Structure-Aware Quantized Retrieval for Long-Document Question Answering

Hui HUANG[1,2], **Julien Velcin**[3], **Yacine Kessaci** [2]
[1]Université de Lyon, Lyon 2, ERIC UR 3083, France
[2]Worldline S.A., France
[3]École Centrale de Lyon, LIRIS CNRS UMR 5205, France
hui.huang@univ-lyon2.fr, julien.velcin@ec-lyon.fr
yacine.kessaci@worldline.com

Abstract

Long-document question answering is challenging because relevant evidence is often scattered across distant sections. Traditional long-document QA/RAG pipelines often suffer from context fragmentation, retrieving locally plausible but structurally misaligned passages. We present the **H**ierarchical **Q**uantized **D**ocument **R**etriever (HQDR), a framework that aligns hierarchical graph representations with a universal token vocabulary and integrates explicit structure into retrieval. By grounding continuous structural features in a fixed, discrete semantic space, HQDR captures universal hierarchical patterns rather than overfitting to specific layouts. We further propose a hybrid scoring mechanism that decouples semantic matching from structural alignment. Extensive experiments on QASPER and Natural Questions demonstrate that HQDR consistently outperforms strong baselines and exhibits superior robustness when transferring across datasets with distinct structural characteristics.

1 Introduction

Long-document question answering (QA) presents a formidable challenge in natural language processing, requiring systems to identify and synthesize evidence often scattered across distant sections of a lengthy document. While the dominant Retrieval-Augmented Generation (RAG) paradigm (Lewis et al., 2020) has proven effective for knowledge-intensive tasks, standard approaches typically treat documents as flat sequences of chunks, as illustrated in Figure 1. This flattening process strips away critical structural context—such as hierarchical relationships between sections and paragraphs—often leading to the retrieval of semantically similar but contextually irrelevant fragments.

Although Large Language Models (LLMs) now support massive context windows, feeding full documents remains unreliable due to the "lost-in-the-middle" phenomenon (Liu et al., 2024) and prohibitive costs. This creates a critical bottleneck for enterprise applications, where the trade-off between operational cost and retrieval precision directly impacts the feasibility of large-scale deployment. Recent works like LongRAG (Zhao et al., 2024) attempt to balance global and local information, yet efficiently modeling latent structural dependencies remains an open challenge. While Graph Neural Networks (GNNs) can encode hierarchy, fusing continuous GNN outputs with Pre-trained Language Model (PLM) semantics is difficult; deep GNNs tend to over-smooth representations, blurring distinct structural roles (Oono and Suzuki, 2021).

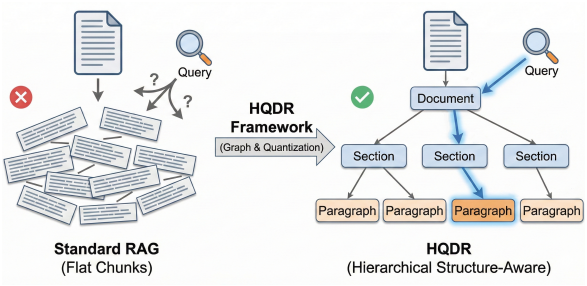

Figure 1: While traditional RAG (left) flattens documents into chunks, HQDR (right) preserves hierarchy via structure-aware graphs.

In this work, we propose the **H**ierarchical **Q**uantized **D**ocument **R**etriever (HQDR). Inspired by the quantization framework of STAG (Bo et al., 2025), which applied soft quantization for node classification in citation networks, we adapt this paradigm to the distinct challenges of retrieval by: (i) constructing hierarchical document graphs and fusing PLM semantics with a novel document-level contrastive objective; (ii) quantizing fused features into a universal discrete space anchored by a frozen LLM token codebook for robustness and transfer; and (iii) introducing a hybrid retrieval mechanism. Unlike standard dense retrieval, this mechanism utilizes a section meta-codebook—a set of discrete

Findings of the Association for Computational Linguistics: ACL 2026, pages 4291–4304

signatures representing the specific sections of a document—allowing the model to explicitly score the alignment between a query and a document's unique structural layout.

We validate HQDR on two benchmarks: QASPER (scientific papers) and Natural Questions (Wikipedia). Results show consistent gains over baselines. HQDR also exhibits cross-dataset transferability, indicating it captures universal hierarchical patterns rather than overfitting to specific domain layouts. Additionally, our approach maintains comparable latency to dense retrievers, offering a highly efficient alternative.

Our contributions are summarized as follows:

- We introduce HQDR, extending structural quantization to hierarchical graphs with a hierarchy-aware contrastive objective tailored for retrieval.
- We propose a hybrid strategy combining dense semantic similarity with a document-specific section meta-codebook, explicitly exposing structure to the scoring function.
- Experiments on QASPER and NQ demonstrate that HQDR consistently outperforms competitive baselines across diverse document layouts while maintaining high inference efficiency comparable to standard dense retrievers.

2 Related Work

In this section, we briefly review each line of research and clarify how HQDR differs from and builds upon prior approaches.

2.1 Structured Document Representation Learning

Handling long documents typically involves sparse attention mechanisms (e.g., Longformer (Beltagy et al., 2020), BigBird (Zaheer et al., 2020)) which treat texts as flat sequences, or hierarchical encoders (e.g., HAT (Chalkidis et al., 2022)) primarily designed for classification rather than retrieval. To explicitly capture dependencies, recent approaches inject structure into PLMs via auxiliary pre-training objectives (SANTA (Li et al., 2023), SEAL (Huang et al., 2025)), leverage external graphs such as AMR (Wang et al., 2023), or employ explicit graph modeling (TextGCN (Yao et al., 2019), G-DSR (Louis et al., 2023)). However, purely continuous approaches struggle to expose explicit structure at inference (Buchmann et al., 2024), while deep GNNs suffer from over-smoothing (Oono and Suzuki, 2021; Bo et al., 2025). HQDR mitigates this by anchoring continuous structural features to a discrete meta-codebook and explicitly exposing global relations to the scoring function, thereby enhancing robustness.

2.2 Vector Quantization and Discrete Representations

Vector Quantization (VQ) bridges continuous and discrete latent spaces. While originally established for discrete representation learning (van den Oord et al., 2017), it has evolved into a cornerstone of contemporary multimodal systems (Li et al., 2025). In graph learning, STAG (Bo et al., 2025) introduced structure-aware quantization to mitigate GNN variance collapse. However, STAG treats quantization as a representation goal rather than a retrieval interface. HQDR extends this paradigm by introducing a section-level meta-codebook and integrating discrete assignments directly into the ranking objective. This transforms quantization from a generic regularizer into a task-aware mechanism that aligns structural context with retrieval needs.

2.3 Long-Document QA and Retrieval

Long-document QA demands synthesizing dispersed evidence, tracked by benchmarks ranging from NQ (Kwiatkowski et al., 2019) and QASPER (Dasigi et al., 2021) to massive context evaluations (Bai et al., 2024). Although modern LLMs with extended context windows allow feeding entire documents as input, they incur high inference costs and suffer from the "lost-in-the-middle" phenomenon (Liu et al., 2024). Conversely, standard RAG pipelines (Lewis et al., 2020) offer efficiency but suffer from context fragmentation.

To address this, recent structure-aware and hierarchical RAG methods have emerged. For instance, RAPTOR (Sarthi et al., 2024) and TreeRAG (Tao et al., 2025) construct hierarchical tree structures by recursively summarizing text chunks, while methods like LongRefiner (Jin et al., 2025) focus on refining retrieved contexts or modeling inter-chunk relations. Similarly, SEAL (Huang et al., 2025) injects structure into PLMs via auxiliary pre-training objectives. While these document-aware strategies successfully incorporate hierarchical metadata, they often add significant architectural complexity

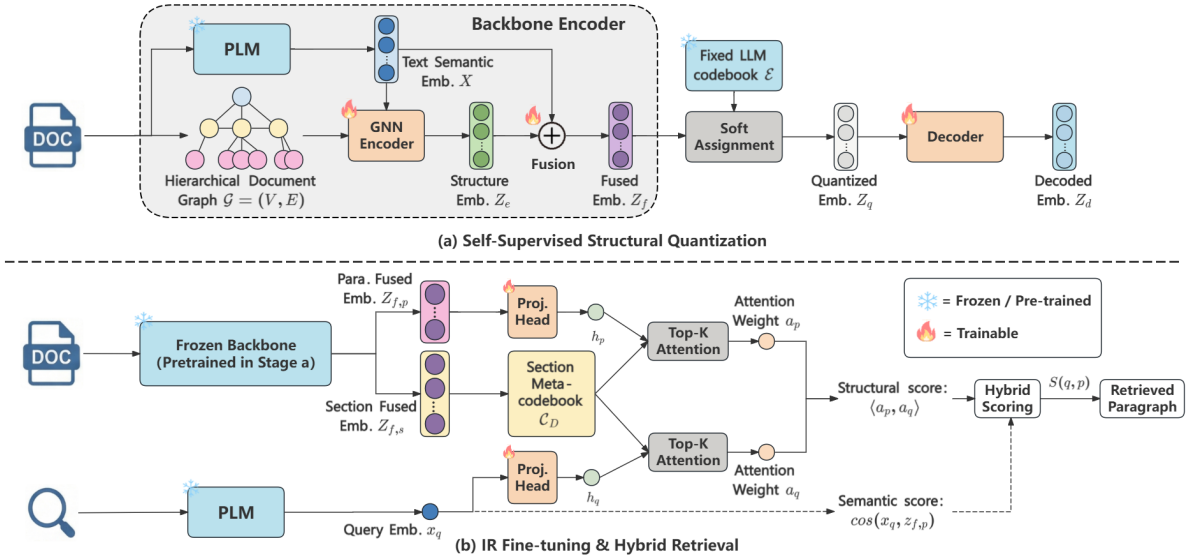

Figure 2: Overview of HQDR. **(a) Self-Supervised Quantization:** Fuse PLM and GNN features, aligning them with a fixed LLM codebook via a decoder-based reconstruction objective. **(b) Hybrid Retrieval:** Freeze the backbone and utilize a document-specific section meta-codebook. Trainable projection heads generate sparse structural profiles via Top-k attention, which are combined with dense semantics for final scoring.

and rely on computationally expensive recursive LLM calls. HQDR bridges this gap by explicitly exposing structure to the scoring function without recursive summarization. By decoupling offline structural learning, HQDR captures the complex structural dependencies while maintaining the modularity, cost-effectiveness, and high operational efficiency of standard dense RAG pipelines.

3 Method

We present the **H**ierarchical **Q**uantized **D**ocument **R**etriever (HQDR), a framework that connects structure-aware graph representations with discrete semantic spaces for long-document retrieval.

As illustrated in Figure 2, HQDR operates in two main stages: (i) Self-Supervised Structural Quantization, where the backbone fuses PLM semantics with hierarchical graph features and aligns them with a fixed LLM codebook; and (ii) IR Fine-tuning & Hybrid Retrieval, where the frozen backbone is utilized with a document-specific section meta-codebook to perform structure-aware hybrid scoring.

3.1 Preliminaries

3.1.1 Task Definition

We study long-document retrieval for QA. Let $\mathcal{D} = \{D_1, \ldots, D_N\}$ be a corpus of long documents. Each document $D \in \mathcal{D}$ comes with an inherent hierarchical and sequential structure: it has a title T, an ordered sequence of sections $\mathcal{S} = (S_1, \ldots, S_m)$, and $\mathcal{P}_j = (P_{j,1}, \ldots, P_{j,n_j})$ is the ordered sequence of paragraphs belonging to section S_j. We denote by $\mathcal{P}_D = \{P_{j,k}\}_{j,k}$ the set of all paragraphs in D.

Given a natural language query q and a document D, a paragraph retriever assigns a relevance score $S(q, p)$ to each paragraph $p \in \mathcal{P}_D$ and identifies the top-k most relevant paragraphs through a ranked list, ranked by decreasing order of the relevance score. In standard passage retrieval, documents are often treated as flat collections of paragraphs. In our setting, the retriever is expected to leverage the explicit document structure to disambiguate context more effectively in long-document QA.

3.1.2 Document Graph Construction and Feature Fusion

Graph construction. We model each document D as a hierarchical graph $G = (V, E)$. The node set V contains three types of nodes: a single document node v_d, section nodes $v_s \in V^{(S)}$, and paragraph nodes $v_p \in V^{(P)}$. $V^{(S)}$ and $V^{(P)}$ denote the sets of section and paragraph nodes. Edges E encode hierarchical (e.g., $v_d \leftrightarrow v_s$) and sequential (e.g., $v_{p_i} \leftrightarrow v_{p_{i+1}}$) relations, enabling global-to-local information flow.

Feature fusion. Let R_i denote the raw text associated with node i (title, section heading, or paragraph body). As shown in Figure 2(a), we first obtain semantic features using a frozen PLM optimized for sentence embeddings (e.g., MPNET (Song et al., 2020) implemented via the SentenceTransformers framework (Reimers and Gurevych, 2019)):

$$x_i = \mathrm{PLM}(R_i) \in \mathbb{R}^d \tag{1}$$

where d is the PLM hidden size. Collecting all node features yields $X = \{x_i\}_{i \in V}$. To inject structure, we employ a 3-layer Graph Attention Network (GAT) (Veličković et al., 2018) to aggregate neighbors with adaptive weighting, yielding structural embeddings:

$$Z_e = \mathrm{GNN}(G, X), \tag{2}$$

where $Z_e = \{z_{e,i}\}_{i \in V}$, $z_{e,i} \in \mathbb{R}^{d'}$ are structure-aware node embeddings. We then construct a fused representation $z_{f,i}$ that preserves the semantic direction while injecting structure:

$$z_{f,i} = \phi \cdot \frac{W_f z_{e,i}}{\|W_f z_{e,i}\|_2} + (1 - \phi) \cdot \frac{x_i}{\|x_i\|_2}, \tag{3}$$

where W_f projects structural features and $\phi \in [0, 1]$ is a learnable scalar.

3.2 Self-Supervised Structural Quantization

The objective of this phase is to learn a universal encoder that, given a document graph, (i) captures its hierarchical and sequential structure, and (ii) preserves the original semantic information encoded by the frozen PLM. Inspired by STAG (Bo et al., 2025), a framework originally proposed for node classification in citation networks, we quantize graph nodes into an LLM's fixed vocabulary.

In HQDR, we adapt STAG to the setting of hierarchical document graphs and long-document retrieval. Concretely, we introduce two key modifications: (i) adding a hierarchy-aware contrastive objective (Sec. 3.2.2) and (ii) enabling hybrid scoring via a section meta-codebook (Sec. 3.3).

3.2.1 Soft Assignment Quantization

Following STAG, we define a fixed codebook $\mathcal{E} = \{e_k\}_{k=1}^K$, where $e_k \in \mathbb{R}^d$ lies in the same embedding space as node features. Specifically, we take the LLaMA-2 tokenizer to obtain a list of vocabulary tokens $\{t_k\}_{k=1}^K$, then pass each token through the same frozen sentence-level PLM used for node initialization:

$$e_k = \mathrm{PLM}(t_k), \quad k = 1, \ldots, K. \tag{4}$$

Keeping the codebook frozen ensures that the learned representations remain grounded in a universal semantic space.

Next, given the fused representation $z_{f,i}$ of a node i, our goal is to express this node in the discrete LLM token space. Instead of a hard assignment, which may lose information (Bo et al., 2025), we follow STAG and represent each node as a soft assignment over codebook entries. The probability of assigning node i to token k is given by temperature-scaled cosine similarity:

$$p_k(z_{f,i}) = \frac{\exp\big(\cos(z_{f,i}, e_k)/\tau\big)}{\sum_{j=1}^K \exp\big(\cos(z_{f,i}, e_j)/\tau\big)}, \tag{5}$$

where $\tau > 0$ is a temperature hyperparameter that controls the softness of the assignment and $\cos(\cdot, \cdot)$ denotes cosine similarity.

The quantized representation $z_{q,i}$ is then the expectation over codebook entries:

$$z_{q,i} = \sum_{k=1}^K p_k(z_{f,i})\, e_k. \tag{6}$$

We denote $Z_q = \{z_{q,i}\}_{i \in V}$ for all quantized node representations.

3.2.2 Optimization Objectives

We train the GNN encoder and fusion parameters with a composite loss that combines reconstruction, commitment, and semantic-alignment terms, and extend it with a document-aware contrastive objective. The total loss $\mathcal{L}$ is defined as:

$$\mathcal{L} = \mathcal{L}_{\mathrm{Rec}} + \mathcal{L}_{\mathrm{Doc}} + \mathcal{L}_{\mathrm{Commit}} + \lambda \mathcal{L}_{\mathrm{KL}}, \tag{7}$$

where λ is a balancing parameter. We detail the specific contributions of these components in the following paragraphs.

Semantic loss. To ensure the quantized representations retain the rich semantics of the frozen PLM, we adopt the objectives from (Bo et al., 2025). Specifically, we employ: (i) a reconstruction loss $\mathcal{L}_{\mathrm{Rec}}$ to ensure the quantized feature $z_{q,i}$ can recover the original text embedding x_i; (ii) a commitment loss $\mathcal{L}_{\mathrm{Commit}}$ to constrain the continuous fused embedding $z_{f,i}$ to remain close to its discrete counterpart; and (iii) a KL alignment loss

$\mathcal{L}_{\mathrm{KL}}$ to align the soft assignment distribution of the fused representation $p(z_{f,i})$ with the original semantic quantization $p(x)$. As these terms are adopted directly from STAG, we detail their exact mathematical formulations in Appendix B to focus our discussion on the retrieval-specific structural adaptation below.

Structure loss. Unlike STAG, which only considers local neighborhood relations in generic graphs, hierarchical documents come with an explicit parent–child structure. We introduce an InfoNCE contrastive loss $\mathcal{L}_{\mathrm{Doc}}$ between parent and child nodes in the document graph. For a parent node u and its child v (e.g., document–section or section–paragraph), we treat (u, v) as a positive pair and other parent nodes in the batch as negatives (see details in Appendix A.4):

$$\mathcal{L}_{\mathrm{Doc}} = -\frac{1}{|E_{\mathrm{pc}}|} \sum_{(u,v)\in E_{\mathrm{pc}}} \log \frac{\exp(s_{u,v}/\tau_{\mathrm{doc}})}{\sum_{w\in\mathcal{N}(u)} \exp(s_{u,w}/\tau_{\mathrm{doc}})}, \tag{8}$$

where $s_{u,v} = \cos(z_{f,u}, z_{f,v})$, E_{pc} is the set of parent–child edges and $\mathcal{N}(u)$ denotes the set of all candidates for u in the batch, including the positive child v and negative nodes.

3.3 IR Fine-tuning and Hybrid Retrieval Strategy

After the self-supervised pre-training stage, we obtain a universal encoder that integrates document structure and semantics. We then design a hybrid strategy (Figure 2(b)), which combines dense similarity in the original backbone space with an interpretable structural score derived from a document-specific section meta-codebook. The key idea is to keep dense similarity in the original backbone space, while deriving an additional, interpretable structural score from a section-level meta-codebook built for each document.

3.3.1 Section Meta-Codebook

While the pre-training phase utilizes a global LLM codebook $\mathcal{E}$ (cf. Sec. 3.2) for general alignment, retrieval requires capturing the specific hierarchical layout of each target document. To this end, we introduce a section meta-codebook $\mathcal{C}_D$ for each document D by collecting the frozen fused embeddings of all its sections:

$$\mathcal{C}_D = \{\mathbf{c}_s = z_{f,v_s} \mid s \in \mathcal{S}\}. \tag{9}$$

where z_{f,v_s} is the structure-aware embedding of the section s derived from Eq. (3). Unlike the fixed vocabulary codebook $\mathcal{E}$, $\mathcal{C}_D$ varies across documents, serving as a set of local structural anchors.

To align queries and paragraphs with these local anchors, we employ a lightweight two-layer MLP projection head $g(\cdot) : \mathbb{R}^d \to \mathbb{R}^d$. This projection adapts the frozen representations specifically for the retrieval task. For a candidate paragraph p with fused feature z_{f,v_p}, and an input query q with PLM-encoded feature x_q, we compute their projected forms:

$$\mathbf{h}_p = g(z_{f,v_p}), \quad \mathbf{h}_q = g(x_q). \tag{10}$$

We then compute sparse structural profiles to represent the probabilistic alignment of the query and paragraph with specific document sections. To ensure sparsity and focus on the most relevant structural contexts, we apply a Top-k Softmax operation:

$$\mathbf{a}_p = \mathrm{softmax}\left(\text{Top-}k\big(\{\cos(\mathbf{h}_p, \mathbf{c}_s)\}_{s\in\mathcal{S}}\big)\right), \tag{11}$$

where Top-$k(\cdot)$ retains the highest k scores and masks the rest. Similarly, we compute the query's structural profile $\mathbf{a}_q$ against $\mathcal{C}_D$.

The Top-k operator is introduced to enforce sparsity in the structural profiles. A standard softmax over all sections would assign non-trivial probability mass to many sections for each query or paragraph, making the structural signal diffuse and less discriminative. By retaining only the k most similar sections, we force each query and paragraph to commit to a small set of candidate structural contexts, which empirically yields clearer query–section–paragraph alignments and improved retrieval performance.

These profiles, $\mathbf{a}_p$ and $\mathbf{a}_q$, serve as sparse "structural attention maps" that encode the alignment between the query, the paragraph, and the document's specific sections. This yields a query $\to$ relevant sections $\to$ paragraphs retrieval view: the quantized structural profiles allow us to first identify which sections of a document a query is most likely targeting, and then prioritize paragraphs whose profiles are consistent with those sections. The resulting structural score is used as an auxiliary signal alongside standard dense retrieval.

3.3.2 Hybrid Scoring and Optimization

The final score decouples semantic and structural signals:

$$S(q, p) = \alpha \cdot \cos\big(x_q, z_{f,p}\big) + (1 - \alpha) \cdot \langle a_q, a_p \rangle, \tag{12}$$

where $\alpha \in [0, 1]$ is a trainable parameter weights the contributions, $\langle \cdot, \cdot \rangle$ denotes the inner product over the discrete top-k vectors. We finetune only the structural projection head $g(\cdot)$ and α, keeping the PLM, GNN, and the universal codebook frozen.

To train the hybrid scorer for the IR task, we adopt the InfoNCE loss to pull the positive passages closer to the question. Let $\mathcal{P}^+$ be the set of positive paragraphs for query q, $\mathcal{P}^-$ be a set of negatives, and τ_{ir} be the temperature. The loss is:

$$\mathcal{L}_{\text{IR}} = -\frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \log \frac{\sum_{p \in \mathcal{P}^+} \exp\big(S(q,p)/\tau_{\text{ir}}\big)}{\sum_{p \in \mathcal{P}^+ \cup \mathcal{P}^-} \exp\big(S(q,p)/\tau_{\text{ir}}\big)}, \quad (13)$$

We report the sampling strategy in Appendix A.4.

3.4 Inference

At inference time, efficiency is critical. HQDR supports offline indexing and lightweight online scoring. Offline indexing separates static document feature computation from dynamic query processing, enabling efficient retrieval without repeated long-sequence encoding.

Offline indexing. For each document $D \in \mathcal{D}$, we: (i) run the frozen backbone to obtain $z_{f,s}$ for section nodes and $z_{f,p}$ for paragraph nodes; (ii) construct the section meta-codebook $\mathcal{C}$ with $c_s = z_{f,s}$; (iii) compute $h_p = g(z_{f,p})$ and store the discrete top-k structural profile a_p.

Online retrieval. Given a query q, we compute x_q with the frozen PLM, obtain $h_q = g(x_q)$, and derive its discrete top-k profile a_q against each document's $\mathcal{C}$. We then evaluate $S(q, p)$ for candidate paragraphs using efficient vector operations over the pre-computed indices.

4 Experiments

4.1 Experimental Setup

Datasets. We use two long-document retrieval benchmarks with complementary structural characteristics, chosen to test both in-domain effectiveness and cross-domain transfer.

QASPER (Dasigi et al., 2021) consists of 5,049 questions over 1,585 NLP papers. Built on full scientific texts, it provides long inputs with explicit hierarchical structure, making it well-suited for evaluating structure-aware retrieval.

Natural Questions (NQ) (Kwiatkowski et al., 2019) is an open-domain benchmark based on real user queries and Wikipedia pages. The original dataset emphasizes realistic questions and long/short answers, but many pages are relatively short or weakly structured. To adapt NQ to long-document retrieval and ensure sufficient structural complexity, we construct a structurally curated subset: (i) we start from the NQ-dev split to maintain a size comparable to QASPER; (ii) for each document, we clean and convert the HTML source into a QASPER-like format with explicit section and paragraph boundaries; (iii) we discard documents with fewer than six sections, retaining only pages with meaningful internal hierarchy for graph construction.

The final processed subset contains 3,211 documents, each with its corresponding question and evidence paragraphs. This filtering yields harder retrieval scenarios where multiple structurally similar sections may contain semantically related content, making the correct structural context crucial. We use official splits for QASPER and a random 70/20/10 split for this curated NQ subset.

Baselines and Metrics. We compare HQDR against a diverse set of strong retrievers: (i) **BM25** (Robertson and Zaragoza, 2009) as a sparse baseline; (ii) **ColBERT** (Khattab and Zaharia, 2020), a state-of-the-art late-interaction retriever; and (iii) two dense retrievers, **MPNet** (Song et al., 2020) and **BGE-base** (Xiao et al., 2023), which also serve as the semantic PLM backbone for HQDR.

To disentangle the effect of structure-aware quantization from generic task-specific training, we additionally include: (i) *fine-tuned* MPNet/BGE baselines, trained as standard passage-level dense retrievers on QASPER or NQ under the same data, epochs, and negative sampling scheme as HQDR (rows with "Train = QASPER/NQ" in Table 1); and (ii) a hierarchical retrieval baseline, **MPNet + RAPTOR** and **BGE-base + RAPTOR**, adopting the retrieval module and collapsed-tree mechanism from RAPTOR (Sarthi et al., 2024).

For HQDR, we report two variants to perform ablation: *"+ structure"* utilizes the fused representations Z_f (cf. Sec. 3.2) but relies solely on dense cosine similarity for scoring; *"+ meta-codebook"* represents our full model, replacing pure dense scoring with the hybrid dense–sparse strategy described in Section 3.3.

We report Hit@K ($K \in \{1, 5, 10\}$), MRR@10, and NDCG@10. All results are averaged across

Method	Train	Target: QASPER					Target: NQ				
		Hit@1	Hit@5	Hit@10	MRR	NDCG	Hit@1	Hit@5	Hit@10	MRR	NDCG
Baselines											
BM25	–	0.151	0.440	0.579	0.278	0.296	0.220	0.582	0.749	0.377	0.418
ColBERT	–	0.292	0.624	0.734	0.438	0.446	0.361	0.607	0.684	0.494	0.482
MPNet	–	0.203	0.539	0.710	0.345	0.365	0.409	0.737	0.854	0.550	0.587
–	QASPER	0.245	0.575	0.764	0.403	0.424	0.401	0.741	0.852	0.548	0.581
–	NQ	0.201	0.543	0.725	0.341	0.373	0.454	0.770	0.867	0.609	0.615
+ RAPTOR	–	0.221	0.541	0.721	0.353	0.372	0.415	0.748	0.864	0.562	0.595
+ Structure	QASPER	0.223	0.551	0.744	0.357	0.382	0.411	0.759	0.857	0.557	0.591
+ Meta-Codebook	QASPER	0.267	0.631	0.793	0.424	0.458	0.429	0.750	0.864	0.567	0.586
+ Meta-Codebook	NQ	0.216	0.587	0.742	0.368	0.410	**0.531**	**0.798**	0.877	**0.645**	**0.631**
BGE-base	–	0.295	0.640	0.796	0.440	0.452	0.351	0.741	0.861	0.514	0.563
–	QASPER	0.302	0.653	0.806	0.454	0.478	0.357	0.754	0.863	0.518	0.566
–	NQ	0.262	0.597	0.783	0.408	0.428	0.401	0.785	**0.892**	0.565	0.600
+ RAPTOR	–	0.295	0.647	0.804	0.445	0.459	0.361	0.752	0.860	0.523	0.570
+ Structure	QASPER	0.296	0.657	0.797	0.444	0.458	0.353	0.745	0.865	0.517	0.564
+ Meta-Codebook	QASPER	**0.312**	**0.689**	**0.821**	**0.512**	**0.489**	0.358	0.776	0.873	0.541	0.575
+ Meta-Codebook	NQ	0.305	0.660	0.803	0.501	0.465	0.399	0.787	0.885	0.553	0.584

Table 1: Retrieval performance of HQDR on QASPER and NQ datasets compared to sparse and dense baselines. **Bold** indicates the best result, and underline indicates the second best. "+ structure" denotes ablation using fused features with dense scoring; "+ meta-codebook" denotes the full hybrid retrieval model. Note that rows train on one dataset and evaluate on the other demonstrate the model's cross-dataset transferability.

three runs with different random seeds, all using the same evaluation protocol.

Implementation details. HQDR operates in two stages: self-supervised structural pre-training and supervised retrieval finetuning. During pre-training, we update the backbone using document graphs derived from the target dataset (QASPER or NQ) via unsupervised objectives (cf. Sec. 3.2). For the finetuning stage, we freeze the backbone and update only the lightweight projection head and scoring weights using the dataset's curated QA pairs. For the Top-k Softmax in Eq. (11), we set $k = 4$ based on a small grid search on the QASPER validation set, which provides the best trade-off between sparsity and robustness.

All baseline models (MPNet, BGE) are evaluated using standard open-source checkpoints. For ColBERT, we use ColBERTv2 via RAGatouille[1] with per-document indexing, and for RAPTOR, we use its public implementation,[2], replacing its original encoder with MPNet/BGE to ensure comparability. Detailed hyperparameters and hardware specifications are provided in Appendix A.

[1]https://ben.clavie.eu/ragatouille/

[2]https://github.com/parthsarthi03/raptor/tree/master

4.2 Main Results

Table 1 summarizes retrieval performance on QASPER and the curated NQ subset. Rows labeled "Train = QASPER" or "Train = NQ" indicate the dataset on which each dense model (or HQDR variant) is trained; the "Target" columns report performance when evaluating on each dataset, allowing us to test both in-domain effectiveness and cross-domain transfer.

Effect of intra-document structure. On both QASPER and NQ, the "+ structure" variants consistently outperform the corresponding PLM baselines (MPNet, BGE), confirming that fusing hierarchical graph features with PLM embeddings improves retrieval even when only dense scoring is used. Gains are particularly pronounced on QASPER, where the document hierarchy is clean and rich, indicating that explicit intra-document structure is beneficial for long scientific texts.

Effect of quantization and hybrid scoring. The full "+ meta-codebook" models further improve Hit@10, MRR, and NDCG over the "+ structure" variants for both MPNet and BGE backbones. The section meta-codebook allows HQDR to favor paragraphs whose structural profiles align with the query's section-level intent, retrieving relevant

paragraphs that pure dense scoring misses and validating the hybrid dense–sparse design.

Cross-dataset transfer. Rows where the training and target datasets differ show strong cross-domain transfer: HQDR trained on QASPER achieves the best results on QASPER and remains competitive on NQ, while HQDR trained on NQ achieves the best results on NQ and remains competitive on QASPER. This robustness stems from two design choices: (i) the universal token codebook anchors pre-training in a shared semantic space, mitigating domain overfitting; and (ii) decoupling semantic and structural scoring allows the meta-codebook to adapt to new document structures without retraining the backbone.

Comparison with ColBERT. While ColBERT performs well on QASPER, HQDR with the BGE backbone surpasses it on all metrics. On NQ, ColBERT's performance degrades more noticeably, whereas HQDR maintains robustness. This suggests that explicit hierarchical modeling offers a stronger inductive bias for irregular web structures than implicit token-level interaction alone.

Comparison with fine-tuned dense and hierarchical baselines. Fine-tuning MPNet and BGE-base as standard dense retrievers substantially improves their performance over off-the-shelf checkpoints on both QASPER and NQ (e.g., on QASPER, MPNet Hit@10 increases from 0.710 to 0.764; on NQ, MPNet MRR increases from 0.550 to 0.609). However, HQDR with the meta-codebook remains superior: BGE-base + Meta-codebook (trained on QASPER) achieves Hit@10 = 0.821 and MRR = 0.512 on QASPER, clearly outperforming the corresponding fine-tuned dense baseline. RAPTOR-style hierarchical retrieval also improves over the dense baselines, confirming the value of hierarchical indexing, but HQDR + Meta-codebook consistently yields larger gains (e.g., for MPNet on QASPER, Hit@10 = 0.721 vs. 0.793). These comparisons indicate that HQDR's advantages arise from structure-aware quantization and hybrid scoring, rather than from leaving dense baselines under-trained.

4.3 Ablation Studies

We further investigate the impact of key components. Unless otherwise specified, we employ MPNet as the default PLM and train on Qasper for all HQDR variants. All experiments share the same graph construction and fusion framework from Section 3 and are evaluated on QASPER or NQ-Hard (cf. Sec. 4.3.4).

Method	QASPER				
	Hit@1	Hit@5	Hit@10	MRR@10	NDCG@10
MPNet	0.203	0.539	0.710	0.345	0.365
HQDR	**0.267**	**0.631**	**0.793**	**0.424**	**0.458**
w/o codebook	0.221	0.568	0.723	0.363	0.389
w meta-codebook	0.208	0.550	0.726	0.354	0.387

Table 2: Impact of LLM token codebook on HQDR performance on QASPER. The universal token codebook anchors the semantic space during pre-training, enhancing retrieval beyond graph structure alone, while early inclusion of the meta-codebook may impact transferability.

Method	Qasper				
	Hit@1	Hit@5	Hit@10	MRR@10	NDCG@10
MPNet	0.203	0.539	0.710	0.345	0.365
HQDR	**0.267**	**0.631**	**0.793**	**0.424**	**0.458**
Dense only	0.226	0.593	0.758	0.382	0.420
Sparse only	0.180	0.480	0.651	0.306	0.353
w/o Projection Head	0.216	0.573	0.750	0.371	0.412

Table 3: Evaluation of different scoring strategies for HQDR on QASPER. The hybrid dense-sparse retrieval approach improves precision and recall by effectively combining semantic and structural signals.

4.3.1 Effectiveness of LLM Token Codebook

Table 2 compares MPNet, full HQDR, HQDR without the token codebook ("w/o codebook"), and a variant that replaces the universal token codebook with a section meta-codebook during pre-training ("w meta-codebook").

The universal token codebook contributes beyond regularization: it anchors graph-fused representations to a stable semantic basis, improving discriminability and cross-dataset transfer. Without the codebook, gains over the PLM baseline persist but shrink, indicating over-reliance on topology. Using a document-level meta-codebook during pre-training hampers generality: localized codes overfit structure seen in training, while reserving the meta-codebook for retrieval-time profiling preserves transfer.

4.3.2 Effectiveness of Hybrid Retrieval Strategy

Table 3 evaluates HQDR's hybrid retrieval strategy on QASPER by comparing three scoring variants, all using Eq.(12): a dense-only variant (fixing $\alpha = 1$), a sparse-only variant (fixing $\alpha = 0$), and a

variant without the projection head. In each variant, we isolate key components of HQDR to test their respective influences.

Hybrid scoring consistently outperforms dense-only and sparse-only variants: dense similarity captures fine-grained meaning, whereas the structural term resolves section-level ambiguity. The variant without the projection head yields moderate results, highlighting that explicitly projecting structural features into the ranking mechanism is crucial for optimizing relevance scoring.

4.3.3 Computational Efficiency Analysis

Method	Doc. Indexing	Struct. Overhead	Retrieval	Total (ms)
BM25	-	-	2.4	2.4
ColBERT	921.8	-	526.9	1448.7
MPNet	238.7	-	11.3	250.0
HQDR	240.6	23.8	12.7	277.1

Table 4: End-to-end latency analysis on QASPER. Doc. Indexing denotes PLM encoding for new documents; Struct. Overhead refers to GNN/Graph construction; Retrieval includes query encoding and scoring.

To assess deployability in cold-start scenarios, we benchmark the end-to-end latency against BM25, MPNet, and ColBERT. We report the total processing time for a query-document pair, decomposing costs into base encoding (Doc. Indexing), structural overhead (GNN computation), and the final retrieval phase. As shown in Table 4, ColBERT incurs substantial latency (>1.4s) driven by heavy token-level interactions. In contrast, HQDR preserves the efficiency of dense retrieval, adding only marginal overhead (23.8ms for graph encoding and 1.4ms for hybrid scoring) to the MPNet baseline. This contrasts sharply with full-document LLM prompting, where Transformer self-attention incurs quadratic time and memory growth with respect to input length, rendering long-context inference substantially slower (Vaswani et al., 2017; Keles et al., 2023). Consequently, HQDR achieves a total latency of 277.1ms—approximately 5× faster than ColBERT—demonstrating that it can leverage robust structural signals without prohibitive computational costs.

4.3.4 Performance on Hard Query Subsets

We further evaluate structural reasoning on the NQ-Hard subset from DAPR (Wang et al., 2024), applying our filtering criteria (cf. Sec. 4.1) to retain complex QA pairs anchored in long documents. HQDR consistently outperforms baselines on this subset,

Method	NQ-Hard				
	Hit@1	Hit@5	Hit@10	MRR@10	NDCG@10
BM25	0.128	0.333	0.551	0.221	0.266
ColBERT	0.128	0.244	0.513	0.194	0.235
MPNet	0.175	0.407	0.608	0.284	0.330
HQDR	**0.253**	**0.451**	**0.657**	**0.323**	**0.378**

Table 5: Retrieval performance on the NQ-Hard subset. HQDR (MPNet backbone) is compared against baselines. ColBERT struggles with these structurally complex queries, while HQDR remains robust.

whereas ColBERT struggles to match sparse BM25 performance. This performance gap highlights a specific challenge in complex queries: the target paragraph shares high semantic similarity with distracting passages distributed across different sections. While purely semantic models struggle to distinguish these false positives due to their semantic proximity, HQDR leverages explicit structural signals to filter out these semantically plausible but structurally irrelevant false positives.

5 Conclusion and Future Work

We presented HQDR, a structure-aware retrieval framework that bridges hierarchical document graphs and discrete semantic spaces for long-document QA. Our approach addresses two key challenges: (i) integrating explicit document structure into retrieval while remaining compatible with strong dense encoders, and (ii) enabling cross-dataset transfer between domains with very different document organizations. Extensive experiments show consistent gains over baselines, and highlight that our decoupled semantic/structural design is both effective and transferable.

Future work will extend this framework in several directions. First, we plan to handle richer document modalities by learning a latent or noisy structure. Second, we aim to incorporate multimodal elements such as tables or images. Finally, we will explore tighter integration with RAG pipelines, using structural signals not only for retrieval but also to guide evidence aggregation and answer generation in long-context LLMs.

6 Limitations

HQDR assumes access to reasonably well-structured documents with identifiable sections and paragraphs; its benefits are therefore most pronounced when a meaningful hierarchy can be ex-

tracted. This assumption holds in many practical domains—such as scientific articles, technical reports, manuals, and wiki-style content—where explicit headings and section layouts are readily available. For noisier web pages or unstructured text, HQDR would need to be paired with automatic structure induction modules (e.g., heading detection, layout-aware parsing, or learned hierarchical segmentation) to construct approximate document graphs. In the extreme case where no reliable structure can be inferred, HQDR reverts to a pure semantic dense retriever, without providing additional structural gains. In addition, while the universal token codebook aids cross-dataset transfer, our experiments are restricted to English scientific papers and Wikipedia; applying HQDR to other domains or languages may require domain-specific codebook reconstruction and learning new structural patterns.

7 Acknowledgments

This project was provided with computing AI and storage resources by GENCI at IDRIS thanks to the grant 2024-AD011014704R2 on the supercomputer Jean Zay's V100 and A100 partitions. We also thank Worldline for its support, which enabled us to leverage internal resources to conduct our experiments.

References

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137, Bangkok, Thailand. Association for Computational Linguistics.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Jianyuan Bo, Hao Wu, and Yuan Fang. 2025. Quantizing text-attributed graphs for semantic-structural integration. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pages 107–118, Toronto ON Canada. ACM.

Jan Buchmann, Max Eichler, Jan-Micha Bodensohn, Ilia Kuznetsov, and Iryna Gurevych. 2024. Document structure in long document transformers. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1056–1073, St. Julian's, Malta. Association for Computational Linguistics.

Ilias Chalkidis, Xiang Dai, Manos Fergadiotis, Prodromos Malakasiotis, and Desmond Elliott. 2022. An exploration of hierarchical attention transformers for efficient long document classification.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, Online. Association for Computational Linguistics.

Xinhao Huang, Zhibo Ren, Yipeng Yu, Ying Zhou, Zulong Chen, and Zeyi Wen. 2025. Seal: Structure and element aware learning improves long structured document retrieval. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 8537–8547, Suzhou, China. Association for Computational Linguistics.

Jiajie Jin, Xiaoxi Li, Guanting Dong, Yuyao Zhang, Yutao Zhu, Yongkang Wu, Zhonghua Li, Ye Qi, and Zhicheng Dou. 2025. Hierarchical document refinement for long-context retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3502–3520.

Feyza Duman Keles, Pruthuvi Mahesakya Wijewardena, and Chinmay Hegde. 2023. On the computational complexity of self-attention. In *International conference on algorithmic learning theory*, pages 597–619. PMLR.

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*, SIGIR '20, pages 39–48, New York, NY, USA. Association for Computing Machinery.

Thomas Kipf and Max Welling. 2016. Variational graph auto-encoders. In *NeurIPS Workshop on Bayesian Deep Learning*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

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.

Jindong Li, Yali Fu, Jiahong Liu, Linxiao Cao, Wei Ji, Menglin Yang, Irwin King, and Ming-Hsuan Yang. 2025. Discrete tokenization for multimodal llms: A comprehensive survey.

Xinze Li, Zhenghao Liu, Chenyan Xiong, Shi Yu, Yu Gu, Zhiyuan Liu, and Ge Yu. 2023. Structure-aware language model pretraining improves dense retrieval on structured data. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11560–11574, Toronto, Canada. Association for Computational Linguistics.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Antoine Louis, Gijs van Dijck, and Gerasimos Spanakis. 2023. Finding the law: Enhancing statutory article retrieval via graph neural networks. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2761–2776, Dubrovnik, Croatia. Association for Computational Linguistics.

Kenta Oono and Taiji Suzuki. 2021. Graph neural networks exponentially lose expressive power for node classification.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3980–3990, Hong Kong, China. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. 2024. Raptor: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*, volume 33, pages 16857–16867. Curran Associates, Inc.

Wenyu Tao, Xiaofen Xing, Yirong Chen, Linyi Huang, and Xiangmin Xu. 2025. Treerag: Unleashing the power of hierarchical storage for enhanced knowledge retrieval in long documents. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 356–371.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural discrete representation learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6309–6318.

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.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Cunxiang Wang, Zhikun Xu, Qipeng Guo, Xiangkun Hu, Xuefeng Bai, Zheng Zhang, and Yue Zhang. 2023. Exploiting abstract meaning representation for open-domain question answering. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2083–2096, Toronto, Canada. Association for Computational Linguistics.

Kexin Wang, Nils Reimers, and Iryna Gurevych. 2024. Dapr: A benchmark on document-aware passage retrieval. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4313–4330, Bangkok, Thailand. Association for Computational Linguistics.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7370–7377.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, volume 33, pages 17283–17297. Curran Associates, Inc.

Qingfei Zhao, Ruobing Wang, Yukuo Cen, Daren Zha, Shicheng Tan, Yuxiao Dong, and Jie Tang. 2024. Longrag: A dual-perspective retrieval-augmented generation paradigm for long-context question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22600–22632, Miami, Florida, USA. Association for Computational Linguistics.

A Implementation Details

A.1 Hardware Configuration

All experiments were conducted on a Linux server equipped with dual Intel Xeon Gold 6248 processors (20 cores each at 2.5 GHz). For training and

inference acceleration, we utilized an Nvidia Tesla V100 SXM2 GPU with 32 GB of VRAM.

A.2 Model Configuration

We employ the AdamW optimizer for both the self-supervised pre-training and the supervised finetuning stages.

Codebook Construction. The semantic codebook is initialized using the subword vocabulary from LLaMA-2. Following the protocol established in STAG, we cleaned the vocabulary by removing non-English and non-alphabetical entries and eliminating whitespace-based duplicates. Each token was embedded using the same Pre-trained Language Model (PLM) as our backbone (MPNet or BGE-base), resulting in a codebook size of $|\mathcal{E}| = 15,562$ and a dimensionality of $d = 768$.

Hyperparameters. The detailed hyperparameter settings for both stages are summarized in Table 6. The GNN utilizes a Graph Attention Network (GAT) with residual connections and layer normalization. For Top-k Softmax operation in Eq.(11), we set $k = 4$ based on empirical performance.

Hyperparameter	Pre-training	Fine-tuning
Backbone		
GNN Type	GAT (3 layers, 2 head)	
Hidden Dimension	256	
Optimizer	AdamW	
Dropout	0.1	
Codebook Temp (τ)	0.1	
Loss		
SCE Gamma (γ)	2.0	-
Commitment (β)	0.7	-
KL Divergence (λ)	0.8	-
τ_{doc}	0.1	-
Top-k Softmax (k)	-	4
τ_{ir}	-	0.2
Optimization		
Learning Rate	3×10^{-4}	5×10^{-5}
Weight Decay	0.01	0.001
Epochs	20	5

Table 6: Hyperparameter configuration for HQDR.

A.3 Dataset Filtering

While QASPER consists of naturally long scientific documents, the Natural Questions (NQ) dataset is derived from Wikipedia pages exhibiting high variance in length. To adapt NQ for our long-document QA task and ensure structural complexity, we applied a filtering pipeline. We selected the nq-dev split as our raw source to maintain a dataset size comparable to QASPER.

We cleaned and converted the HTML source of each NQ document into the QASPER format. We then filtered out documents with fewer than 6 sections to ensure the retained data contained sufficient structural information for training graph representations. The final processed dataset consists of 3,211 documents, each with its corresponding QA pair and evidence paragraphs.

A.4 Negative Sampling

The negative sampling strategy is tailored to the specific objectives of each training stage.

Pre-training. For the structural contrastive loss in Section.3.2.2, we select negative samples exclusively from the same document. Specifically, for a given parent-child pair, we select sibling nodes (nodes at the same hierarchical level) as negatives. Preliminary experiments indicated that including in-batch negatives from different documents slowed convergence and yielded no performance gains for the downstream retrieval task.

Finetuning. During the retrieval finetune stage, we construct the set of negative passages using two strategies to ensure hardness:

1. **Structural Negatives:** We include paragraphs that belong to the same section as the ground-truth paragraph.

2. **Semantic Hard Negatives:** We utilize the original frozen PLM to encode the query and all non-positive paragraphs. We select the top-k paragraphs with the highest cosine similarity to the query as hard negatives.

B Losses

Reconstruction loss $\mathcal{L}_{\text{Rec}}$. To make quantization genuinely useful for downstream retrieval, the discrete representations must not destroy the fine-grained semantics provided by the frozen PLM.

We reconstruct x_i from $z_{q,i}$ using a lightweight decoder $\text{Dec}(\cdot)$ and minimize a scaled cosine error (SCE):

$$\mathcal{L}_{\text{Rec}} = \frac{1}{|V|} \sum_{i \in V} \big(1 - \cos(x_i, z_{d,i})\big)^{\gamma}, \qquad (14)$$

where $z_{d,i} = \mathrm{Dec}(z_{q,i})$, $\gamma \geq 1$ controls the emphasis on harder examples.

The decoder is only used during pre-training: it provides a learning signal that encourages the encoder and the quantization layer to retain PLM-level meaning, but is discarded during IR fine-tuning and inference (Kipf and Welling, 2016).

Commitment loss $\mathcal{L}_{\mathrm{Commit}}$. To keep the fused representation $z_{f,i}$ compatible with its quantized counterpart $z_{q,i}$, we add a commitment loss:

$$\mathcal{L}_{\mathrm{Commit}} = \frac{1}{|V|} \sum_{i \in V} \beta\big(1 - \cos\big(z_{f,i}, \mathrm{sg}[z_{q,i}]\big)\big), \tag{15}$$

where $\mathrm{sg}[\cdot]$ is the stop-gradient operator and $\beta \in (0, 2]$ controls the strength of the constraint.

Semantic alignment loss $\mathcal{L}_{\mathrm{KL}}$. Finally, we align the structural fusion with the original semantic quantization. Let $p(z_{f,i})$ be the assignment distribution in Eq. (5), and $p(x_i)$ the distribution obtained by replacing $z_{f,i}$ with x_i. We minimize the KL divergence

$$\mathcal{L}_{\mathrm{KL}} = \frac{1}{|V|} \sum_{i \in V} D_{\mathrm{KL}}\big(p(x_i) \,\|\, p(z_{f,i})\big). \tag{16}$$

Intuitively, this constrains the fused representations to be quantized into LLM tokens similar to the original semantic features, preventing structural information from distorting core meaning while allowing beneficial adjustments.

C Qualitative Analysis

C.1 Embedding Visualization

To validate our fusion mechanism, Figure 3 visualizes T-SNE projections of node representations from a QASPER document at three distinct stages: initial semantic embeddings (X), GNN-derived structural embeddings (Z_e), and final fused representations (Z_f).

Initial PLM embeddings (X, left) cluster primarily by semantic overlap, leaving section-paragraph relationships implicit. Conversely, GNN embeddings (Z_e, middle) capture the explicit hierarchical topology but diverge significantly from the semantic space. The final fused representations (Z_f, right) demonstrate the efficacy of our approach: they retain the global semantic layout of X that is essential for dense retrieval, while exhibiting a visibly tighter alignment between section nodes and their constituent paragraphs. This confirms that HQDR successfully injects explicit hierarchical signals without distorting fine-grained semantics, resulting in a geometry that optimally supports our hybrid retrieval strategy.

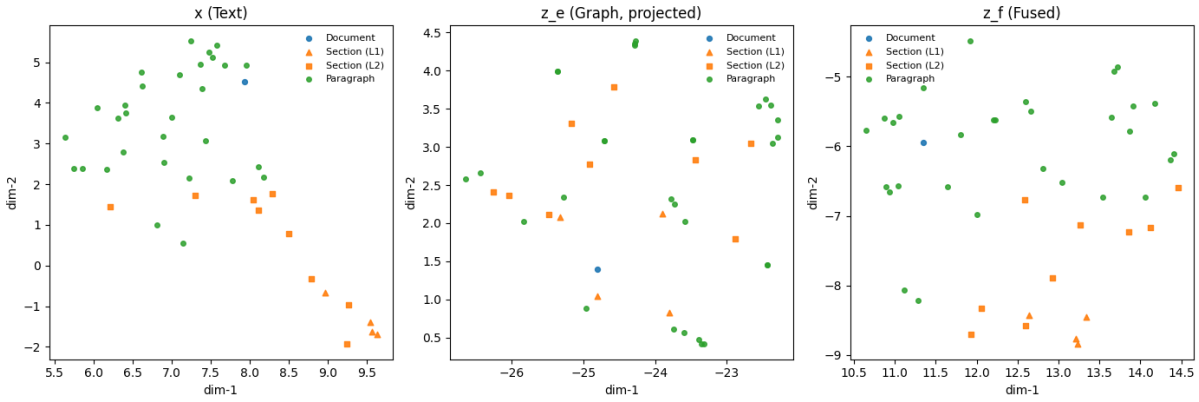

Figure 3: T-SNE visualization of node embeddings for a sampled document. **Left:** Initial PLM text embeddings ($\mathbf{x}$) showing semantic distribution. **Middle:** Projected GNN structural embeddings ($\mathbf{z}_e$) revealing topological clusters. **Right:** Fused embeddings ($\mathbf{z}_f$) demonstrating the integration of structural awareness into the semantic space.