HiGoE: Hierarchical Graph of Evidence to Enhance Retrieval-Augmented Generation for Long-context Summarization

Long Yuan[1],Kaiwen Tian[2], Zi Chen[1]*, Bolong Zheng[1]*,Chuan Ma[3]
[1]Wuhan University of Technology, [2]Nanjing University of Science and Technolgy, [3]Chongqing University
{longyuan,zichen,bolongzheng}@whut.edu.cn, tiankaiwen@njust.edu.cn, chuan.ma@cqu.edu.cn

Abstract

Long-context summarization is pivotal for extracting core insights from extensive documents. While Large Language Models (LLMs) show remarkable capabilities, they frequently encounter attention dilution and hallucination with lengthy inputs. Retrieval-Augmented Generation (RAG) partially mitigates this, but conventional RAG relies on shallow similarity retrieval of fragmented chunks, failing to capture high-level thematic structures and long-range dependencies. Although graph-based RAG approaches have emerged to address these structural limitations, existing solutions, such as Graph of Records (GoR), critically suffer from a fundamental flaw: they paradoxically re-introduce hallucinations by constructing graphs based on unreliable, LLM-generated responses. To overcome these challenges, we introduce *Hierarchical Graph of Evidence* (**HiGoE**) (Code link https://github.com/tkw123/HiGOE). HiGoE redefines the retrieval process by replacing unreliable chunk-based methods with a filtered proposition–evidence graph, ensuring verifiable fact grounding and substantially reducing hallucination. Moreover, HiGoE leverages Personalized PageRank (PPR) to cluster related nodes into thematic hierarchies, thereby restoring global document structure and effectively mitigating attention dilution. To model complex, multi-level relations beyond mere shallow similarity, we develop an Enhanced Graph Attention Network. Experiments show HiGoE consistently surpasses baselines in quality and efficiency.

1 Introduction

In an era of overwhelming textual information, long-context summarization has evolved from simple text compression into a versatile interface for knowledge navigation. This evolution is driven by its indispensable role in diverse downstream applications, such as extracting actionable decisions from multi-hour meeting transcripts (Kirstein et al., 2024), distilling legislative nuances from voluminous government reports (Gesnouin et al., 2024), and tracking narrative arcs in literary works (Chang et al., 2024). Crucially, a robust long-context summarization needs to efficiently answer diverse user queries, covering everything from global thematic overviews to specific factual details, all without repeated, computationally expensive full-text processing. For example, a long literary work may be queried multiple times to generate focused summaries of its plot development, character relationships, or thematic evolution across different chapters. Although Large Language Models (LLMs) have demonstrated substantial progress (Grattafiori et al., 2024; Yang et al., 2025; Guo et al., 2025) in long-context understanding (Li et al., 2024b; Edge et al., 2024), they inherently struggle with long-context summarization. Specifically, when processing lengthy inputs, LLMs often exhibit attention dilution, leading them to overlook crucial details or misprioritize salient content (Chen et al., 2024; Ding et al., 2024; Zhao et al., 2025). Moreover, their generative nature makes them susceptible to hallucination (Ji et al., 2023). While Retrieval-Augmented Generation (RAG) mitigates hallucination risks via external corpus retrieval (Yue et al., 2025; Jin et al., 2025), standard RAG approaches typically rely on myopic filters that retrieve fragmented chunks based solely on surface similarity, thereby inadvertently shattering the global narrative structure of the document (Karpukhin et al., 2020; Izacard et al., 2022).

These limitations have motivated graph-based extensions to RAG, aiming to model intricate dependencies across scattered information units. However, existing graph-based methods remain constrained. Prior graph-based or neural techniques (Li et al., 2024a; Wu et al., 2021; Huang et al., 2021; Kryscinski et al., 2022) often capture only shallow structural dependencies, leav-

*Corresponding Author

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 19703–19724

ing deeper logical and thematic relations underexplored. More recently, Graph of Records (GoR) (Zhang et al., 2025a) organizes retrieved information into graphs by generating hypothetical query-response pairs for each chunk. While GoR attempts to bridge semantic gaps, its reliance on unverified, model-generated query-response paradigms is structurally flawed: building the retrieval index on potentially hallucinated LLM outputs directly injects inaccuracies into the core retrieval backbone. Furthermore, the flat graph structure employed by such methods inherently lacks hierarchical abstraction, severely limiting reasoning over high-level themes and global contexts.

To address these limitations, we argue that robust summarization demands a knowledge structure that is both factually verifiable and hierarchically organized. Motivated by this principle, we propose a novel framework: the *Hierarchical Graph of Evidence* (**HiGoE**). HiGoE introduces three paradigm shifts: (1) From Response to Verifiable Evidence: HiGoE constructs its graph using a two-stage filtered process to extract atomic, verifiable propositions, rigorously grounding every node and edge in solid, factual evidence. This eliminates the "verification gap" and prevents hallucination propagation into the retrieval mechanism. (2) From Flat to Hierarchical Structure: To restore global structural understanding, HiGoE leverages Personalized PageRank (PPR) (Brin and Page, 1998) to detect thematic communities, organizing related nodes into meaningful hierarchies. This allows the retriever to "zoom out" for global overviews and "zoom in" for specific details, mitigating attention dilution and enhancing contextual awareness. (3) From Shallow Matching to Structural Alignment: To capture deep reasoning dependencies, we design an Enhanced Graph Attention Network (GAT) (Veličković et al., 2018), trained with a novel joint contrastive-ranking objective. This GAT aligns structural evidence with summary queries, enabling HiGoE to model complex relationships and extract nuanced insights from the document's underlying structure. To validate these architectural shifts, we conduct comprehensive experiments on five long-context benchmarks. Our main contributions are summarized as follows:

- We propose HiGoE, a two-stage evidence graph construction framework that ensures precise evidence organization by replacing hallucination-prone responses with verifiable facts.
- We introduce a hierarchical PPR-enhanced graph and an Enhanced GAT encoder with multi-scale aggregation to integrate local evidence with global semantics, optimized under a joint contrastive-ranking objective.
- We conduct extensive experiments on five long-context summarization benchmarks, consistently outperforming the state-of-the-art GoR and other strong methods in both summarization quality and inference efficiency.

2 Preliminaries

Retrieval-Augmented Generation. Retrieval-Augmented Generation (RAG) (Ram et al., 2023) enhances large language models by conditioning generation on retrieved passages. Given a query q, a retriever selects the top-k passages $\{\zeta_i\}_{i=1}^{k}$ from a corpus $\mathbb{C}$ based on relevance scores $p(\zeta \mid q)$, forming the retrieved set $\mathbb{C}_k(q)$. The generator then produces output y based on both q and each retrieved passage, leading to the overall generation probability, which is defined as:

$$p(y \mid q) = \sum_{\zeta \in \mathbb{C}_k(q)} p(y \mid q, \zeta)\, p(\zeta \mid q)$$

Problem Formulation. Given a long document $\mathcal{D}$, which typically exceeds the context window limit of LLMs, our goal is to generate a concise and coherent summary y. We formulate this task as a retrieve-then-generate process. First, the document $\mathcal{D}$ is segmented into a sequence of chunks $\mathcal{C} = \{c_1, c_2, \ldots, c_N\}$. Instead of feeding the entire sequence $\mathcal{C}$ directly into the LLM, we construct a structured index in the form of a graph $\mathcal{G}$ to represent the evidential relations within $\mathcal{D}$. Given a summary-oriented query q, the system retrieves a subset of relevant evidence $\mathcal{Z}(\mathcal{G}, q) \subset \mathcal{G}$ and generates y conditioned on both q and $\mathcal{Z}$. The optimal summary $\hat{y}$ is obtained by maximizing the following probability as:

$$\hat{y} = \arg\max_{y}\ p(y \mid q, \mathcal{Z}(\mathcal{G}, q))$$

3 Proposed Method: HiGoE

As shown in Figure 1, HiGoE transforms unstructured documents into a verifiable, hierarchical structure to enhance long-context summarization. It operates in four stages: (1) Graph construction, which grounds retrieval in atomic proposition-evidence pairs to minimize hallucinations; (2) Hierarchical enhancement, utilizing Personalized PageRank (PPR) to recover global thematic structures; (3)

Training preparation and loss calculation, which derives fine-grained self-supervised signals for learning task-aligned representations; and (4) Node representation learning, where an Enhanced GAT aligns structural evidence with summary queries via a joint contrastive-ranking objective.

3.1 Graph Construction

Existing state-of-the-art methods like GoR construct graphs by linking retrieved chunks to LLM-generated hypothetical responses. While this creates semantic bridges, it fundamentally treats unverified model outputs as ground truth, leading to error propagation. Unlike GoR's query-response pairs which are prone to hallucination accumulation, our proposition-evidence paradigm enforces a strict verification bottleneck. By extracting atomic facts and filtering them via a two-stage process, we ensure that every edge in our graph represents a reliable semantic link.

Proposition Extraction and Evidence Retrieval. Each document is chunked into semantically coherent segments, from which an LLM extracts propositions that can be supported or contradicted by the text. A dense retriever (e.g., Contriever (Izacard et al., 2022)) retrieves evidence passages for each proposition, and the resulting proposition–evidence pairs are linked to construct the initial graph structure. This proposition-centered process grounds the graph in verifiable facts and explicitly encodes evidential relations, yielding a more reliable semantic structure than query–response based graph construction.

Two-Stage Filtering. To ensure the reliability of propositions at scale and mitigate the risk of hallucination inherent to LLMs, we apply a two-stage filtering strategy. Unlike previous approaches (e.g., GoR) that prompt LLMs to generate heuristic query-response pairs from scratch, which is a process highly susceptible to hallucination, our method focuses on extracting verifiable atomic facts directly grounded in the source text. A rule-based stage first removes trivial, redundant, or malformed propositions. Subsequently, an LLM-as-a-Judge stage assesses factual consistency, relevance, and clarity, retaining only propositions with high confidence. Crucially, in this pipeline, the LLM operates strictly as an evaluator rather than an unconstrained generator. Recent literature confirms that the LLM-as-a-Judge paradigm demonstrates high alignment with human judgment and excellent reliability in verification tasks (Zheng et al., 2023; Li et al., 2025a). Ultimately, this rigorous filtering ensures that the graph contains concise, trustworthy evidence units with strong semantic grounding, forming a robust foundation for downstream retrieval and reasoning.

3.2 Hierarchical Enhancement via PPR

While the proposition–evidence graph provides local factual precision, it remains structurally flat and cannot capture high-level themes. A flat graph traps the retriever in local neighborhoods. To enable global reasoning, we leverage Personalized PageRank (PPR) to diffuse node influence, identifying densely connected thematic communities. This effectively constructs a semantic scaffolding, allowing the model to retrieve not just facts, but the context in which they reside. For a source node s, the PPR vector $\mathbf{r}_s \in \mathbb{R}^{|V|}$ is defined as:

$$\mathbf{r}_s = \alpha\, \mathbf{e}_s + (1-\alpha)\, P\, \mathbf{r}_s$$

where $\alpha \in (0,1)$ is the teleport probability, $\mathbf{e}_s$ is the one-hot vector of s, and $P \in \mathbb{R}^{|V|\times|V|}$ is a column-stochastic transition matrix with P_{ij} denoting the probability of moving from j to i. Intuitively, the process either restarts at s with probability α or follows the graph transitions, yielding a stationary distribution that captures both local proximity and long-range influence. Based on PPR scores, we rank nodes by their row-sum influence, select the top ones as seeds, and assign each remaining node to the seed exerting the strongest influence. We then aggregate the content of each community and use an LLM to produce a summary node representing its central theme. This step converts the flat evidence graph into a hierarchical structure that exposes both fine-grained details and high-level semantic organization.

3.3 Training Preparation & Loss Calculation

The hierarchical graph alone cannot support effective retrieval unless node representations reflect their semantic relevance to summary queries. This necessitates a tailored training procedure and objective capable of shaping node representations to reflect their functional roles within the graph.

Self-Supervised Training. A core challenge in the training procedure is the lack of an effective supervisory signal. Manually annotating fine-grained labels for long documents is costly and impractical at scale, and relying solely on document-level references such as introductions or abstracts provides

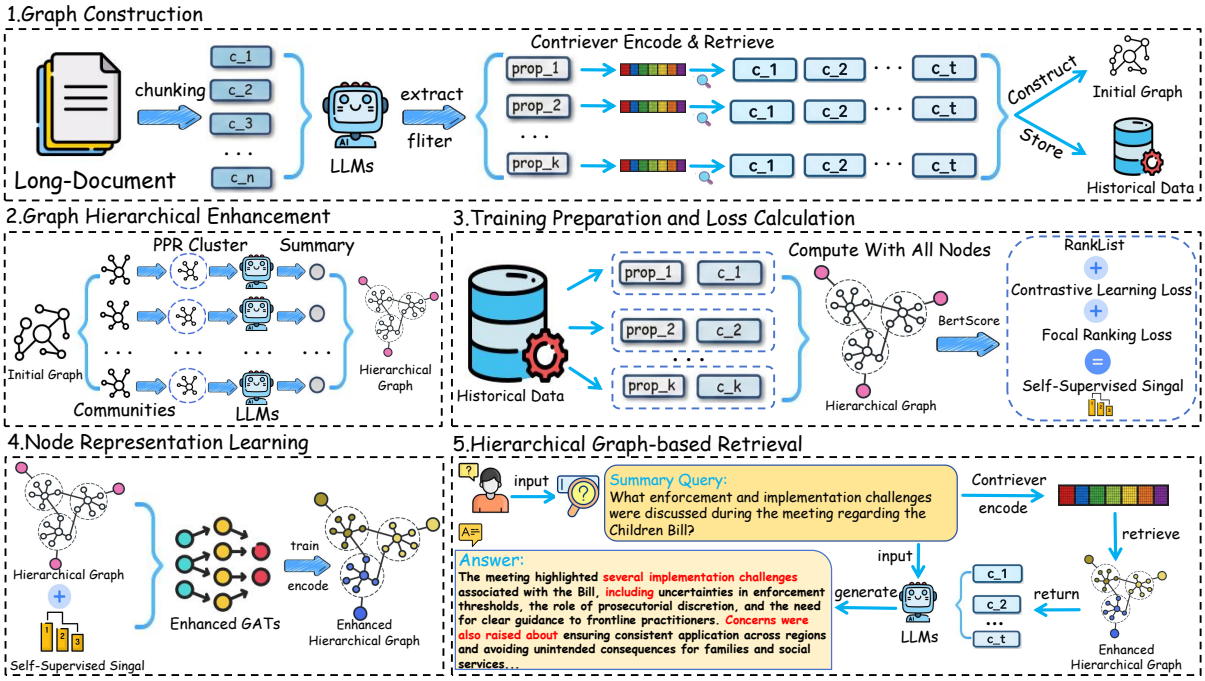


Figure 1: The overall framework of **Hierarchical Graph of Evidence (HiGoE)**. A graph-based framework to enhance RAG for long-context summary, integrating graph construction, hierarchical enhancement via PPR, training preparation & loss calculation, node representation learning and hierarchical graph-based retrieval.

only coarse signals that fail to capture important details dispersed throughout the text. We address the supervision bottleneck through a fractal-like self-alignment strategy. Since a global summary is too coarse to supervise local node selection, we treat each proposition as a micro-summary of its source chunk. This provides rigorous, fine-grained supervision signals without human annotation. Each document is segmented into chunks and encoded using Contriever. From every chunk c_i, the LLM generates a proposition $\mathbf{q}_i$, yielding k propositions per document that pass the filtering criteria. We compute BERTScore (Zhang et al., 2019) between c_i and each node in hierarchical graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to produce a similarity ranking as:

$$\mathcal{R}_i = [\, v^+_{i(1)}, v^-_{i(2)}, \ldots, v^-_{i(|\mathcal{V}|)} \,]$$

where $v^+_{i(1)}$ is the positive node for $\mathbf{q}_i$, and the remaining nodes are negatives. This ranking serves as node-level supervision without human annotation. In practice, each proposition serves as a summary-oriented query, and the model must learn node representations whose similarities respect the ranking. A single objective cannot exploit these signals effectively. Contrastive loss distinguishes positives from negatives but ignores their relative ordering, while ranking loss preserves ordering but lacks margin constraints. We therefore combine both signals in a joint contrastive–ranking objective that learns proposition–evidence alignment in a fully self-supervised manner.

Contrastive Learning Loss. To strictly distinguish matching evidence from distractors, we employ an InfoNCE-style objective (Oord et al., 2018). For each proposition $\mathbf{q}_i$ within a training batch, we define a raw similarity score with any node $\mathbf{v}$ as their dot product, scaled by a learnable temperature τ like:

$$s(\mathbf{q}_i, \mathbf{v}) = \exp\left(\frac{\mathrm{sim}(\mathbf{q}_i, \mathbf{v})}{\tau}\right)$$

Let $\mathcal{T}$ be the set of all propositions across all graphs within a single training batch. The contrastive loss over the entire batch is:

$$\mathcal{L}_{\mathrm{cl}} = -\frac{1}{|\mathcal{T}|} \sum_{\mathbf{q}_i \in \mathcal{T}} \log \frac{s(\mathbf{q}_i, v_i^+)}{s(\mathbf{q}_i, v_i^+) + \sum_{\mathbf{n}^- \in \mathcal{N}_i^-} s(\mathbf{q}_i, \mathbf{n}^-)}$$

where v_i^+ is the positive node for proposition $\mathbf{q}_i$, and $\mathcal{N}_i^-$ is the set of its corresponding negative nodes. The objective is to maximize the similarity between a summary query and its positive sample while minimizing similarity to all negatives.

To further enhance fine-grained discrimination, we adopt a hard-negative mining strategy. Specifically, we introduce a margin-based triplet loss that enforces the similarity to the hardest negative sample $\mathbf{n}^-_{\mathrm{hard}}$ to be at least γ lower than that of the

positive sample:

$$\mathcal{L}_{\text{hard}} = \frac{1}{|\mathcal{T}|} \sum_{\mathbf{q}_i \in \mathcal{T}} \text{ReLU}\big(\text{sim}(\mathbf{q}_i, \mathbf{n}^-_{\text{hard}}) - \text{sim}(\mathbf{q}_i, v_i^+) + \gamma\big)$$

The final loss is a weighted sum of the contrastive and hard-negative penalty terms, $\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{cl}} + \lambda\mathcal{L}_{\text{hard}}$, where γ is the margin and λ is a weighting coefficient. This combined objective forces the positive sample's similarity to surpass that of even the most challenging distractors.

Focal Ranking Loss. Contrastive loss distinguishes positives from negatives but ignores the relative ordering among negatives. To address this, we adopt a pair-wise focal ranking loss (Cao et al., 2007). For each pair (i, j) with $y_i > y_j$ in BERTScore ranking, the model is encouraged to assign a higher similarity score s_i than s_j. The loss is defined as:

$$\mathcal{L}_{\text{rank}} = \frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \alpha\big(1 - \sigma(s_i - s_j)\big)^\beta \cdot \log\Big(1 + e^{-(s_i - s_j)}\Big)$$

where $\mathcal{P} = (i, j) : y_i > y_j$ is the set of valid ranking pairs. The focal weight $(1 - \sigma(s_i - s_j))^\beta$ down-weights easy pairs and emphasizes harder ones, enforcing consistency between predicted similarities and target rankings. The overall training objective combines the contrastive loss (including the hard-negative term) and the ranking loss:

$$\mathcal{L}_{\text{total}} = (\mathcal{L}_{\text{cl}} + \lambda\mathcal{L}_{\text{hard}}) + w_{\text{rank}}(e) \cdot \mathcal{L}_{\text{rank}}$$

where $w_{\text{rank}}(e)$ is an epoch-dependent weight that gradually increases the contribution of the ranking loss during training. More details about training can be seen in A.4.

3.4 Node Representation Learning

Currently, while Contriever-initialized nodes offer basic semantics, effective retrieval requires representations that also encode structural dependencies within the hierarchical graph. Classical embedding methods such as random-walk embeddings (Grover and Leskovec, 2016) and label propagation (Kang et al., 2006) are non-differentiable and unsuitable for end-to-end learning. Although Graph Attention Networks (GATs) (Veličković et al., 2018) support differentiable graph encoding, the vanilla GAT used in GoR is not ideal for long-context summarization due to its locality-biased receptive field, which limits cross-segment reasoning, and over-smoothing in deeper layers, which leads to indistinguishable node representations (Wu et al., 2020).

To enable structural reasoning across local evidence and global discourse, we develop an Enhanced GAT incorporating hierarchical and relational signals. Initialized with Contriever embeddings, it refines node representations via multi-head attention, residual connections, layer normalization, and Jumping Knowledge (JK) aggregation (Xu et al., 2018). At layer l, node i is updated as:

$$\tilde{\mathbf{h}}_i^{(l+1)} = \text{Concat}_{k=1}^{K}\, \sigma\left(\sum_{j\in\mathcal{N}(i)} \alpha_{ij}^k \mathbf{W}^k \mathbf{h}_j^{(l)}\right),$$

$$\mathbf{h}_i^{(l+1)} = \text{LayerNorm}\Big(\tilde{\mathbf{h}}_i^{(l+1)} + \mathbf{W}_{\text{proj}}\mathbf{h}_i^{(l)}\Big).$$

where K is the number of attention heads, $\mathcal{N}(i)$ is the neighbor set of node i, α_{ij}^k the attention weight, $\mathbf{W}^k$ the projection matrix, and σ a nonlinear activation. After L layers, we employ JK pooling to aggregate multi-scale features: $\mathbf{h}_{\text{final}} = \max(\mathbf{h}^{(0)}, \mathbf{h}^{(1)}, \ldots, \mathbf{h}^{(L)})$. This design adaptively fuses representations across layers, enabling each node to capture the most informative features.

3.5 Hierarchical Graph-based Retrieval

During inference, HiGoE retrieves evidence from the hierarchical proposition–evidence graph, where each node has a learned embedding that encodes both fine-grained facts and community-level semantics. Crucially, we employ a Unified Vector Space approach: both the summary nodes (thematic summaries generated during hierarchical enhancement) and the normal nodes (original text propositions) are indexed together in the same dense vector space. The graph is maintained in two synchronized forms for text storage (NetworkX) and computation (DGL), allowing embeddings to be aligned with their original textual content. Given a summary query, we encode it with Contriever and compute cosine similarity against all node embeddings. This unified design allows the system to naturally adapt to different query intents: global queries (e.g., asking for an overarching theme) typically yield higher similarity scores for summary nodes, while local, specific queries favor normal nodes containing exact facts.

To prevent redundant retrieval from a single cluster, we apply a Cross-Community Top-k Filtering strategy. By checking the PPR community tags dur-

ing retrieval, we penalize over-represented communities, forcing the system to extract evidence across different thematic clusters for better global coverage. After the selected nodes are mapped back to their textual forms, they are aggregated to construct the context. When a summary node is retrieved, its pre-computed text is fetched and prepended to the generator's context, effectively compressing massive information from multiple source chunks into a single manageable paragraph. Finally, we concatenate these texts with the query and feed them into the LLM. This community-aware retrieval captures both local and global relevance, yielding more coherent and comprehensive summaries than standard dense retrievers.

4 Experiments

Datasets and Evaluation. We evaluate HiGoE on five diverse long-context benchmarks: QMSum (Zhong et al., 2021), WCEP (Gholipour Ghalandari et al., 2020), BookSum (Kryscinski et al., 2022), GovReport (Huang et al., 2021), and SQuALITY (Wang et al., 2022), spanning meetings, news, narratives, and government reports. Following the setup of GoR (Zhang et al., 2025a), we report ROUGE-1, ROUGE-2, and ROUGE-L scores (Lin, 2004). Detailed statistics are provided in Appendix A.1 and A.2.

Implementation Details. Documents are split into chunks using LangChain. GPT-4 generates verifiable propositions, while Contriever retrieves the top-6 supporting chunks to construct the proposition–evidence graph. Hierarchical enhancement applies PPR for community detection, with GPT-4 generating community summaries. Crucially, we treat the pre-trained Contriever strictly as a frozen feature extractor without fine-tuning its internal Transformer layers. All nodes are initialized with its static 768-dimensional embeddings and subsequently refined via a specialized 2-layer Enhanced GAT. To prevent oversmoothing and stabilize gradients, this trainable Graph Encoder incorporates residual connections and layer normalization, alongside a Jumping Knowledge strategy using max-pooling aggregation to adaptively fuse local and global structural features. During training, only the GAT projection matrices and the temperature parameter τ are fully tuned using the Adam optimizer with a Cosine Annealing scheduler (learning rate of 1e-3). Queries are then encoded with Contriever and combined with retrieved chunks for generation using LLMs. Fixed random seeds ensure reproducibility, with further details in Appendix A.5.

Baselines. For a comprehensive evaluation, we compare HiGoE with representative baselines: (1) Random-walk embeddings: Node2Vec (Grover and Leskovec, 2016). (2) Sparse retrievers: BM25 (Robertson et al., 2009) and TF-IDF (Ramos et al., 2003). (3) Dense retrievers: Contriever (Izacard et al., 2022), DPR (Karpukhin et al., 2020), Dragon (Lin et al., 2023), Sentence-BERT (Reimers and Gurevych, 2019). (4) Hybrid retrievers: BM25 + DPR with reciprocal rank fusion. (5) Long-context LLMs: Gemma-8K (Team et al., 2024), Mistral-8K (Jiang et al., 2023). (6) Full-context input: A retrieval-free baseline feeding the entire document into an LLM. It uses two settings: GoR's original all-MiniLM-L6-v2 (randomly sampling spans for long documents) and GPT-4-Turbo, a strong upper bound supporting up to 128K tokens. (7) Thought retriever: Thought-R (Feng et al., 2024), retrieving intermediate reasoning steps. (8) Graph of Records (GoR): original GoR (Zhang et al., 2025a) linking retrieved chunks and LLM outputs via a GNN with contrastive and pairwise ranking objectives, with further details in Appendix A.3.

4.1 Main Results

We report our main results of experiments on five datasets in Table 1.

Superiority over retriever-based methods. Compared with sparse retrievers, dense retrievers, and hybrid retrievers, HiGoE achieves clear improvements. This advantage comes from the hierarchical proposition–evidence graph, which encodes multi-level semantic dependencies and supports retrieval beyond shallow similarity matching.

Advantage over long-context LLMs. HiGoE's superiority over long-context LLMs (e.g., Gemma-8K and Mistral-8K) highlights a critical insight: Structured retrieval beats brute-force context scaling. Even with 8K context, standard LLMs suffer from attention dilution, whereas HiGoE's graph precisely routes attention to salient evidence.

Effectiveness against full-context. To rigorously evaluate against brute-force context scaling, we compare HiGoE with two full-context baselines: GoR's original setup based on all-MiniLM-L6-v2, which handles long documents by randomly sampling text spans, and a stronger upper-bound baseline using GPT-4-Turbo (128K) that processes the entire document directly without retrieval. No-

Model	QMSum			WCEP			BookSum			GovReport			SQuALITY		
	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2
Node2Vec	18.5	31.8	6.3	13.9	20.1	6.3	13.6	27.4	4.6	18.1	36.7	12.4	17.0	32.9	7.7
BM25	18.4	32.1	6.1	15.5	22.6	7.3	13.7	26.7	4.9	18.2	39.2	13.0	17.0	31.4	8.1
TF-IDF	18.3	31.2	6.3	15.3	22.3	7.3	13.6	26.6	4.9	18.1	39.2	12.8	17.0	31.4	8.1
Contriever	19.1	32.7	7.7	15.7	23.5	7.7	14.4	29.8	5.5	20.2	39.8	17.6	16.8	32.6	8.3
DPR	18.6	32.1	6.7	15.6	22.5	7.5	13.8	27.1	4.8	19.1	39.4	15.5	17.4	33.1	8.4
Dragon	19.2	33.5	7.7	14.6	21.8	6.8	13.7	27.2	4.8	19.6	38.2	16.0	16.2	29.6	7.5
SBERT	19.0	33.0	7.4	13.7	20.5	5.5	14.4	29.5	5.4	20.0	39.8	15.8	17.1	32.1	7.8
BM25+DPR	18.3	31.8	6.6	15.7	22.1	7.6	14.1	28.9	5.4	19.4	37.4	15.0	16.6	31.5	7.4
Gemma-8K	19.8	33.5	7.3	15.6	21.9	7.7	12.8	23.4	4.2	17.4	33.8	11.4	12.9	19.7	5.8
Mistral-8K	19.6	31.2	7.2	16.7	24.2	8.8	13.5	26.2	5.3	16.0	28.9	9.4	16.9	32.2	8.1
Full Context(all-MiniLM-L6-v2)	19.4	33.1	6.8	14.4	21.0	7.1	14.4	28.9	5.9	18.4	39.1	13.8	17.8	34.0	8.8
Full Context(GPT-4-Turbo)	21.4	34.1	**9.2**	**22.9**	**34.4**	10.3	14.3	33.0	6.1	21.8	47.1	17.6	16.3	38.3	**10.6**
Thought-R	19.0	33.9	7.6	15.2	22.4	7.4	14.2	29.5	5.7	20.4	40.3	17.0	17.3	32.0	8.0
GoR(Original)	19.8	34.5	7.8	18.1	25.4	9.2	14.9	31.5	6.6	20.9	41.4	16.8	17.8	34.0	8.5
GoR(GPT-4+LLaMA-2-7b-chat)	18.9	35.7	8.3	18.5	28.1	8.5	14.9	35.3	6.3	20.4	50.1	17.1	17.7	38.9	8.4
GoR(GPT-4+GPT-4)	20.6	36.5	7.4	21.5	31.2	9.7	15.5	36.2	6.6	21.5	52.3	18.5	17.5	39.3	8.7
HiGOE(GPT-4+LLaMA-2-7b-chat)	21.1	37.0	8.2	20.5	30.6	9.7	16.0	**37.4**	7.1	23.7	**53.5**	**19.8**	**18.8**	41.8	9.7
HiGOE(GPT-4+GPT-4)	**21.7**	**37.7**	8.5	22.7	32.8	**11.1**	**16.2**	37.2	**7.3**	**23.9**	52.6	19.3	18.6	**41.9**	9.9

Table 1: **Experimental results on QMSum, WCEP, BookSum, GovReport and SQuALITY datasets over long-context global summarization tasks w.r.t. Rouge-L (R-L), Rouge-1 (R-1), and Rouge-2 (R-2)**. GoR (Original) denotes the original GoR's setting of Mixtral-8x7B-Instruct-v0.1+LLaMA-2-7b-chat.

tably, HiGoE outperforms both baselines in most cases. This demonstrates that even when an LLM is natively capable of ingesting massive inputs, our structured graph-based retrieval remains essential for accurate information synthesis in high-density or narrative-heavy documents by actively mitigating attention dilution.

Comparison with GoR. GoR constructs graphs from potentially hallucinated query-response pairs, while HiGoE builds on verifiable propositions and enhances them with PPR-based hierarchical communities. This shift from generative to evidential graph construction, coupled with our contrastive-ranking objective, leads to superior summarization quality and robustness across all datasets.

Other observations. Node2Vec performs poorly due to the lack of semantic supervision, Full Context suffers from input truncation, and Thought Retriever falls short without hierarchical summaries or graph-enhanced representations.

4.2 Ablation Study

We conduct ablation experiments to assess the contribution of each module in HiGoE, with results summarized in Table 2.

Necessity of graph-aware representation learning. Removing training (w/o train) leads to clear degradation compared with HiGoE, showing that learned graph representations capture richer semantics than raw retriever embeddings.

Synergy of joint optimization. Both contrastive loss ($\mathcal{L}$cl) and ranking loss ($\mathcal{L}$rank) are indispensable. Dropping either term results in consistent performance drops across datasets, indicating their complementarity.

Superiority of fine-grained self-supervision. Switching to supervised training with global summaries harms performance, confirming that the indirect self-supervised signal better aligns with retrieval-based summarization.

Impact of hierarchy and verification. Removing PPR-based clustering (w/o ppr) or the two-stage filtering (w/o filter) reduces performance, highlighting the importance of hierarchical structure and high-quality proposition–evidence construction.

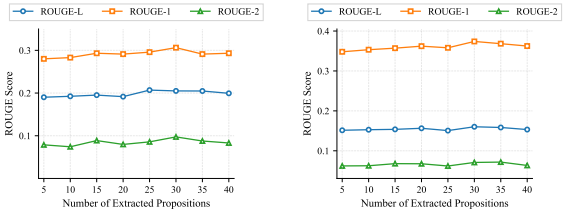
Overall, these results verify that each component contributes meaningfully to the final performance.

4.3 Discussions

Effect of the number of extracted propositions. We investigate how varying the number of extracted propositions per document affects performance. As shown in Figure 2, ROUGE scores rise with minor fluctuations before reaching 30 propositions. However, a downward trend emerges as the number of propositions exceeds 30. This decline is driven by redundancy and, more importantly, the limited information in local chunks, which cannot sustain a large set of diverse, high-quality propositions. As a result, low-quality propositions are introduced

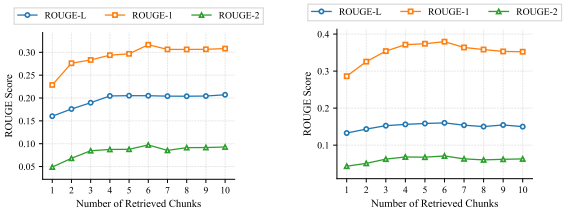
Variant	QMSum			WCEP			BookSum			GovReport			SQuALITY		
	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2
w/o train (*)	19.9	34.6	7.2	18.1	27.0	6.8	14.3	31.3	4.9	22.5	52.1	17.9	18.0	37.8	7.6
w/o $\mathcal{L}_{\text{cl}}$	20.6	36.4	7.7	17.8	26.6	6.6	15.6	**37.6**	6.9	23.4	53.0	19.0	18.6	41.3	9.5
w/o $\mathcal{L}_{\text{rank}}$	20.0	35.9	8.0	**20.7**	30.1	9.3	15.6	37.3	6.6	22.7	52.0	18.4	18.4	40.5	9.2
w/ sup (*)	17.2	32.0	5.2	17.7	27.3	6.5	13.5	28.9	4.0	19.1	47.1	12.8	18.0	38.4	7.4
w/o ppr	20.1	36.0	7.7	20.0	29.0	8.0	15.3	35.7	6.4	22.4	52.8	18.4	18.7	**41.8**	9.5
w/o filtering	20.2	35.6	7.8	20.2	29.3	8.4	15.4	36.3	6.5	23.4	53.4	19.3	18.6	41.3	9.6
HiGOE(GPT-4+LLaMA-2-7b-chat)	**21.1**	**37.0**	**8.2**	20.5	**30.6**	**9.7**	**16.0**	37.4	**7.1**	**23.7**	**53.5**	**19.8**	**18.8**	**41.8**	**9.7**

Table 2: **Ablation study results on QMSum, WCEP, BookSum, GovReport and SQuALITY datasets w.r.t. R-L, R-1 and R-2**. **(*)** indicates significant performance degradation compared to the full model.



(a) WCEP (b) BookSum

Figure 2: **Effect of the number of extracted propositions w.r.t. R-L, R-1 and R-2**.



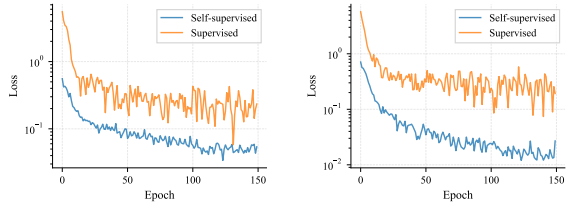
(a) WCEP (b) BookSum

Figure 3: **Effect of the number of retrieved chunks w.r.t. R-L, R-1 and R-2**.

as noise, ultimately hindering generation quality. Aggregated results across five datasets show that 30 propositions consistently rank either first or second, and additional results for other datasets are provided in Appendix A.6. Hence, we set 30 as the default choice in HiGoE.

Effect of the number of retrieved chunks. We further analyze the impact of varying the number of retrieved chunks from 1 to 10. As shown in Figure 3, performance across ROUGE-1, ROUGE-2, and ROUGE-L generally follows a rise-then-fall trend. Both WCEP and BookSum exhibit this pattern, with performance peaking at around 6 chunks before dropping slightly when more are added. Considering all five datasets, retrieving 6 chunks consistently ranks first or second. Results for other datasets are provided in Appendix A.7. Therefore, we adopt 6 chunks as the default configuration in HiGoE.

Supervised training vs self-supervised training. To compare supervised and self-supervised training, we run experiments using global summaries



(a) WCEP (b) BookSum

Figure 4: **Differences between self-supervised and supervised training w.r.t loss**.

as supervision. Each document provides only one global summary, whereas the self-supervised method generates 30 local propositions per sample. To match corpus size, we replicate global summaries and train with a standard negative log-likelihood loss. Figure 4 shows training losses on WCEP and BookSum. Self-supervised training yields lower and more stable losses because global summaries offer coarse, poorly aligned supervision for individual nodes, introducing optimization noise. In contrast, self-supervised labels derive from fine-grained local evidence and provide more precise learning signals. Additional details are provided in Appendix A.8.

Efficiency and Computational Cost Analysis. We extensively evaluated the efficiency of HiGoE against GoR from two perspectives: offline graph construction and online inference. As shown in Table 3 and Table 4, HiGoE exhibits superior efficiency in both phases. During the offline graph construction phase, GoR requires LLMs to generate both queries and answers from scratch. In contrast, while HiGoE involves LLM-based proposition generation, LLM-as-a-judge quality filtering, and PPR community summarization, its overall computational cost per document remains significantly lower than GoR's dual-generation process (Table 3). Furthermore, for online retrieval inference (ignoring the LLM generation time), Table 4 shows that HiGoE consistently requires less time than GoR, yielding an average reduction of about 9%. The online efficiency gains are most evident

Dataset	GoR(s/doc)	HiGoE(s/doc)
QMSum	270.68	**243.55**
WCEP	263.52	**214.74**
BookSum	223.04	**168.39**
GovReport	317.02	**186.31**
SQuALITY	272.99	**228.67**
Average	269.45	**208.33**

Table 3: **Computational cost comparison for offline graph construction.** We report the overall construction time (seconds/document) across five datasets. GoR's construction time includes query and answer generation, while HiGoE's includes proposition extraction, filtering, and community summarization.

Dataset	GoR(s/query)	HiGoE(s/query)
QMSum	0.504	**0.451**
WCEP	0.501	**0.417**
BookSum	0.429	**0.421**
GovReport	0.503	**0.440**
SQuALITY	0.402	**0.394**
Average	0.468	**0.425**

Table 4: **Inference time comparison between GoR and HiGoE across different datasets.** We report the online inference time (seconds/query, excluding LLM generation time).

on datasets with longer inputs, such as WCEP and GovReport, where HiGoE achieves reductions of 16.8% and 12.5%, respectively. In these cases, the hierarchical organization effectively reduces redundant retrieval from overlapping passages. On relatively shorter-input datasets like BookSum and SQuALITY, HiGoE still delivers slightly better efficiency.

5 Related Work

Long-context Summarization with LLMs. Recent LLMs have advanced long-context processing (Grattafiori et al., 2024; Yang et al., 2025; Guo et al., 2025). Existing approaches either extend context windows (Chen et al., 2024; Ding et al., 2024; Zhao et al., 2025) or adopt RAG (Yue et al., 2025; Jin et al., 2025). Long-context LLMs can process extended inputs, but they struggle with redundancy and loss of details. RAG is more efficient but mainly relies on surface similarity. GoR (Zhang et al., 2025a) mitigate some issues but remain constrained by LLM hallucinations and absence of hierarchical summaries.

Graph-based Retrieval-augmented Generation. Graphs have become an effective tool for modeling structural relations in RAG (Han et al., 2024). GraphRAG (Edge et al., 2024) builds a graph index over document elements and applies community detection for query-focused summarization. GNN-RAG (Mavromatis and Karypis, 2025) and GNN-Ret (Li et al., 2025b) leverage GNNs to connect related chunks for multi-hop retrieval. FG-RAG (Hong et al., 2025) employs context-aware entity expansion to enhance fine-grained details in query-focused tasks, while HGOT (Fang et al., 2024) utilizes a hierarchical graph of thoughts to decompose complex queries and improve factuality. Other approaches like GRAG (Hu et al., 2025) and Clue-RAG (Su et al., 2025) focus on retrieving subgraphs or multi-partite connections to enable joint reasoning. Despite recent progress, existing approaches remain ill-suited for long-context summarization. Generative methods (e.g., GoR and HGOT) build graphs over unverified model-generated thoughts, resulting in a verification gap, while entity-centric approaches (e.g., FG-RAG) emphasize local semantics without recovering document-level thematic structure. Consequently, neither paradigm jointly ensures faithfulness and global coherence as ours.

6 Conclusion

We present HiGoE, a proposition-evidence hierarchical retrieval framework for long-context summarization. Unlike prior generative graph-based RAG methods that suffer from error propagation, HiGoE builds graphs from verifiable atomic facts via a two-stage filtering pipeline, which effectively reduces hallucinations and ensures a reliable semantic basis. With Personalized PageRank for hierarchical abstraction and an Enhanced GAT trained with a joint contrastive-ranking loss, our model captures global document structure and learns task-adaptive node representations. A unified vector space and cross-community top-k filtering further enable diverse, non-redundant retrieval matching both global and local query intents. Experiments on five long-context benchmarks show that HiGoE outperforms strong retrieval-based baselines and long-context models by alleviating attention dilution. Comprehensive analyses verify its higher computational efficiency and improved factual accuracy.

Limitations

While HiGoE achieves significant improvements, several limitations remain. First, the construction of proposition–evidence graphs relies on LLM-based proposition extraction, which may still introduce occasional errors or omissions, especially for highly technical or ambiguous content. Second, the efficiency of HiGoE depends on the quality of intermediate retrieval and clustering, because suboptimal parameter choices (e.g., the number of extracted propositions or retrieved chunks) may lead to redundancy or information loss. Third, our experiments focus on summarization benchmarks in English, and the generalizability to other domains and languages remains to be validated. Finally, although Enhanced GAT improves representation learning, scaling to extremely large graphs may introduce computational overhead, suggesting the need for further optimization or approximation strategies.

Future work could explore multilingual adaptation, dynamic graph construction, and integration with reasoning-oriented LLMs to further improve robustness and scalability.

Acknowledgements

Long Yuan is supported by NSFC62472225. Zi Chen is supported by NSFC62402216 and the Natural Science Foundation of Jiangsu Province under Grant BK20241381.

References

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136.

Yapei Chang, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2024. Booookscore: A systematic exploration of book-length summarization in the era of LLMs. In *The Twelfth International Conference on Learning Representations*.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2024. LongloRA: Efficient fine-tuning of long-context large language models. In *The Twelfth International Conference on Learning Representations*.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. 2024. LongroPE: Extending LLM context window beyond 2 million tokens. In *Forty-first International Conference on Machine Learning*.

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*.

Yihao Fang, Stephen Thomas, and Xiaodan Zhu. 2024. HGOT: Hierarchical graph of thoughts for retrieval-augmented in-context learning in factuality evaluation. In *Proceedings of the 4th Workshop on Trustworthy Natural Language Processing (TrustNLP 2024)*, pages 118–144, Mexico City, Mexico. Association for Computational Linguistics.

Tao Feng, Pengrui Han, Guanyu Lin, Ge Liu, and Jiaxuan You. 2024. Thought-retriever: Don't just retrieve raw data, retrieve thoughts. In *International Conference on Learning Representations Workshop: How Far Are We From AGI*.

Joseph Gesnouin, Yannis Tannier, Christophe Gomes Da Silva, Hatim Tapory, Camille Brier, Hugo Simon, Raphael Rozenberg, Hermann Woehrel, Mehdi El Yakaabi, Thomas Binder, and 1 others. 2024. Llamandement: Large language models for summarization of french legislative proposals. *arXiv preprint arXiv:2401.16182*.

Demian Gholipour Ghalandari, Chris Hokamp, Nghia The Pham, John Glover, and Georgiana Ifrim. 2020. A large-scale multi-document summarization dataset from the Wikipedia current events portal. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1302–1308, Online. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Haoyu Han, Yu Wang, Harry Shomer, Kai Guo, Jiayuan Ding, Yongjia Lei, Mahantesh Halappanavar, Ryan A Rossi, Subhabrata Mukherjee, Xianfeng Tang, and 1

others. 2024. Retrieval-augmented generation with graphs (graphrag). *arXiv preprint arXiv:2501.00309*.

Yubin Hong, ChaoFan Li, Jingyi Zhang, and Yingxia Shao. 2025. Context-aware fine-grained graph rag for query-focused summarization. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 4802–4806.

Yuntong Hu, Zhihan Lei, Zheng Zhang, Bo Pan, Chen Ling, and Liang Zhao. 2025. GRAG: Graph retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4145–4157, Albuquerque, New Mexico. Association for Computational Linguistics.

Haoyu Huang, Yongfeng Huang, Junjie Yang, Zhenyu Pan, Yongqiang Chen, Kaili Ma, Hongzhi Chen, and James Cheng. 2025. Retrieval-augmented generation with hierarchical knowledge. *arXiv preprint arXiv:2503.10150*.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1419–1436, Online. Association for Computational Linguistics.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

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, Vienna, Austria. Association for Computational Linguistics.

Feng Kang, Rong Jin, and Rahul Sukthankar. 2006. Correlated label propagation with application to multi-label learning. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1719–1726. IEEE.

Vladimir Karpukhin, Barlas Oguz, 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.

Frederic Kirstein, Terry Ruas, Robert Kratel, and Bela Gipp. 2024. Tell me what I need to know: Exploring LLM-based (personalized) abstractive multi-source meeting summarization. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 920–939, Miami, Florida, US. Association for Computational Linguistics.

Wojciech Kryscinski, Nazneen Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir Radev. 2022. BOOKSUM: A collection of datasets for long-form narrative summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6536–6558, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, Kai Shu, Lu Cheng, and Huan Liu. 2025a. From generation to judgment: Opportunities and challenges of LLM-as-a-judge. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 2757–2791, Suzhou, China. Association for Computational Linguistics.

Shilong Li, Yancheng He, Hangyu Guo, Xingyuan Bu, Ge Bai, Jie Liu, Jiaheng Liu, Xingwei Qu, Yangguang Li, Wanli Ouyang, Wenbo Su, and Bo Zheng. 2024a. GraphReader: Building graph-based agent to enhance long-context abilities of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12758–12786, Miami, Florida, USA. Association for Computational Linguistics.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. 2024b. Long-context llms struggle with long in-context learning. *arXiv preprint arXiv:2404.02060*.

Zijian Li, Qingyan Guo, Jiawei Shao, Lei Song, Jiang Bian, Jun Zhang, and Rui Wang. 2025b. Graph neural network enhanced retrieval for question answering of large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6612–6633, Albuquerque, New Mexico. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 2023. How to train your dragon: Diverse augmentation towards generalizable dense retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6385–6400, Singapore. Association for Computational Linguistics.

Costas Mavromatis and George Karypis. 2025. GNN-RAG: Graph neural retrieval for efficient large language model reasoning on knowledge graphs. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16682–16699, Vienna, Austria. Association for Computational Linguistics.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Juan Ramos and 1 others. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. New Jersey, USA.

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*. Association for Computational Linguistics.

Stephen Robertson, Hugo Zaragoza, and 1 others. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Yaodong Su, Yixiang Fang, Yingli Zhou, Quanqing Xu, and Chuanhui Yang. 2025. Clue-rag: Towards accurate and cost-efficient graph-based rag via multi-partite graph and query-driven iterative retrieval. *arXiv preprint arXiv:2507.08445*.

Hao Sun, Hengyi Cai, Yuchen Li, Xuanbo Fan, Xiaochi Wei, Shuaiqiang Wang, Yan Zhang, and Dawei Yin. 2025. Enhancing retrieval-augmented generation via evidence tree search. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24116–24127, Vienna, Austria. Association for Computational Linguistics.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, and 1 others. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

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

Alex Wang, Richard Yuanzhe Pang, Angelica Chen, Jason Phang, and Samuel R. Bowman. 2022. SQuALITY: Building a long-document summarization dataset the hard way. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1139–1156, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zhanghao Wu, Paras Jain, Matthew Wright, Azalia Mirhoseini, Joseph E Gonzalez, and Ion Stoica. 2021. Representing long-range context for graph neural networks with global attention. *Advances in neural information processing systems*, 34:13266–13279.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24.

Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. 2018. Representation learning on graphs with jumping knowledge networks. In *International conference on machine learning*, pages 5453–5462. pmlr.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Zhenrui Yue, Honglei Zhuang, Aijun Bai, Kai Hui, Rolf Jagerman, Hansi Zeng, Zhen Qin, Dong Wang, Xuanhui Wang, and Michael Bendersky. 2025. Inference scaling for long-context retrieval augmented generation. In *The Thirteenth International Conference on Learning Representations*.

Haozhen Zhang, Tao Feng, and Jiaxuan You. 2025a. Graph of records: Boosting retrieval augmented generation for long-context summarization with graphs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23780–23799, Vienna, Austria. Association for Computational Linguistics.

Qinggang Zhang, Zhishang Xiang, Yilin Xiao, Le Wang, Junhui Li, Xinrun Wang, and Jinsong Su. 2025b. FaithfulRAG: Fact-level conflict modeling for context-faithful retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 21863–21882, Vienna, Austria. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Xinping Zhao, Dongfang Li, Yan Zhong, Boren Hu, Yibin Chen, Baotian Hu, and Min Zhang. 2024. SEER: Self-aligned evidence extraction for retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3027–3041, Miami, Florida, USA. Association for Computational Linguistics.

Yunlong Zhao, Haoran Wu, and Bo Xu. 2025. Leveraging attention to effectively compress prompts for long-context llms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 26048–26056.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. QMSum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5905–5921, Online. Association for Computational Linguistics.

A Experimental Details

A.1 Datasets

We evaluate our HiGoE framework on five representative long-context summarization benchmarks, following the evaluation setup of GoR. These datasets span diverse domains and discourse styles. We present dataset statistics in Table 5.

- **QMSum** (Zhong et al., 2021). QMSum is a meeting summarization dataset containing multi-speaker long meeting transcripts with both general summaries and 1,808 query-focused summaries across 232 meetings. Each query targets specific aspects of the discussion, making QMSum a challenging benchmark for selective and span-grounded meeting summarization.
- **WCEP** (Gholipour Ghalandari et al., 2020). WCEP is a multi-document news summarization corpus constructed from the Wikipedia Current Events Portal. It provides human-written event summaries paired with clusters of related news articles, capturing evolving descriptions of real-world events across sources.
- **BookSum** (Kryscinski et al., 2022). BookSum is a long-form narrative summarization dataset built from full-length literary works, providing paragraph, chapter, and book-level abstractive summaries. It emphasizes discourse coherence and long-range causal and temporal reasoning, making it a challenging benchmark for narrative understanding.
- **GovReport** (Huang et al., 2021). GovReport is a long-document summarization dataset composed of extensive U.S. government and Congressional reports paired with expert-written executive summaries. It emphasizes factual compression over large contexts and serves as a challenging benchmark for long-range informational synthesis.
- **SQuALITY** (Wang et al., 2022). SQuALITY is a question-focused summarization dataset built from public-domain short stories, containing human-written summaries conditioned on specific queries. It evaluates a model's ability to generate faithful, concise, and targeted summaries grounded in the source narrative.

A.2 Evaluation Metrics

Following GoR, we evaluate models using ROUGE-1, ROUGE-2, and ROUGE-L (Lin, 2004). ROUGE is a recall-oriented framework for summary quality assessment that compares system-generated summaries against human references by measuring overlapping textual units. ROUGE-1 and ROUGE-2 capture unigram and bigram overlaps, reflecting lexical coverage, while ROUGE-L measures the longest common subsequence, providing a more flexible assessment of fluency and structural alignment. These metrics have been widely adopted in large-scale summarization evaluations and remain standard for benchmarking factual and structural faithfulness in summarization.

A.3 Baselines

To comprehensively evaluate HiGoE, we compare it against a diverse set of representative baselines spanning multiple retrieval and reasoning paradigms. This includes classical lexical-based methods (*sparse retrievers*), neural semantic models (*dense retrievers*), hybrid approaches combining lexical and semantic signals, graph-based embedding and retrieval methods, as well as long-context and reasoning-oriented large language models. By covering these categories, we aim to benchmark HiGoE against state-of-the-art techniques that capture a wide spectrum of retrieval strategies, structural reasoning, and context utilization capabilities.

- **Node2Vec** (Grover and Leskovec, 2016). A random-walk–based graph embedding method that learns continuous feature representations for nodes in a network. It generates node embeddings by simulating flexible, biased random walks that capture diverse network neighborhoods, allowing the learned representations to preserve both local and global structural information.
- **Sparse Retrievers**: **TF-IDF** (Ramos et al., 2003) and **BM25** (Robertson et al., 2009). These are classical lexical-matching retrieval methods that rely on word frequency statistics and term weighting to rank documents based on query relevance. TF-IDF measures the importance of terms within a document relative to the corpus, while BM25 extends this by incorporating term saturation and document length normalization for more robust ranking. Although effective for exact keyword matching and many traditional IR tasks, sparse retrievers often struggle with semantic variation, paraphrasing, and understanding queries beyond surface-level word overlap.
- **Dense Retrievers**: **Contriever** (Izacard et al.,

Dataset	QMSum	WCEP	BookSum	GovReport	SQuALITY
#Train	162	400	400	400	100
#Test	30	30	30	30	27
Average Input Token Length	18.2K	10.7K	20.9K	17.2K	7.8K
Average Output Token Length	0.15K	0.08K	0.47K	0.65K	0.71K

Table 5: Dataset statistics

2022), **DPR** (Karpukhin et al., 2020), **Dragon** (Lin et al., 2023), and **SBERT** (Reimers and Gurevych, 2019). These are neural embedding models that map both queries and passages into a shared semantic space, allowing retrieval based on semantic similarity rather than exact lexical matches. Contriever is a self-supervised retriever that learns high-quality document embeddings without labeled data; DPR is trained on question-passage pairs to retrieve relevant passages effectively; Dragon leverages diverse data augmentation strategies to improve generalization in dense retrieval; and SBERT adapts BERT into a siamese network to produce semantically meaningful sentence embeddings suitable for tasks such as similarity search and clustering.

- **Hybrid Retriever**. Combines BM25 and DPR results using Reciprocal Rank Fusion (RRF), balancing sparse lexical signals with dense semantic similarity for more robust retrieval coverage.
- **Long-context LLMs**: **Gemma-8K** (Team et al., 2024) and **Mistral-8K** (Jiang et al., 2023). These are Transformer-based language models equipped with extended context windows of up to 8K tokens, enabling them to process and reason over long documents directly. Such models are capable of capturing document-level dependencies and performing complex comprehension tasks without relying on external retrieval. However, they remain limited by computational cost, potential input redundancy, and the practical constraints of context length, which can affect performance on tasks requiring reasoning over extremely large or multi-document contexts.
- **Full-context Input**. A retrieval-free baseline feeding the entire document into an LLM. It uses two settings: GoR's original all-MiniLM-L6-v2 (randomly sampling spans for long documents) and GPT-4-Turbo, a strong upper bound supporting up to 128K tokens.
- **Thought Retriever (Thought-R)** (Feng et al., 2024). A reasoning-oriented retrieval method designed to enhance the faithfulness and logical coherence of large language model outputs. Instead of retrieving raw data chunks, it retrieves intermediate "thoughts" or rationale segments generated by the LLM during previous queries. These thoughts are filtered, organized in a thought memory, and selectively retrieved to guide the LLM in addressing new queries, effectively enabling reasoning over arbitrarily long external knowledge without being constrained by context length. Thought-Retriever has demonstrated improved performance on tasks requiring deep reasoning and long-context comprehension, serving as a model-agnostic framework for reasoning-augmented retrieval.
- **Graph of Records (GoR)** (Zhang et al., 2025a). GoR is a graph-enhanced retrieval generation framework designed to leverage the often-neglected potential of LLM-generated historical responses. Distinct from traditional retrieval methods, it constructs a graph by establishing edges between retrieved text chunks and their corresponding LLM-generated responses via a retrieve-then-generate paradigm. To capture intricate correlations without explicit supervision, GoR employs a Graph Neural Network (GNN) optimized through a novel self-supervised objective: it utilizes BERTScore to derive pseudo-labels for node ranking, driving a joint optimization of contrastive and pair-wise ranking losses. As the strongest existing graph-based baseline for long-context global summarization, GoR effectively bridges the gap between scattered text chunks and global insights by exploiting the semantic bridges provided by historical records.

A.4 Additional Explanation on Self-Supervised Training

For each document, we segment the text into fixed-length chunks and use an LLM to generate a proposition from each chunk, yielding N propositions $\{\mathbf{q}_i\}_{i=1}^{N}$ whose source chunks c_i are known. Given the constructed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we compute the BERTScore similarity between each source chunk c_i and all nodes $v \in \mathcal{V}$, producing a ranking list $\mathcal{R}_i = [v^+_{i(1)}, v^-_{i(2)}, \ldots, \ldots, v^-_{i(|\mathcal{V}|)}]$ sorted by descending similarity, which serves as the self-supervised signal. Nodes from other documents in the same batch serve as additional negatives. During training, the model receives only the proposition $\mathbf{q}_i$ (not the source chunk) and is optimized to produce similarity scores whose ordering approximates the BERTScore derived ranking of c_i. This matches real usage: a query functions as an abstractive summarization request over either the entire document or a specific local segment, analogous to a proposition-level summary query, and the model is expected to retrieve the most relevant summary or evidence node. The contrastive loss pulls $\mathbf{q}_i$ toward its top-ranked node while pushing it away from negative ones, the hard-negative margin term penalizes cases where the hardest negative is insufficiently separated, and the focal ranking loss encourages the full predicted ordering to align with $\mathcal{R}_i$, with greater emphasis on difficult or ambiguous pairs. Together, these objectives yield node representations that faithfully capture fine-grained document semantics.

A.5 Additional Implementation Details

We segment each document into 256-token chunks with an overlap of 32 tokens using LangChain's TokenTextSplitter. For each long document, we use GPT-4 to generate verifiable propositions, and compared to larger models, it is more cost-effective and efficient, yet sufficient for high-quality propositions. We use Contriever to retrieve the top 6 chunks as supporting passages for each proposition, forming the initial proposition–evidence graph. For hierarchical enhancement, we apply PPR ($\alpha = 0.12$, $max\_iter = 150$) to form communities and generate community summaries with GPT-4 as well. All graph nodes are initialized with Contriever embeddings (for propositions, evidences, chunks and summaries). Enhanced GAT (2 layers, 768 hidden dim, 4 heads) refines these Contriever embeddings by leveraging graph structure, yielding graph-enhanced node representations. For retrieval, queries are encoded with Contriever, and at inference, the query is concatenated with retrieved chunks and selected proposition–evidence texts before being fed to LLaMA-2-7b-chat for generation. The query typically reflects a global or local summary of the underlying document, analogous to the propositions we derive during graph construction from individual document chunks. We implement our proposed method using PyTorch and Deep Graph Library (DGL). All experiments are conducted on a single NVIDIA GeForce RTX 4090 GPU (24GB). All experiments use fixed random seeds for reproducibility. We present detailed hyper-parameters on the QMSum, WCEP, BookSum, GovReport, and SQuALITY datasets in table 6.

A.6 Effect of the Number of Extracted Propositions

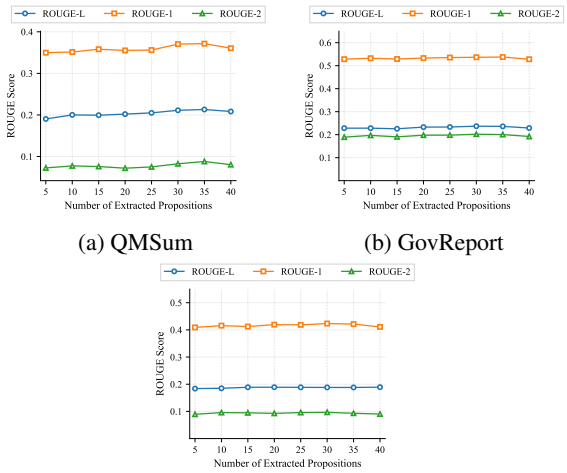
To further substantiate this conclusion, we present results on three additional datasets (QMSum, GovReport, and SQuALITY) in Figure 5. Consistent with our previous observations, performance across ROUGE-1, ROUGE-2, and ROUGE-L generally improves as the number of propositions increases, but it eventually exhibits a downward trend at higher counts. This confirms that excessive extraction introduces low-quality content that interferes with the results. Synthesizing the results from all five datasets, the configuration of 30 consistently ranks as either the best or a near-top choice. This empirical evidence solidifies our adoption of 30 as the robust default for HiGoE.

A.7 Effect of the Number of Retrieved Chunks

To further validate this observation, we present results on three additional datasets (QMSum, GovReport, and SQuALITY) in Figure 6. Across these datasets, the ROUGE-1, ROUGE-2, and ROUGE-L metrics also demonstrate a rise-then-fall pattern as the number of retrieved chunks increases from 1 to 10. Notably, performance peaks around 6 chunks for all three metrics across these datasets, aligning with the trends observed in WCEP and BookSum. Combining results from all five datasets, retrieving 6 chunks consistently ranks as either the best or the second-best configuration, showing robust and stable performance. Thus, this empirical evidence across diverse datasets solidifies our choice of 6 as the default number of retrieved chunks.

Dataset	QMSum	WCEP	BookSum	GovReport	SQuALITY
Chunk Size	256	256	256	256	256
Chunk Overlap	32	32	32	32	32
PPR Teleport Probability α	0.12	0.12	0.12	0.15	0.12
PPR Max Iterations	100	100	100	100	100
#Enhanced GAT Layers	2	2	2	2	2
#Enhanced GAT Heads	4	4	4	4	4
Hidden Dimension	768	768	768	768	768
Dropout Rate	0.2	0.1	0.2	0.5	0.1
Batch Size	32	32	32	32	32
Epoch	300	300	300	300	300
Learning Rate	1e-3	1e-3	1e-3	1e-3	1e-3
Focal Ranking Loss α	2.0	2.0	2.0	2.0	2.0
Focal Ranking Loss β	1.0	1.0	1.0	1.0	1.0
Hard Negative Margin γ	0.1	0.1	0.1	0.1	0.1
Loss Coefficient λ	0.9	0.7	0.2	0.7	0.4

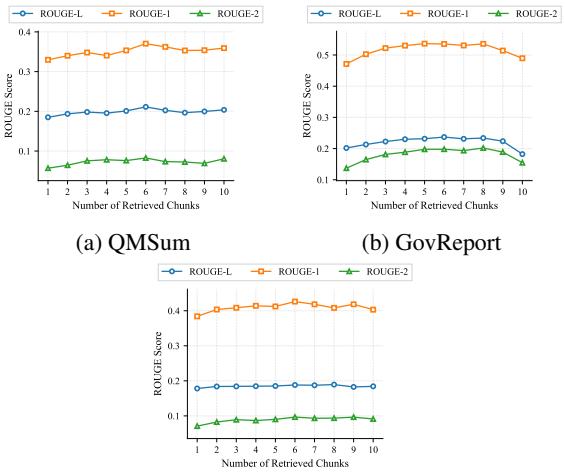
Table 6: Hyper-parameters



(a) QMSum (b) GovReport

(c) SQuALITY

Figure 5: **Effect of the number of extracted propositions w.r.t. R-L, R-1 and R-2**. We show the results on the QMSum, GovReport and SQuALITY datasets.



(a) QMSum (b) GovReport

(c) SQuALITY

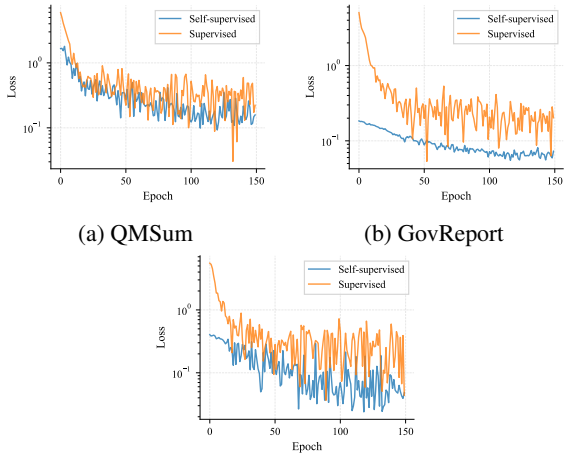
Figure 6: **Effect of the number of retrieved chunks w.r.t. R-L, R-1 and R-2**. We show the results on the QMSum, GovReport and SQuALITY datasets.

A.8 Global Summary Supervised Training vs Self-Supervised Training

We show additional results on the QMSum, GovReport and SQuALITY datasets in Figure 7. Across these datasets, self-supervised training consistently achieves lower and more stable losses compared to supervised training. These results align with our prior findings on WCEP and BookSum, reinforcing that self-supervised training, which leverages fine-grained local evidence, avoids the optimization noise introduced by the coarse-grained, weakly aligned supervision of global summaries. Thus, across all five datasets, self-supervised training demonstrates superior loss stability and magnitude, validating its ability to guide the model toward better node representations through precise learning signals.

A.9 Additional Explanation on Global Summary Supervised Training Experiment

In the supervised setting, each document provides a single global summary, usually appearing at the beginning of the document, which we encode as the



(a) QMSum (b) GovReport

(c) SQuALITY

Figure 7: **Differences between self-supervised and supervised training w.r.t loss**. We show the results on the QMSum, GovReport and SQuALITY datasets.

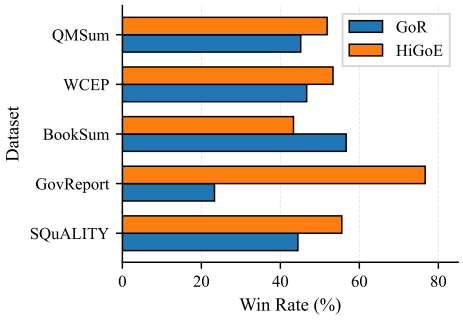


Figure 8: **LLM evaluation w.r.t overall win rates on QMSum, WCEP, BookSum, GovReport, and Squality**.

query vector. To ensure a fair comparison with the self-supervised setup, where each document yields N proposition-level queries from local chunk summaries, we replicate the global summary query N times so that both regimes operate on the same number of query instances. Unlike self-supervised training, which derives fine-grained BERTScore-based rankings aligned with proposition source chunks, supervised training directly uses the global summary query. At each step, the Enhanced GAT encodes all graph nodes, and dot-product similarities between the query and node representations are normalized via `log_softmax` to form a node distribution. We then treat the node with the highest current similarity as the pseudo-target for that iteration and optimize the model using a negative log-likelihood (NLL) loss to maximize the probability assigned to this dynamically chosen target. This training procedure contrasts with the fine-grained, chunk-aligned signal used in self-supervised training and highlights the limitations of relying solely on a single global summary as the supervisory signal.

A.10 LLM Evaluation

We evaluate pairwise summary quality using GPT-5.1, driven by the prompt shown in D. For each example we present two candidate summaries (GoR vs. HiGoE), along with the same test query and the golden answer, and ask the judge LLM to compare the two outputs according to five criteria: *Information Accuracy*, *Comprehensiveness*, *Adherence to the Golden Answer*, *Diversity*, and *Empowerment*. The judge was instructed to pick the better answer for each criterion, provide a brief justification, and then select an overall winner. Figure 8 reports the aggregated overall win rates for GoR and HiGoE on five datasets.

HiGoE attains higher overall win rates on QMSum, WCEP, GovReport and SQuALITY, whereas GoR performs better on BookSum. These results suggest that proposition–evidence and hierarchical aggregation strategy improves judged summary quality for most domains, especially for structured and factual corpora such as GovReport. In contrast, BookSum is a narrative and discourse-heavy dataset, and it remains a challenging setting where GoR occasionally produces outputs that are preferred by human evaluators.

B Other Related Work

Evidence Aggregation and Hierarchical Summarization in Retrieval-Augmented Generation. A complementary line of research focuses on improving evidence selection and aggregation to enhance retrieval-augmented generation. SEER (Zhao et al., 2024) learns to extract concise and self-aligned evidence from retrieved passages, addressing limitations of heuristic filtering and reducing redundant context. ETS (Sun et al., 2025) formulates evidence selection as a tree-search problem, leveraging MCTS to model inter-sentence dependencies and identify high-quality multi-sentence evidence sets. FaithfulRAG (Zhang et al., 2025b) tackles fact-level conflicts between retrieved context and parametric knowledge by explicitly modeling discrepancies and guiding LLMs through a self-thinking process to integrate conflicting facts. For long-context scenarios, LongRefiner (Jin et al., 2025) improves RAG efficiency through hierarchical document structuring and adaptive refinement,

which implicitly enables a coarse-to-fine summarization process over long documents. Beyond extraction. HIRAG (Huang et al., 2025) enhances evidence reasoning through hierarchical instruction tuning, equipping RAG models with progressively structured abilities for filtering, composition, and knowledge-grounded reasoning.

Together, these works highlight the importance of selecting, refining, and organizing evidence and leveraging hierarchical structures, yet most operate at the sentence or passage level. In contrast, HiGoE performs proposition–evidence aggregation with an explicit hierarchical graph structure, enabling more fine-grained and semantically grounded summarization.

C Case Study

This case study compares three outputs from the same source: (1) **GoR's Summary**, (2) **HiGoE's Summary**, and (3) **Golden Answer** (reference). It analyzes five criteria: information accuracy, comprehensiveness, adherence to the golden answer, diversity, and empowerment by highlighting concrete discrepancies (e.g., misplaced focus, omitted key information) and extraneous facts, then assessing each summary's alignment with the reference in precision and relevance.

The Summary Generated by GoR

The meeting discussed the **conceptual design** for a new remote control, with a focus on **user interface and industrial design**. **The project manager outlined the agenda and introduced the participants, including a user interface designer, an industrial designer, and a marketing representative.** The group discussed various aspects of the remote control, including **its appearance, functionality, and color scheme**. They also talked about the importance of evaluating the product and considering current trends in ruits and vegetables. The industrial designer presented their ideas for the remote control's design, and the group provided feedback and suggestions. The meeting concluded with the project manager summarizing the key points and **tasks assigned to each participant for the next meeting**.

The Summary Generated by HiGoE

The meeting focused on the **new remote control's conceptual design**, with key discussions on **UI and industrial design**. The project manager outlined the agenda and introduced the **UI designer, industrial designer, and marketing representative**. The UI designer presented **three concepts (Google, fancy and intelligent)** with the **first two approved and the third questioned**. The team **decided to integrate the first two, plus simple vocal commands**. For industrial design, the industrial designer proposed **wooden shell with LCD buttons**, **dual power supply**, and **low-power chips**, **all unanimously agreed**. The marketing rep shared market insights: **users want fancy look, technological innovation, and user-friendliness**. The meeting ended with **task assignments after discussing implementation**.

Golden Answer

This meeting was about conceptual design. The user interface designer first introduced three concepts in the user interface - Google controller, fancy controller, and intelligent controller. The first two were approved by the project manager, but the last one was seriously questioned. The group finally decided to mix the Google controller and fancy controller with some simple vocal commands. The discussion about product industrial design covered three aspects —material, alimentation, and chip. The industrial designer suggested using wood and having buttons with LCD, applying the classical battery and solar energy alimentation, and using low level chips, which was agreed by the group. The marketing expert gave three points from the market analysis. Users would like to have a fancy look and feel and the product should be technologically innovative. At the same time, being user-friendly was also important. Then the group discussed how to make these requirements into practice.

The bolded text indicates content that is more consistent with the golden answer. Across all five

evaluations, HiGoE provides a broader, more accurate view of the meeting, precisely capturing its core emphasis on conceptual design across user interface, industrial design, and marketing dimensions. It offers a structured and factual depiction of the discussion, incorporating a wider range of concrete decisions and specific proposals, enabling a clearer understanding of the meeting's true scope and outcomes. In contrast, GoR contains inaccurate information, such as mistakenly associating "market trends" with "current trends in fruits and vegetables", which is irrelevant to the remote control design. Overall, HiGoE better represents the meeting's multi-faceted, decision-driven nature, aligns more closely with the factual precision of the Golden Answer, and offers a far more balanced and informative depiction of the discussion.

D LLM Prompts

This appendix presents the full set of prompt templates used in our system, spanning five key components: proposition generation, proposition evaluation via LLM-as-a-Judge, community-level summarization, LLM-as-a-Judge evaluation for comparing model outputs, and retrieval-augmented generation (RAG). Together, these prompts define the behavior of each LLM module, ensure consistent interactions across components, and enable full reproducibility of our experimental pipeline.

D.1 LLM Prompts for Proposition Generation

To extract fine-grained semantic units from text, we employ the following prompt to guide the LLM to generate concise, factual, and verifiable propositions.

Prompt for proposition Generation

You are an excellent extractor of factual information and are highly skilled at identifying clear, verifiable propositions within a passage. Your goal is to produce exactly ONE concise factual proposition from the text provided below. The extracted proposition must be a single sentence (no more than 30 words), specific, and directly verifiable. Do not produce questions or ambiguous statements. Output only the proposition, with no additional text.
DOCUMENT:
{ document }

D.2 LLM Prompts for Community Summarization

For constructing higher-level graph nodes, we summarize multiple related text fragments into a unified semantic theme. The following prompt is used to generate these community-level summaries.

Prompt for Proposition Generation

As a top-tier Knowledge Architect, your mission is to read and comprehend the following collection of related text fragments. Synthesize these scattered pieces of information into a highly condensed, overarching core theme or summary. This summary will serve as the "title" or "central idea" for this knowledge cluster and must be both concise and information-dense.
[Text Fragments to be Summarized]:
{full_text}
[Core Theme Summary]:

D.3 LLM-as-a-Judge for Proposition Generation

To ensure the quality of extracted propositions, we adopt an LLM-as-a-judge framework. The following prompt instructs the LLM to evaluate each proposition along multiple quality dimensions.

Prompt for Proposition Evaluation

As an expert evaluator, please assess the quality of this extracted proposition based on the original text.
Original Text:
{original_text}
Extracted Proposition:
{proposition}
Please evaluate on these criteria (score 1–5 for each):
Factual Accuracy: Is the proposition factually consistent with the original text?
Verifiability: Can this proposition be verified or fact-checked?
Completeness: Does the proposition capture important information from the text?
Clarity: Is the proposition clear and unambiguous?
Specificity: Is the proposition specific enough to be meaningful?
Provide scores and a brief explanation.

D.4 LLM-as-a-Judge for for LLM Evaluation

To systematically compare GoR and HiGoE, we employ an LLM-as-a-judge framework. The following prompt describes the evaluation criteria used to assess the relative quality of two model-generated summaries.

LLM Evaluation - Instruction

You are an expert tasked with evaluating two answers to the same question based on five criteria.
Your goal is to evaluate two answers to the same question based on these criteria:
Information Accuracy: How factually correct is the information provided?
Comprehensiveness: How much detail does the answer provide to cover all aspects and details of the question?
Adherence to the Golden Answer: How closely does the answer align with the provided golden answer?
Diversity: How varied and rich is the answer in providing different perspectives and insights on the question?
Empowerment: How well does the answer help the reader understand and make informed judgments about the topic?
For each criterion, choose the better answer (either Answer 1 or Answer 2) and explain why. Then, select an overall winner based on these five categories.

The next prompt provides the concrete inputs passed to the judge model, including the question, the GOLDEN ANSWER, and two candidate summaries.

LLM Evaluation - Input

Here is the question: {query}
Here is the golden answer for reference: {golden_answer}
Here are the two answers:
Answer 1: {answer1}
Answer 2: {answer2}
Evaluate both answers using the five criteria listed above and provide detailed explanations for each criterion. Avoid any potential bias and ensure that the order in which the answers were presented does not affect your judgment.

Finally, we specify the expected output format to ensure that the judge model produces structured, consistent, and comparable evaluations across all examples.

LLM Evaluation - Output

Output your evaluation in the following JSON format. Do not output any other text before or after the JSON.

```
{
"Information Accuracy": {
 "Winner": "[Answer 1 or Answer 2]",
 "Explanation": "[Provide explanation
 here]"
 },
"Comprehensiveness": {
 "Winner": "[Answer 1 or Answer 2]",
 "Explanation": "[Provide explanation
 here]"
 },
"Adherence to the Golden Answer": {
 "Winner": "[Answer 1 or Answer 2]",
 "Explanation": "[Provide explanation
 here]"
 },
"Diversity": {
 "Winner": "[Answer 1 or Answer 2]",
 "Explanation": "[Provide explanation
 here]"
 },
"Empowerment": {
 "Winner": "[Answer 1 or Answer 2]",
 "Explanation": "[Provide explanation
 here]"
 },
"Overall Winner": {
 "Winner": "[Answer 1 or Answer 2]",
 "Explanation": "[Summarize why this
 answer is the overall winner based
 on the five criteria]"
 }
}
```

D.5 LLM Prompts for RAG

During reasoning and query answering, we use the following prompt to guide the LLM in producing grounded responses based on retrieved evidence.

Prompt for Proposition Generation

Refer to the following supporting materials and answer the question with brief but complete explanations.
SUPPORTING MATERIALS:
{materials}
QUESTION:
{question}