventions to show that MLP layers are key loci for storing factual knowledge. These findings motivate our investigation of internal model signals for guiding external document retrieval during inference.

B Supplementary Experiments

B.1 Reproducibility Details

We use ChromaDB as the vector datastore. Collections are named {dataset}_{split}_{sample_size} and use an HNSW index with L2 distance. For 2Wiki and HotpotQA, sentences sharing the same

Table 2: Comparison of representative multi-hop retrieval methods.

Paradigm	Representative Methods	Non-Generative	No External Graph/Path	Stepwise Retrieval Guidance	Robust to Error Propagation
Dense Retrieval	DPR, ANCE, GTR-Large	✓	✓	✗	✓
Generative Reasoning-Guided Retrieval	IRCoT, ReAct, Self-RAG	✗	✓	✓	✗
External-Structure-Guided Retrieval	GraphRAG, Hierarchical KG-RAG, Topic-Graph RAG	✗	✗	✓	✓
IN-Retriever		✓	✓	✓	✓

title are concatenated into one document; for MuSiQue, paragraphs sharing the same title are aggregated. All documents are encoded with Qwen3-Embedding-4B before insertion into the collection.

The candidate samples for $\mathcal{D}_{\text{calib}}$ are drawn from the 2Wiki training split. We construct 2,000 candidate samples of the compositional type, each containing at least two complete evidence triples. The object entity in the first-hop evidence triple is used as the bridging clue E . During filtering, E and context C are lowercased, and a sample is retained only when E appears as a contiguous string in C . BFS computation first uses exact token-sequence matching to locate E and falls back to its lowercased token sequence when needed. Only samples producing valid BFS scans are aggregated to compute each attention head’s final BFS.

B.2 Stability of Elite Attention Heads

To verify the stability of elite attention heads, we design an experiment to examine whether elite attention head remains consistent across different datasets. To this end, we construct evaluation subsets from 2Wiki and MuSiQue. For 2Wiki, we retain only samples labeled as compositional and containing complete evidence triples. For MuSiQue, we use question decomposition annotations to extract samples with explicit bridging dependencies. Finally, we select 2,000 high-confidence samples from each dataset.

In the experiment, we extract the top-8 attention heads according to BFS score in layer $L = 23$, denoted as $\mathcal{H}_{2\text{Wiki}}$ and $\mathcal{H}_{\text{MuSiQue}}$, respectively. The Jaccard similarity coefficient is used as the evalua-

tion metric, as defined in Formula 9.

$$J(\mathcal{H}_{ds}, \mathcal{H}_{\text{elite}}) = \frac{|\mathcal{H}_{ds} \cap \mathcal{H}_{\text{elite}}|}{|\mathcal{H}_{ds} \cup \mathcal{H}_{\text{elite}}|} \quad (9)$$

where $ds \in \{2\text{Wiki}, \text{MuSiQue}\}$.

The experimental results are shown in Figure 6. The figure shows that, as the number of samples increases, the top- H attention heads identified from partial data subsets eventually align perfectly with the elite attention heads $\mathcal{H}_{\text{elite}}$. Specifically, on 2Wiki, complete overlap is achieved when the sample size reaches 100. Even on MuSiQue, full overlap is achieved once the sample size reaches 250. The results indicate that elite attention heads are consistent across different datasets, and exhibit a certain degree of cross-task stability instead of overfitting to a specific dataset. Second, this overlap converges rapidly even with only a small number of calibration samples, indicating that the calibration cost is relatively low.

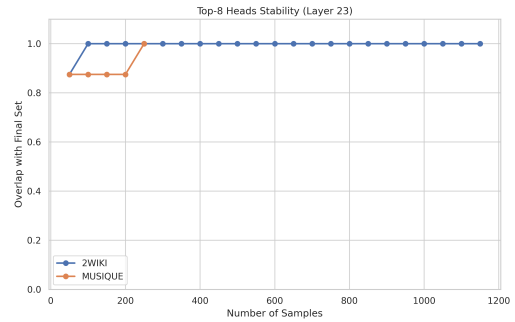


Figure 6: Overlap ratio with the elite attention heads as the number of calibration samples increases on 2Wiki and MuSiQue.

B.3 Beam Width Analysis

IN-Retriever adopts the beam search mechanism, which allows it to maintain multiple potential reasoning trajectories simultaneously, thereby enabling the model to tolerate local uncertainty. In practice, the number of candidate paths (beam width, denoted by B) may affect the final retrieval results. In this subsection, we evaluate both retrieval effectiveness and the number of deduplicated documents in the final output under different values of B , in order to guide the selection of B .

Figure 7 shows the curves between B and Recall@20 on three datasets. The curves exhibit an evident elbow between $B = 8$ and $B = 10$, beyond which the marginal gains become very limited. For example, Recall@20 on 2Wiki reaches 0.93 at $B = 8$, and subsequent improvement slowed down apparently. The results suggest that tracking the top- B highest-confidence paths is sufficient to cover the vast majority of critical evidence. Accordingly, we set $B = 8$ in the experiments.

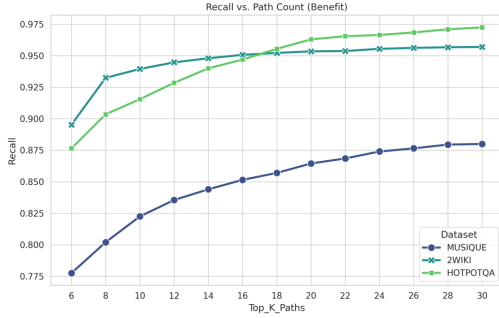


Figure 7: Curves of Recall@20 vs. beam widths B on MuSiQue, 2Wiki, and HotpotQA

Figure 8 shows the relation between the number of selected paths B and the number of deduplicated documents. As shown in the figure, although each path may contain multiple documents, the aggregate document count does not scale multiplicatively with B ; instead, it exhibits an approximately 1:1 growth trend with B . The result indicates that even though IN-Retriever retains more paths during reasoning, the number of documents does not diverge arbitrarily, but concentrates within a limited range.

The relation between path number and the size of documents further supports the selection of $B = 8$. Real effective reasoning paths tend to converge within the top-8 paths and point to the same correct answer. Additional paths beyond this range mostly diverge into noisier regions. Therefore, limiting the number of paths in IN-Retriever is not only

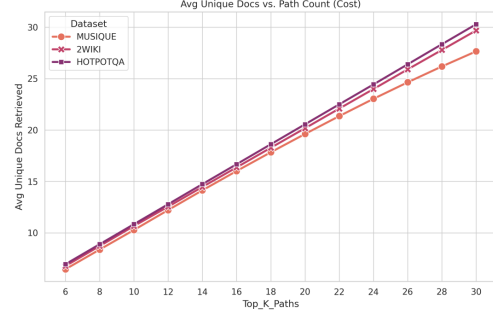


Figure 8: Average number of deduplicated documents versus path number B on 2Wiki, MuSiQue, and HotpotQA.

a matter of computational efficiency, but also an effective way to filter out long-tail noise and reduce reasoning drift.

B.4 Path-Level F1 under Path Budget B

In contrast to fixed-document-budget retrieval, IN-Retriever returns a variable number of deduplicated documents for each query. A flexible evaluation metric is helpful for evaluating retrieval quality under path-aware evidence aggregation. Therefore, we use $F1$ at B paths as an additional metric to further quantify the retrieval quality of IN-Retriever.

For each query, the final retrieval documents are collected from B paths. Let the corresponding gold document set be *Relevant*, and the set of documents finally returned by IN-Retriever be *Retrieved* at B paths. The number of correctly matched documents $Hit@B$, $Precision@B$, $Recall@B$, and $F1@B$ at B paths are defined as Formula 10, 11, 12, and 13, where B refers to B paths.

$$Hit@B = |Relevant \cap Retrieved@B|. \quad (10)$$

$$Precision@B = \frac{Hit@B}{|Retrieved|} \quad (11)$$

$$Recall@B = \frac{Hit@B}{|Relevant|} \quad (12)$$

$$F1@B = 2 \times \frac{Precision@B \times Recall@B}{Precision@B + Recall@B} \quad (13)$$

Assuming that the top- B paths contain R_B deduplicated documents, we extract the same number R_B of documents from each baseline and compute document-level $F1$ under this aligned document budget. This alignment ensures that all compared methods are evaluated under approximately comparable reading length and downstream computational budget, thereby reducing the possibility that

performance differences arise merely from returning more documents. The experimental results are shown in Table 3.

As shown in Table 3, IN-Retriever achieves the best F1 scores on MuSiQue and 2Wiki across all evaluated values of B , indicating that the proposed method improves not only evidence recall but also the balance between retrieved evidence coverage and retrieval compactness. On HotpotQA, IN-Retriever remains competitive but does not outperform SFR-Embedding-Mistral-7B, which is consistent with the relatively weaker multi-hop dependency of this dataset. These results further support that IN-Retriever is more advantageous on datasets where evidence must be connected through explicit bridging clues.

B.5 Normalized Weight Sensitivity Analysis

With the total local-score weight fixed at 35, we normalize the three weights as $(\alpha_v, \alpha_i, \alpha_s) = (\lambda_{\text{vec}}, \lambda_{\text{intent}}, \lambda_{\text{sym}})/35$. We vary α_v and α_i over $\{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$, set $\alpha_s = 1 - \alpha_v - \alpha_i$, and retain only combinations for which all three normalized weights are non-negative. Table 4 reports Path Recall@20 on the 2Wiki validation set.

Under the default setting, $(\lambda_{\text{vec}}, \lambda_{\text{intent}}, \lambda_{\text{sym}}) = (8, 7, 20)$ corresponds to the normalized ratio (0.229, 0.200, 0.571). The results suggest that the current setting lies in a relatively stable region on 2Wiki.

B.6 Lexical Contribution Ablation

The numerical weights in Eq. 7 alone cannot determine the actual contribution of each signal. We therefore compare *Full* with *w/o symbolic term* ($\lambda_{\text{sym}} = 0$, $\lambda_{\text{vec}} = 8$, and $\lambda_{\text{intent}} = 7$) and *Symbolic-only* on the three validation sets. The results are reported in Table 5.

Removing the symbolic term reduces Path Recall on all three datasets, showing that lexical calibration makes a practical contribution. Symbolic-only also performs below Full on all three datasets, with particularly large gaps on MuSiQue and HotpotQA. These results show that lexical matching does not independently drive the full model’s performance. The full model benefits from the complementary contributions of symbolic calibration, vector relevance, and residual-intent signals.

B.7 Instruction-Template Controls

To examine whether the effect comes only from the presence of an instruction-like prefix, we con-

duct experiments using different instruction templates while fixing the other retrieval settings. *Full* denotes the task-oriented instruction used in IN-Retriever; *T1*, *T2*, and *T3* denote generic, task-irrelevant, and wrong-direction instructions, respectively. The four templates are as follows.

- **Full task-oriented:** “Identify the key evidence in the context that helps find the next context to finally answer the question. Focus specifically on the entity type requested in the query.”
- **T1 Generic:** “Identify the relevant information in the context that helps understand the query before answering the question. Focus specifically on the content and wording provided in the context.”
- **T2 Task-irrelevant:** “Identify the stylistic information in the context that helps describe the writing before answering the question. Focus specifically on punctuation, capitalization, and sentence structure in the context.”
- **T3 Wrong-direction:** “Identify the final answer in the context that helps answer the question directly. Focus specifically on answer content rather than evidence needed to find the next context.”

The corresponding Path Recall@20 results are reported in Table 6.

T1’s generic instruction lacks explicit guidance on next-hop evidence and entity type. T2 redirects attention to textual style and therefore produces the largest decrease. T3 focuses on the final answer rather than the bridging evidence used to locate the next document, thereby weakening next-hop retrieval guidance. These results show that the task-oriented instruction is more consistent with the objective of our bridging clue probing and further support its effectiveness.

B.8 Attention Patterns across Different Embedding Models

To examine whether the BFS distribution pattern is specific to a particular backbone model, we repeat the experiment described in Section 3.2.1 on two other mainstream embedding models, namely Mistral and Llama. We keep the same instruction template, bridging clue annotation protocol, BFS definition, and sampling procedure. Based on the

Table 3: Comparison of retrieval methods in terms of document-level $F1$ under different path budgets B . The columns $B = 5, 10, 15, 20$ denote the number of selected reasoning paths used to construct the retrieved document set. Each reported value is the average $F1$ over all queries. The best performances are **bolded**, the second-best performances are underlined, and IN-Retriever is highlighted in gray.

Retrieval Methods	MuSiQue				2Wiki				HotpotQA			
	$B = 5$	$B = 10$	$B = 15$	$B = 20$	$B = 5$	$B = 10$	$B = 15$	$B = 20$	$B = 5$	$B = 10$	$B = 15$	$B = 20$
<i>Simple Baselines</i>												
BM25	25.14	16.71	12.98	10.68	36.34	24.31	18.36	14.75	38.17	27.43	20.63	16.43
GTR (T5-Large)	27.70	18.84	14.58	12.12	37.01	24.43	18.33	14.76	39.30	26.33	19.62	15.64
<i>Large Embedding Models</i>												
SFR-Embedding-Mistral-7B	<u>33.94</u>	<u>22.92</u>	<u>17.73</u>	<u>14.45</u>	<u>41.82</u>	<u>27.79</u>	<u>20.82</u>	<u>16.72</u>	45.55	29.82	21.83	17.21
Qwen3-Embedding-8B	33.81	22.88	<u>17.73</u>	14.37	41.81	27.44	20.69	16.62	<u>44.55</u>	<u>29.25</u>	21.55	17.02
<i>Baseline RAG</i>												
Qwen3-Embedding-4B (Direct)	33.04	22.16	17.15	14.13	41.08	27.01	20.23	16.24	43.21	28.87	21.28	16.83
IN-Retriever	40.16	26.33	19.60	15.90	50.81	34.49	25.64	20.38	43.84	29.03	<u>21.56</u>	<u>17.20</u>

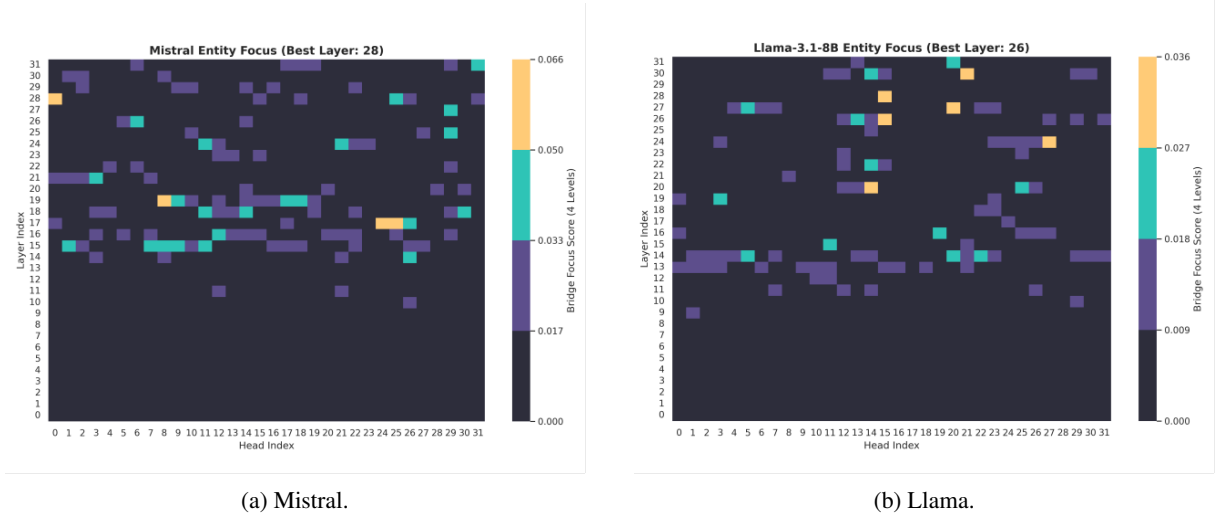


Figure 9: BFS heatmaps across all attention layers under two embedding backbones on the 2Wiki dataset.

Table 4: Fixed-total weight sensitivity on 2Wiki (Path Recall@20); dashes denote infeasible settings.

$\alpha_v \backslash \alpha_i$	0.0	0.2	0.4	0.6	0.8	1.0
0.0	85.28	94.83	94.17	93.95	93.70	0.00
0.2	95.73	95.80	95.90	95.58	90.20	—
0.4	95.80	95.93	95.93	91.27	—	—
0.6	95.75	95.80	94.33	—	—	—
0.8	95.58	94.17	—	—	—	—
1.0	93.80	—	—	—	—	—

Table 5: Controlled lexical ablation on the three validation sets (Path Recall@5 / Path Recall@20).

Dataset	Full	w/o symbolic term	Symbolic-only
MuSiQue	76.40 / 86.90	74.80 / 86.60	34.90 / 34.90
2Wiki	89.05 / 95.80	83.45 / 90.75	85.18 / 85.28
HotpotQA	86.65 / 96.60	84.75 / 95.90	53.70 / 53.70

Table 6: Instruction-template controls on the three validation sets (Path Recall@20).

Dataset	Full	T1	T2	T3
2Wiki	95.80	93.30	83.23	91.98
MuSiQue	86.90	86.85	83.65	86.80
HotpotQA	96.60	96.45	94.55	96.60

resulting BFS values, we visualize the results in Figure 9.

From Figure 9, we find that, consistent with the results of Qwen3-Embedding-4B, attention heads with high BFS values are mainly concentrated in the middle-to-late layers of each model, while such heads are extremely sparse in the shallow layers. It suggests that the layer-wise sensitivity of bridging-clue-related attention signals is not unique to a single backbone, but instead reflects a broader pattern

Table 7: Ablation study of IN-Retriever on three multi-hop datasets in terms of Recall@5.

Method	Components			Datasets (Recall@5)		
	Steering	Residual	Cumulative	MuSiQue	2Wiki	HotpotQA
IN-Retriever (Full Model)	✓	✓	✓	76.45	89.10	86.35
w/o Cumulative Scoring	✓	✓	–	75.05	82.95	84.20
w/o Residual-based Reranking	✓	–	✓	75.95	88.95	85.60
w/o Intrinsic Steering	–	✓	✓	66.75	71.10	77.95
Direct Retrieval (Baseline)	–	–	–	63.80	72.82	86.10

across architectures. It also indicates that internal signals relevant to bridging clues are more likely to emerge in layers where a certain degree of semantic integration has already taken place.

On the other hand, the figure shows that attention heads with high BFS values are not uniformly distributed across the entire parameter space, but are concentrated in only a small number of layers and heads. The observation is consistent with the sparse distribution of high-BFS heads in Qwen3-Embedding-4B, indicating that this pattern is unlikely to be an accidental property of a single model.

B.9 Ablation Study

To assess the contributions of each component, we conduct an ablation study by removing the three main modules one by one and report Recall@5 in Table 7.

After removing internal-attention-based clue extraction (w/o Intrinsic Steering), the fine-grained bridging clue e is no longer available for residual calculation and cumulative scoring. To keep the remaining modules executable, we replace e with a coarse document-level representation of the previous-hop document. In this setting, the model expands the evidence chain based on the entire previous document rather than on the specific clue localized by internal attention. As shown in Table 7, this variant leads to the largest performance drop across all datasets: Recall@5 decreases by 9.70, 18.00, and 8.40 percentage points on MuSiQue, 2Wiki, and HotpotQA, respectively. These results indicate that fine-grained bridging clues extracted from internal attention provide more targeted next-hop guidance than coarse document-level expansion, and constitute the primary source of the performance gains achieved by our method.

The cumulative scoring mechanism considers not only semantic relevance but also keyword

matching. As shown in Table 7, removing the cumulative scoring mechanism (w/o Cumulative Scoring) reduces Recall@5 by 1.40, 6.15, and 2.15 percentage points on the three datasets. The reductions suggest that relying solely on dense semantic similarity is insufficient to distinguish some key bridging entities in multi-hop retrieval. Explicit constraints introduced by keyword matching effectively complement this weakness and are helpful for the precise assessment of key entities and clues.

Finally, when removing the semantic residual step (w/o Residual Projection/Reranking), we disable the projection-based residual computation and remove its use in candidate reranking. In this variant, the model directly follows the documents retrieved by the bridging clue, without using the unresolved part of the original query as an additional constraint. As shown in Table 7, the Recall@5 of this variant consistently decreases across the three datasets. This suggests that the residual signal helps refine clue-based retrieval by reducing repeated focus on already covered evidence and by keeping the search aligned with the remaining information need of the original query. Therefore, although the bridging clue provides the main expansion direction, the residual step offers a useful complementary constraint for selecting more appropriate next-hop documents.

Overall, the ablation study verifies the indispensability of three core modules: internal-attention-based clue extraction, cumulative scoring mechanism, and semantic residual-based reranking.