Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

GraphCLaRa uses graph guided document selection for multihop question answering with compressed memory

Cong Tran
Posts and Telecommunications Institute of Technology
Hung Bui Viet
Hanoi University of Science and Technology
Hai Pham Van
haipv@soict.hust.edu.vn

Hanoi University of Science and Technology

Research Article

Keywords:

Posted Date: November 4th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10814365/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

GraphCLaRa uses graph guided document selection for multihop question answering with compressed memory

Cong Tran
Posts and Telecommunications Institute of Technology, Ha Dong, Hanoi, Vietnam

Hung Bui Viet
School of Information Technology and Communication,
Hanoi University of Science and Technology, Hai Ba Trung, Hanoi, Vietnam

Hai Van Pham
School of Information Technology and Communication,
Hanoi University of Science and Technology, Hai Ba Trung, Hanoi, Vietnam

September 8, 2026

Abstract Retrieval-Augmented Generation (RAG) for multi-hop question answering suffers from noisy context and weak retrieval–generation alignment. Apple's Continuous Latent Reasoning (CLaRa) improves efficiency by compressing retrieved documents into dense latent memory tokens, but this compression obscures the structural relationships between documents required for multi-hop reasoning. We propose **GraphCLaRa**, a lightweight ($\sim$1.3M-parameter) graph-based reranker for a *frozen* CLaRa reader. It constructs a query-aware document graph, propagates evidence with a Graph Attention Network (GAT), and selects a compact document subset with an adaptive context-budget head. Across four single- and multi-hop QA benchmarks, GraphCLaRa improves answer F1 on three of four datasets (up to $+6.6$) while using fewer documents, and trails by only 1.1 F1 on the retrieval-saturated benchmark. Our analysis shows that graph-guided selection consistently helps when retrieval leaves room for improvement, whereas on saturated retrieval tasks compression can only discard evidence. These results demonstrate that graph-based evidence propagation complements latent-memory compression.

1 Introduction

Multi-hop question answering (QA) requires retrieving and reasoning over evidence distributed across multiple documents rather than a single passage [1, 2]. Recent Retrieval-Augmented Generation (RAG) systems increasingly adopt latent-memory architectures that compress retrieved documents into compact continuous representations for efficient long-context reasoning [3, 4]. Apple's Continuous Latent Reasoning (CLaRa) is one such architecture, encoding documents into dense memory tokens while substantially reducing context length [4]. However, end-to-end QA remains sensitive to retrieval quality, as irrelevant documents introduce noise that degrades

downstream reasoning [5, 6].

Graph-based retrieval methods capture structural dependencies and evidence chains in multi-hop QA by modeling bridge entities and multi-step reasoning paths [7, 8]. However, they have been studied almost exclusively with text-conditioned readers, leaving it unclear whether graph-guided retrieval remains beneficial when reasoning is performed over compressed latent memory instead of raw text.

To address this question, we propose **GraphCLaRa**, a graph-guided reranking framework for latent-memory retrieval. Given a question and a fixed candidate pool, it constructs a query-aware document graph, propagates relational evidence with a Graph Attention Network (GAT) [9], and passes only a compact graph-curated subset to a frozen CLaRa reader. Across four QA benchmarks, GraphCLaRa improves answer F1 on three of four while using fewer documents, with the only regression on a retrieval-saturated benchmark (Section 4).

In summary, our contributions are:

- We propose **GraphCLaRa**, a lightweight graph-guided reranking framework that combines query-aware graph propagation with semantic query–document interaction via a Graph Attention Network (GAT), modeling multi-document reasoning before a frozen latent-memory reader.
- We show consistent improvements on three of four QA benchmarks (up to +6.6 F1) while reducing retrieved context, and we provide an analysis explaining when graph-guided selection benefits latent-memory retrieval.

2 Related Research

GraphCLaRa relates to two lines of work and departs from them by placing graph-guided *selection* before a frozen latent-memory generator.

Multi-hop retrieval and graph reasoning. Multi-hop dense retrieval assembles evidence chains iteratively [6, 10, 11, 12, 13]. A parallel line builds entity-, sentence-, or passage-level graphs over benchmarks such as HotpotQA [1], 2WikiMultiHopQA [14], and WikiHop [15] via hierarchical graph networks [7], select-answer-explain models [16], and knowledge-graph prompting for LLMs [17]. All graft graph structure onto a *text-conditioned* reader; *GraphCLaRa* instead leaves CLaRa's retrieval untouched and uses graph attention [9] — self-attention [18] on graphs, with GCNs [19] as background — only to rerank a fixed candidate pool before a frozen generator.

Contrastive retrieval and context compression. Dense retrievers and sentence encoders [5, 20, 21] built on pretrained transformers [22] typically use contrastive objectives with a single positive, which introduces false negatives when several gold documents exist; *GraphCLaRa* instead supervises with *all* gold indices through a multi-positive InfoNCE objective [23] (Section 3.6). Separately, retrieval-augmented generation [24, 3] and fusion-in-decoder readers [25, 26, 27] have been made cheaper by prompt compression [28, 29] and adaptive retrieval [30, 31], while latent-memory methods such as CLaRa [4] compress documents into continuous memory tokens. *GraphCLaRa* is complementary: it compresses at the *document-selection* level, keeping CLaRa frozen. Because CLaRa compresses retrieved documents into latent memory, the selector must choose a compact subset that survives compression — so our experiments test whether structured selection alone, with no generator fine-tuning or larger budgets, improves latent-memory QA.

3 Method

3.1 Problem Formulation

This work considers retrieval-augmented multi-hop question answering with a fixed candidate document pool. Each instance consists of a natural-language question q and a candidate set $\mathcal{D} = \{d_1, \ldots, d_N\}$ with $N = 20$ passages, following the standard CLaRa evaluation protocol [4]. A frozen BGE-large-en-v1.5 encoder [32] maps q and each d_i into a shared $d_e = 1024$-dimensional embedding space, producing the query embedding $\mathbf{e}_q$ and document embeddings $\mathbf{e}_i \in \mathbb{R}^{d_e}$, which remain fixed throughout training. During training, each instance additionally provides a gold supporting-document set $\mathcal{Y} \subseteq \{1, \ldots, N\}$ and a reference answer a.

From these frozen embeddings, the model learns two question-conditioned functions: (i) a ranking function $f_\theta(q, \mathcal{D}) \to \mathbf{s} = (s_1, \ldots, s_N) \in \mathbb{R}^N$ that assigns a score to every candidate document, and (ii) a budget predictor $g_\phi(q, \mathcal{D}) \to K_\star \in \mathcal{K} = \{5, 10, 20\}$. Selecting the top-$K_\star$ documents according to $\mathbf{s}$ produces the ordered context $\mathcal{S}_{K_\star}$, which is passed to the frozen CLaRa reader to generate the predicted answer $\hat{a}$.[1] The model therefore learns only document ranking and adaptive budget prediction, while both the encoder and reader remain frozen; answer F1 is used only for evaluation, never optimized directly.

3.2 System Overview

Figure 1 summarizes GraphCLaRa. Given a question q and a fixed candidate set $\mathcal{D} = \{d_1, \ldots, d_N\}$, frozen BGE embeddings are formed into a query-aware document graph. A lightweight graph retriever fuses each document with the query and propagates evidence through a two-layer GAT; two heads then produce a per-document ranking score and an adaptive budget $K_\star \in \{5, 10, 20\}$. The top-$K_\star$ documents are read by the frozen CLaRa reader. Only the graph retriever and budget head are trained; the BGE encoder and CLaRa reader remain frozen. Sections 3.3–3.6 define each component; standard components are cited where introduced.

3.3 Query-Aware Document Graph Construction

We define a query-aware adjacency matrix $A \in [0,1]^{N \times N}$ that determines which documents may exchange evidence during GAT propagation (Section 3.4). It is a deterministic, parameter-free function of the frozen BGE embeddings [32] with one hyperparameter, the similarity threshold $\tau = 0.3$, computed in two steps (Eqs. (1)–(2)).

Step 1 — Query-aware raw adjacency. For each document d_i we take its rescaled query relevance $\rho_i \triangleq \frac{1}{2}\big(\cos(\mathbf{e}_q, \mathbf{e}_i) + 1\big) \in [0,1]$ and combine it multiplicatively with the pairwise cosine into a raw adjacency $A^{\text{raw}} \in \mathbb{R}^{N \times N}$,

$$A^{\text{raw}}_{ij} \;=\; \cos(\mathbf{e}_i, \mathbf{e}_j) \cdot \underbrace{\frac{\rho_i + \rho_j}{2}}_{\mu_{ij} \in [0,1]}, \tag{1}$$

with $\mu_{ij} \triangleq (\rho_i + \rho_j)/2$. The multiplicative form keeps an edge strong only when two documents are both relevant to q and mutually similar, suppressing irrelevant clusters before propagation.

Step 2 — Sparsify, self-loop, clamp. We then sparsify the raw matrix by a threshold $\tau = 0.3$ (which also drops anti-correlated pairs, since $\tau > 0$), add unit self-loops (the $N \times N$ identity matrix I_N) so each node keeps a copy of its own features, and clip the result to yield the final adjacency, where $\mathbf{1}[\cdot] \in \{0,1\}$

[1] The frozen reader applies its own soft top-5 filter (`generation_top_k`= 5): each selected document is compressed into 16 latent memory tokens, and only the top $\min(5, |\mathcal{S}_{K_\star}|)$ are forwarded to the decoder.

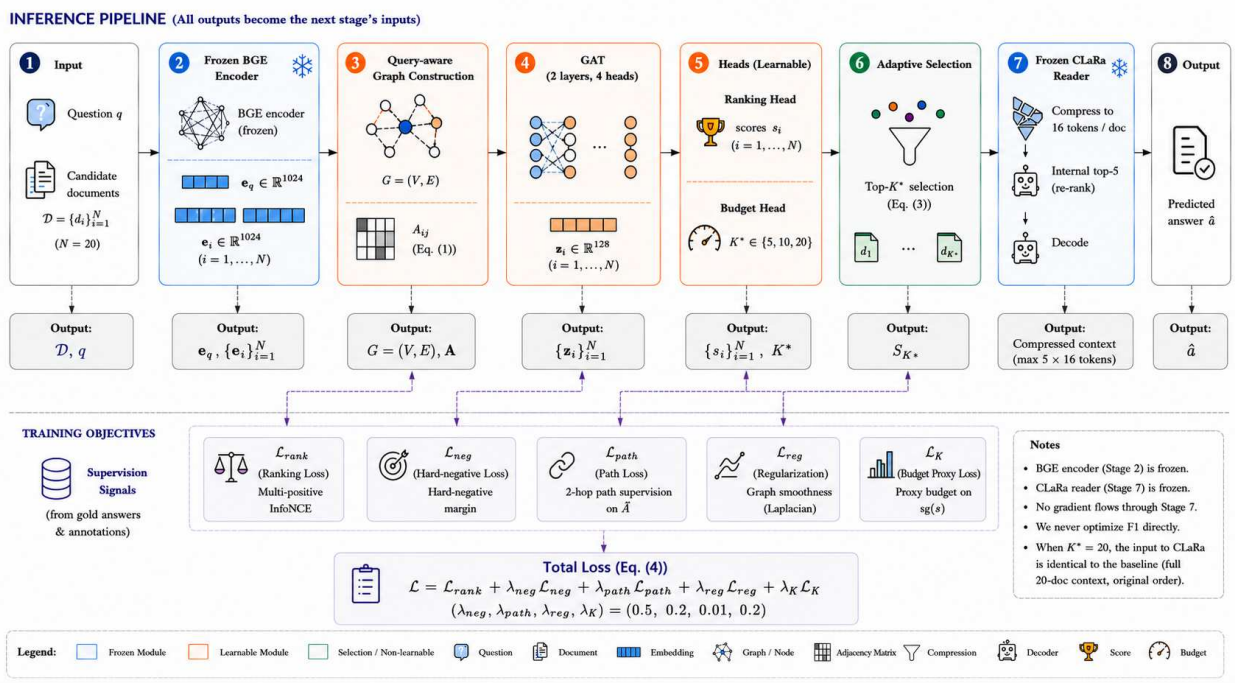


Figure 1: GraphCLaRa pipeline. Frozen BGE embeddings are converted into a query-aware document graph, reranked by a lightweight graph retriever, adaptively filtered, and passed to the frozen CLaRa reader. Blue modules are frozen, orange modules are trainable, and green denotes selection.

denotes the indicator function:

$$A_{ij} = \min\Big(A_{ij}^{\text{raw}}\,\mathbf{1}[A_{ij}^{\text{raw}} > \tau] + \mathbf{1}[i = j],\ 1\Big) \in [0, 1]. \tag{2}$$

The resulting matrix is symmetric and non-negative, with $A_{ii} = 1$.

3.4 Graph-Guided Reranking

The ranking function f_θ maps the frozen BGE embeddings and graph adjacency A to document scores $\mathbf{s}$ through four learnable components: query–document fusion, graph propagation, content scoring, and graph-residual fusion. The graph features are also pooled to produce the representation used by the adaptive budget predictor.

Step 1 — Query-aware document enhancement. Each frozen document embedding is fused with the query through the matching feature $[\mathbf{e}_i; \mathbf{e}_q; \mathbf{e}_i \odot \mathbf{e}_q]$, a standard pairwise interaction pattern [33]; GraphCLaRa adapts it with a two-layer MLP, residual connection, and LayerNorm:

$$\mathbf{h}_i = \text{LayerNorm}\big(\mathbf{e}_i + \text{MLP}_{\text{fuse}}([\mathbf{e}_i; \mathbf{e}_q; \mathbf{e}_i \odot \mathbf{e}_q])\big). \tag{3}$$

Following residual zero-initialization schemes [34], only the final projection is zero-initialized, so $\mathbf{h}_i = \mathbf{e}_i$ at initialization and training starts from the strong BGE prior; gradients remain non-zero because the preceding layers use standard initialization.

Step 2 — Graph propagation. A two-layer, four-head GATv1 [9] propagates $\{\mathbf{h}_i\}$ over the graph A,

attending only to neighbors with $A_{ij} > 0$ and using learnable residual projections. The layer widths are $1024 \to 256 \to 128$, producing graph features $\mathbf{z}_i$. As in standard GATv1, A enters attention only as a binary connectivity mask ($A_{ij} > 0$); the continuous edge weights are consumed instead by the learnable edge-refinement gate below (Eq. (4)) and the two-hop path supervision (Eq. (11)), both of which operate on the weighted adjacency.

We modify the GAT pipeline with a learnable edge-refinement step: a linear head predicts $\beta_i = \mathbf{w}_\beta^\top \mathbf{z}_i + b_\beta$ and gates the existing edges:

$$\tilde{A}_{ij} = \min\Big(A_{ij}\,\sigma\big(\tfrac{\beta_i+\beta_j}{2}\big) + \mathbf{1}[i=j], 1\Big), \qquad (4)$$

followed by one additional GAT layer over $\tilde{A}$ — three GAT layers in total when refinement is enabled (two propagation layers over A and one refinement layer over $\tilde{A}$). Because $\tilde{A}$ depends on learned features, graph supervision back-propagates through this one-pass refinement.

Step 3 — Content scoring. GraphCLaRa defines a content residual head that predicts $\delta_i^c = \mathrm{MLP}_c(\mathbf{h}_i)$, which refines the frozen cosine similarity:

$$s_i^c = \mathrm{stop_grad}(\cos(\mathbf{e}_q, \mathbf{e}_i)) + \delta_i^c, \qquad (5)$$

where $\mathrm{stop_grad}(\cdot)$ keeps the encoder frozen.

Step 4 — Graph-residual fusion. A gate $\gamma_i = \sigma(\mathbf{w}_\gamma^\top \mathbf{z}_i + b_\gamma)$ and residual $r_i = \mathbf{w}_r^\top \mathbf{z}_i + b_r$ combine the content and graph signals. The heads are initialized near the cosine prior.

We define the final ranking score as

$$\boxed{s_i = s_i^c \gamma_i + r_i.} \qquad (6)$$

Finally, the graph features are mean-pooled, $\bar{\mathbf{z}} = \frac{1}{N}\sum_i \mathbf{z}_i$, and passed to the adaptive budget predictor.

3.5 Document Selection Strategies

Given the scores $\mathbf{s}$, the selection stage forms the subset $\mathcal{S}_K$ passed to the reader. Let $\mathrm{TopK}_k(\mathbf{v})$ denote the indices of the k largest entries of $\mathbf{v}$. Following the motivation of adaptive retrieval [31, 30], all main experiments use adaptive selection: a two-layer budget head maps $\bar{\mathbf{z}} = \frac{1}{N}\sum_i \mathbf{z}_i$ to logits over $\mathcal{K} = \{5, 10, 20\}$, and predicts $K_\star = \arg\max_{k\in\mathcal{K}} [\mathrm{MLP}_g(\bar{\mathbf{z}})]_k$. The selected subset is

$$\mathcal{S}^{\mathrm{Ada}} = \begin{cases} \big(d_j \,:\, j \in \mathrm{TopK}_{K_\star}(\mathbf{s})\big) & \text{if } K_\star < N, \\ \mathcal{D} & \text{if } K_\star = N, \end{cases} \qquad (7)$$

In evaluation, the $K_\star = N$ branch reuses the baseline answer/top-K trace, making it safety-equivalent to the full-context baseline.

3.6 Training Objective

Since CLaRa is frozen, only the reranker f_θ and budget predictor g_ϕ are optimized using the supporting-document annotations $\mathcal{Y}$ and the model's internal quantities $\mathbf{s}$, $\tilde{A}$, $\mathbf{z}_i$, and $\bar{\mathbf{z}}$.

The total loss combines standard contrastive/margin/smoothness terms with two GraphCLaRa-specific terms, path supervision and proxy budget prediction:

$$\mathcal{L} = \mathcal{L}_{\mathrm{rank}} + \lambda_{\mathrm{neg}}\mathcal{L}_{\mathrm{neg}} + \lambda_{\mathrm{path}}\mathcal{L}_{\mathrm{path}} + \lambda_{\mathrm{reg}}\mathcal{L}_{\mathrm{reg}} + \lambda_K \mathcal{L}_K, \qquad (8)$$

with $(\lambda_{\mathrm{neg}}, \lambda_{\mathrm{path}}, \lambda_{\mathrm{reg}}, \lambda_K) = (0.5, 0.2, 0.01, 0.2)$.

Term 1 — Multi-positive InfoNCE ranking. The primary ranking objective adapts InfoNCE [23] to the multi-positive setting [35]:

$$\mathcal{L}_{\mathrm{rank}} = \log\left(\sum_{i=1}^{N} e^{s_i/T}\right) - \log\left(\sum_{i\in\mathcal{Y}} e^{s_i/T}\right), \qquad (9)$$

with temperature $T = 0.1$, encouraging all supporting documents to receive higher scores without competing against one another.

Term 2 — Hard-negative margin. Let $\mathcal{N}^- = \{1,\dots,N\} \setminus \mathcal{Y}$ and $\mathcal{H} = \mathrm{TopK}_\kappa(\mathbf{s}_{\mathcal{N}^-})$ with $\kappa = 3$. We use a hinge-style hard-negative margin objective related to triplet ranking losses [36]:

$$\mathcal{L}_{\text{neg}} = \frac{1}{|\mathcal{Y}||\mathcal{H}|} \sum_{p\in\mathcal{Y}} \sum_{h\in\mathcal{H}} [m_{\text{neg}} - s_p + s_h]_+. \quad (10)$$

where $m_{\text{neg}} = 0.5$.

Term 3 — Two-hop path supervision. For every gold pair $(p,p') \in \mathcal{P}$, let $B_{pp'} = \max\big(\tilde{A}_{pp'}, \max_{m\neq p,p'} \tilde{A}_{pm}\tilde{A}_{mp'}\big)$ be the strongest length-≤ 2 path strength. The path loss hinges it toward a margin $m_{\text{path}} = 0.2$:

$$\mathcal{L}_{\text{path}} = \frac{1}{|\mathcal{P}|} \sum_{(p,p')\in\mathcal{P}} [m_{\text{path}} - B_{pp'}]_+. \quad (11)$$

Term 4 — Graph smoothness. Following graph smoothness regularization [37], adjacent nodes are encouraged to receive similar scores:

$$\mathcal{L}_{\text{reg}} = \frac{1}{|\mathcal{E}|} \sum_{(i,j)\in\mathcal{E}} (s_i - s_j)^2. \quad (12)$$

where $\mathcal{E} = \{(i,j) : i \neq j,\, A_{ij} > 0\}$.

Term 5 — Budget prediction. The budget head is supervised by a retrieval-only proxy target $K^{\text{prox}}_\star = \min\{k \in \mathcal{K} : \mathcal{Y} \subseteq \mathrm{TopK}_k(\mathbf{s})\}$ (defaulting to 20 if no such k exists), computed on $\mathrm{stop_grad}(\mathbf{s})$. It is trained with target-weighted cross-entropy using weights $(w_5, w_{10}, w_{20}) = (0.65, 1.00, 0.85)$.

Curriculum. Training uses a two-stage curriculum. During epochs 0 and 1, $\lambda_K = 0$, allowing the ranker to stabilize before enabling budget supervision. After warm-up, $\lambda_K = 0.2$, while all other loss weights remain unchanged.

Dataset	Hops	Avg. $\lvert\mathcal{Y}\rvert$	Role in evaluation
NQ	1	$\sim$1	Single-hop control
HotpotQA	2	2	Bridge-entity reasoning
2Wiki	2	2	Compositional multi-hop QA
MuSiQue	2–4	2–4	Long evidence chains

Table 1: Evaluation datasets. All use the same input format: one question, 20 candidate documents, gold supporting-document indices, and a reference answer. "2Wiki" abbreviates 2WikiMultiHopQA.

4 Experiments

This section evaluates whether graph-guided selection improves frozen-reader QA and when it helps. After the data/setup (Sections 4.1–4.2), Section 4.3 reports the main comparison and Section 4.4 explains the per-dataset outcomes through retrieval headroom and answer sensitivity.

4.1 Datasets

We evaluate on four datasets from the public `apple/CLaRa_multi_stage` release in the gold-annotated `with_pos` setting: Natural Questions (NQ) [38], HotpotQA [1], 2WikiMultiHopQA [14], and MuSiQue [2]. These are the same datasets used by CLaRa [4], which allows a controlled comparison in which the reader is fixed and only the document selection mechanism changes.

Each example consists of a question q, a fixed candidate pool $\mathcal{D} = \{d_1,\dots,d_N\}$ with $N = 20$ documents, a gold supporting-document set $\mathcal{Y}$, and a reference answer a; the four datasets span single-hop to long-chain retrieval regimes (Table 1).

For evaluation, we keep only examples with non-empty supporting-document annotations. To avoid positional or shard-order bias, each dataset is sampled with a fixed seed and each candidate pool is shuffled with a deterministic per-example seed before being scored by both the baseline and GraphCLaRa. Thus, both systems receive the same

candidate documents in the same initial order, and any difference is due to GraphCLaRa's reranking and adaptive context selection.

4.2 Experimental Setup

Baseline. The primary comparison is the canonical CLaRa baseline (**CLaRa-BL**) [4]: all $N = 20$ documents, in shuffled original order, are passed to the frozen CLaRa-7B-E2E reader (`compression-16`) without external reranking or adaptive budgeting. Thus, any difference from GraphCLaRa comes from document ordering and context selection, not from reader fine-tuning.

Metrics. We report retrieval metrics over the document list delivered to the reader: R@k, P@k for $k \in \{2,5\}$, and MRR. For MuSiQue, which may require up to four supporting documents, we also report coverage@5 and all@5. End-to-end answer quality is measured by token-level F1 and Exact Match (EM) after standard SQuAD-style normalization.

Implementation. GraphCLaRa adds 1,291,271 trainable parameters on top of the frozen BGE-large-en-v1.5 encoder, with fp16 embedding caching. Training uses AdamW (lr 10^{-4}, weight decay 0.01), gradient accumulation of 4, mixed precision, and the two-stage curriculum in Section 3.6. Checkpoint selection uses all@5 $+ 0.25 \cdot$ MRR, with early stopping from Stage B. All runs use a single NVIDIA RTX 5060 Ti (16 GB); during end-to-end evaluation, CLaRa occupies most VRAM, so the GraphCLaRa retriever is placed on CPU. Table 2 collects the full hyperparameter setting.

Hyperparameter	Value
GAT layers / heads	2 (+1 refinement) / 4
GAT hidden dims	$1024 \to 256 \to 128$
Edge threshold τ	0.3
InfoNCE temperature T	0.1
Hard negatives κ	3
Margins m_{neg} / m_{path}	0.5 / 0.2
Loss weights $(\lambda_{\text{neg}}, \lambda_{\text{path}}, \lambda_{\text{reg}}, \lambda_K)$	$(0.5, 0.2, 0.01, 0.2)$
Learning rate / weight decay	10^{-4} / 0.01
Budget set $\mathcal{K}$	$\{5, 10, 20\}$

Table 2: GraphCLaRa hyperparameters, collected from Sections 3–4.2.

4.3 Main Results

Table 3 contrasts the baseline (BL) and GraphCLaRa (GC) on each benchmark's full evaluation set. GraphCLaRa improves answer F1 on three of four, largest on the two hardest multi-hop sets (+6.6 on 2Wiki, +6.4 on MuSiQue, +2.5 on NQ); Exact Match moves in lockstep (+5.1, +6.8, +2.3), so the gains reflect more exactly-correct answers, not token overlap. These are precisely the datasets where the baseline leaves room to improve (Section 4.4.1).

The retrieval columns show where the gains originate: on MuSiQue and 2Wiki GraphCLaRa lifts MRR (+0.19, +0.11) and pushes R@5 near ceiling (0.953, 0.997), surfacing gold documents the baseline had buried. HotpotQA is the lone exception — its baseline ranking is already saturated (R@5= 0.988, MRR= 0.904), so there is little to recover and GC trails slightly (−1.1 F1); it is also the only dataset where GC's MRR (0.865) falls *below* the baseline's.

4.4 Analysis

4.4.1 Where reranking helps: a headroom analysis

To explain the per-dataset pattern of Table 3, Table 4 decomposes the baseline by retrieval success: *retALL* is how often *all* gold documents already lie in its top-5, and F1 | all / F1 | miss split its F1 by that event. Together these reveal both *how much* retrieval headroom each dataset offers a reranker and *how much* that headroom matters for the answer.

		F1 (%)			EM (%)		R@5		MRR	
Dataset	N	BL	GC	Δ	BL	GC	BL	GC	BL	GC
MuSiQue	2,417	25.9	**32.4**	+6.4	14.9	**21.7**	0.844	**0.953**	0.606	**0.800**
2Wiki	12,576	42.7	**49.4**	+6.6	36.3	**41.4**	0.958	**0.997**	0.807	**0.920**
NQ	6,489	56.6	**59.1**	+2.5	46.0	**48.3**	0.892	**0.932**	0.698	**0.781**
HotpotQA	7,384	**69.5**	68.5	−1.1	**56.8**	55.5	**0.988**	0.983	**0.904**	0.865

Table 3: Main results: CLaRa baseline (BL, full 20-document context) vs. GraphCLaRa (GC, adaptive K). Best per metric in **bold**; Δ uses unrounded F1.

	Retrieval headroom			Budget $K_\star$ (%)		
Dataset	retALL	F1|all	F1|miss	5	10	20
MuSiQue	15.2	0.469	0.222	97.8	0.1	2.1
2Wiki	39.5	0.550	0.347	88.0	12.0	0.0
NQ	26.4	0.553	0.570	36.8	3.7	59.6
HotpotQA	77.0	0.744	0.534	99.3	0.3	0.3

Table 4: Baseline retrieval headroom and GraphCLaRa predicted-budget distribution. *retALL* is the fraction of examples whose gold set lies fully in the baseline top-5; F1|all/miss splits baseline F1 by that event.

Two conditions must hold for a reranker before a frozen reader to help. First, *retrieval headroom*: the baseline must miss the full gold set often enough to leave something to fix — on MuSiQue and 2Wiki it retrieves all gold documents only 15.2% and 39.5% of the time, headroom GraphCLaRa converts into R@5 of 0.953 and 0.997. Second, retrieval must *matter*, quantified by the gap F1 | all − F1 | miss, large on MuSiQue (+0.247) and 2Wiki (+0.203): when retrieval fails there the answer collapses, so closing the gap translates directly into answer gains. Both conditions hold on exactly the two datasets where GraphCLaRa wins widest.

On the other two datasets one condition fails. HotpotQA is retrieval-*saturated* (retALL = 77.0% with a full-context baseline): the gold pair is almost always present, so there is little to recover and compression can only remove evidence, while the head's collapse to K=5 on 99.3% of instances (Table 4) compounds the loss: that top-5 keeps both golds only 74.0% of the time (Table 5b), and the weighted change $0.74 \times (+0.044) + 0.26 \times (-0.167) \approx -0.011$ reproduces the observed −1.1 F1, so compression helps only when it preserves the full bridge. NQ shows the opposite failure: retrieval is *irrelevant* to the answer, since F1 | all (0.553) and F1 | miss (0.570) are indistinguishable. GraphCLaRa's NQ gain therefore cannot come from reranking; Section 4.4.2 traces it to the budget head.

4.4.2 Decomposing the adaptive budget

The right block of Table 4 shows that the budget head usually selects K=5 on the multi-hop sets: 88.0% on 2Wiki, 97.8% on MuSiQue, and 99.3% on HotpotQA. This compression is useful when the graph top-5 retains the evidence, as on MuSiQue and 2Wiki, but risky when the baseline is already saturated, as on HotpotQA. NQ is different: the head predicts K=20 on 59.6% of instances, often deferring to the full-context baseline.

Table 5a decomposes the NQ gain by predicted budget. The K=20 bucket is identical to the baseline and contributes no F1 change; nearly all of the +2.5 F1 comes from the K=5 bucket, where a shorter, reranked context appears to remove distractors rather than recover missing gold evidence.

(a) NQ gain by predicted budget $K_\star$					
$K_\star$	Inst. (%)	BL F1	GC F1	ΔF1	Contrib.
5	36.8	0.548	0.615	+0.067	+0.025
10	3.7	0.652	0.673	+0.021	+0.001
20	59.6	0.571	0.571	+0.000	+0.000

(b) HotpotQA by GC top-5 bridge retention			
Subset	Inst. (%)	BL F1	GC F1
Keeps bridge	74.0	0.713	**0.757**
Drops ≥1 gold	26.0	**0.646**	0.479
Overall	100.0	**0.695**	0.685

Table 5: (a) NQ gain by predicted budget; "contrib." is share-weighted ΔF1. (b) HotpotQA split by whether GC keeps the full gold bridge: compression helps when it survives but hurts sharply when a gold is dropped.

4.4.3 Retrieval metrics decouple from answer quality

A higher MRR does not translate into higher answer F1 at a fixed rate, because a multi-hop answer needs the *entire* bridge in context: promoting one gold raises MRR without helping if its partner stays outside the budget. This is why we treat end-to-end F1 as primary and report coverage@5/all@5 on MuSiQue, where GraphCLaRa lifts coverage@5 from 0.49 to 0.71 and all@5 from 0.15 to 0.40 — the gains come from recovering complete chains, not single entry points.

4.5 Ablation: role of the graph attention network

To isolate the contribution of graph propagation, we retrain a −GAT variant whose ranking uses only the content head (the GAT, its gate/residual heads, and the graph-dependent losses are removed); all other settings are identical and the frozen BGE cache is shared, so the two models differ only in the graph pathway. Table 6 reports the effect on the full evaluation set (retrieval-only, so no CLaRa call is needed).

		Full	−GAT
Ranking	MuSiQue all@5	**0.40**	0.36
	MuSiQue R@5	**0.953**	0.937
	2Wiki P@5	**0.446**	0.430
	NQ R@5 (single-hop)	0.932	**0.951**
Budget	NQ share of K=20 (%)	**59.6**	0.0

Table 6: Ablating the GAT (ranking uses the content head alone). Graph propagation helps precisely the multi-hop coverage metrics and is unnecessary on single-hop NQ; it is also required for the adaptive budget head, which collapses to a constant K=5 without graph features.

Removing the GAT degrades exactly the multi-hop coverage the reader depends on: MuSiQue all@5 drops 0.40 → 0.36 and 2Wiki P@5 drops 0.446 → 0.430, whereas single-hop NQ is unaffected — indeed marginally better without graph propagation (R@5 0.932 → 0.951), consistent with the headroom analysis that graph structure helps only where multi-hop evidence must be assembled. The GAT is also what makes the *adaptive* budget possible: its pooled features feed the budget head, and without them the head degenerates to a constant, predicting K=5 for every instance and never abstaining to the safety-equivalent K=20 — the NQ K=20 share collapses from 59.6% to 0%, removing the mechanism behind the NQ gain (Section 4.4.2). The path-supervision and refinement terms instead shape the graph for downstream reasoning rather than first-stage ranking, so their effect is not visible in retrieval-only metrics and we leave their end-to-end evaluation to future work.

5 Limitations

The same analysis pinpoints three limitations.

Budget-head collapse. The hard target-weighted proxy is dominated by $K{=}5$ labels once the ranker warms up, so the head collapses to $K{=}5$ on the multi-hop sets (99.3% on HotpotQA; Table 4), removing the full-context fallback exactly where it is needed and producing the -1.1 HotpotQA regression.

Proxy target is not downstream F1. The budget head is supervised by a retrieval surrogate — the smallest K covering the gold set — which keeps training cheap (CLaRa is never invoked) but blind to cases where a larger K would answer better.

Single baseline, no isolated budget ablation. The comparison is only against the full-context CLaRa reader; a BL@K control and stronger baselines such as BM25 and cross-encoders are left for future work.

6 Conclusion

We presented GraphCLaRa, a lightweight ($\sim$1.3M-parameter) graph-based reranker with an adaptive context-budget head that augments a frozen CLaRa-7B-E2E reader. Casting the candidate pool as a query-conditioned weighted graph and propagating evidence with a multi-head GAT, it improves end-to-end answer F1 on three of four benchmarks ($+6.6$ on 2Wiki, $+6.4$ on MuSiQue, $+2.5$ on NQ) and stays within 1.1 F1 on the retrieval-saturated HotpotQA, all with the reader frozen — showing that graph-based evidence propagation and adaptive selection complement latent-memory compression, especially when answering depends on recovering complete evidence chains. Natural next steps are downstream-aware budget objectives, richer budget features, and query decomposition that creates retrieval headroom.

Declarations

Funding

This research received no funding.

Data availability

The datasets analysed during the current study are available in the `apple/CLaRa_multi_stage` repository, https://huggingface.co/datasets/apple/CLaRa_multi_stage. [39] This public release provides the Natural Questions, HotpotQA, 2WikiMultiHopQA, and MuSiQue data used in the gold-annotated `with_pos` setting. The original benchmark resources are also publicly available at https://ai.google.com/research/NaturalQuestions/, https://hotpotqa.github.io/, https://github.com/Alab-NII/2wikimultihop, and https://github.com/StonyBrookNLP/musique, respectively. [40, 41, 42, 43]

Ethical Approval and accordance

Not applicable. This study is a secondary computational analysis of publicly available question answering benchmarks. Human annotation was performed during the original benchmark creation, not as part of this study. No new participants were recruited and no new human annotation was commissioned for this study.

Consent to participate

Not applicable. This study reused existing public benchmark data and did not recruit human participants.

Consent to publish

Not applicable. This study does not report newly collected participant data or identifiable information about the original benchmark annotators.

Clinical trial number

Not applicable.

References

[1] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A

dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380. Association for Computational Linguistics, 2018. URL https://doi.org/10.18653/v1/D18-1259.

[2] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. URL https://doi.org/10.1162/tacl_a_00475.

[3] Patrick Lewis, Ethan Perez, Aleksandra Piktus, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33*, pages 9459–9474. Curran Associates, Inc., 2020.

[4] Jie He, Richard He Bai, Sinead Williamson, Jeff Z. Pan, Navdeep Jaitly, and Yizhe Zhang. CLaRa: Bridging retrieval and generation with continuous latent reasoning, 2025. URL https://doi.org/10.48550/arXiv.2511.18659.

[5] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 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. Association for Computational Linguistics, 2020. URL https://doi.org/10.18653/v1/2020.emnlp-main.550.

[6] Wenhan Xiong, Xiang Lorraine Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Wen-tau Yih, Sebastian Riedel, Douwe Kiela, and Barlas Oğuz. Answering complex open-domain questions with multi-hop dense retrieval. In *International Conference on Learning Representations*. OpenReview, 2021. URL https://doi.org/10.48550/arXiv.2009.12756.

[7] Yuwei Fang, Siqi Sun, Zhe Gan, Rohit Pillai, Shuohang Wang, and Jingjing Liu. Hierarchical graph network for multi-hop question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8823–8837. Association for Computational Linguistics, 2020. URL https://doi.org/10.18653/v1/2020.emnlp-main.710.

[8] Nicola De Cao, Wilker Aziz, and Ivan Titov. Question answering by reasoning across documents with graph convolutional networks. In *Proceedings of NAACL-HLT 2019*, pages 2306–2317. Association for Computational Linguistics, 2019. URL https://doi.org/10.18653/v1/N19-1230.

[9] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*. OpenReview, 2018. URL https://doi.org/10.48550/arXiv.1710.10903.

[10] Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. Learning to retrieve reasoning paths over wikipedia graph for question answering. In *International Conference on Learning Representations*. OpenReview, 2020.

[11] Peng Qi, Xiaowen Lin, Leo Desde, Daniel Khashabi, Hannaneh Hajishirzi, and Christo-

pher D. Manning. Answering complex open-domain questions through iterative query generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 2590–2602. Association for Computational Linguistics, 2019. URL https://doi.org/10.18653/v1/D19-1261.

[12] Peng Qi, Haejun Lee, Tg Sido, and Christopher Manning. Answering open-domain questions of varying reasoning steps from text. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3599–3614. Association for Computational Linguistics, 2021. URL https://doi.org/10.18653/v1/2021.emnlp-main.292.

[13] Tianyang Zhang et al. Beam retrieval: Efficient retrieval for multi-hop question answering. In *Advances in Neural Information Processing Systems 37*. Curran Associates, 2024.

[14] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625. International Committee on Computational Linguistics, 2020. URL https://doi.org/10.18653/v1/2020.coling-main.580.

[15] Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302, 2018. URL https://doi.org/10.1162/tacl_a_00021.

[16] Ming Tu, Kevin Huang, Guang Chen, et al. Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9073–9080. AAAI Press, 2020. URL https://doi.org/10.1609/aaai.v34i05.6441.

[17] Yu Wang, Nedim Lipka, Ryan A. Rossi, Alexa Siu, Ruiyi Zhang, and Tong Sun Soliman. Knowledge graph prompting for multi-document question answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 15206–15222. Association for Computational Linguistics, 2024. URL https://doi.org/10.18653/v1/2024.acl-long.826.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.

[19] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*. OpenReview, 2017. URL https://doi.org/10.48550/arXiv.1609.02907.

[20] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 3982–3992. Association for Computational Linguistics, 2019. URL https://doi.org/10.18653/v1/D19-1410.

[21] Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in*

Natural Language Processing, pages 6894–6910. Association for Computational Linguistics, 2021. URL https://doi.org/10.18653/v1/2021.emnlp-main.552.

[22] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT 2019*, pages 4171–4186. Association for Computational Linguistics, 2019. URL https://doi.org/10.18653/v1/N19-1423.

[23] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2018. URL https://doi.org/10.48550/arXiv.1807.03748.

[24] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. REALM: Retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, pages 3929–3938. PMLR, 2020.

[25] Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, pages 874–880. Association for Computational Linguistics, 2021. URL https://doi.org/10.18653/v1/2021.eacl-main.74.

[26] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Few-shot learning with retrieval augmented language models. In *Advances in Neural Information Processing Systems 35*, pages 25630–25641. Curran Associates, 2022.

[27] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Liang, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. REPLUG: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*, 2023.

[28] Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. *arXiv preprint arXiv:2310.06839*, 2023.

[29] Fangyuan Xu, Weijia Shi, and Eunsol Choi. RECOMP: Improving retrieval-augmented LMs with compression and selective augmentation. *arXiv preprint arXiv:2310.04408*, 2023.

[30] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations*. OpenReview, 2024.

[31] Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 2007–2022. Association for Computational Linguistics, 2024. URL https://doi.org/10.18653/v1/2024.naacl-long.113.

[32] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. BGE embedding: General embedding models for retrieval and reranking. *arXiv preprint arXiv:2402.03216*, 2024.

[33] Alexis Conneau, Douwe Kiela, Holc Schwenk, Loic Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680. Association for Computational Linguistics, 2017. URL https://doi.org/10.18653/v1/D17-1070.

[34] Thomas Bachlechner, Bodhisattwa Prasad Majumder, Huanru Henry Mao, Garrison W. Cottrell, and Julian McAuley. ReZero is all you need: Fast convergence at large depth. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2021.

[35] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, 2020.

[36] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[37] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

[38] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. URL https://doi.org/10.1162/tacl_a_00276.

[39] Apple. CLaRa multi stage [data set]. Hugging Face, 2025. URL https://huggingface.co/datasets/apple/CLaRa_multi_stage.

[40] Tom Kwiatkowski et al. Natural Questions [data set]. Google Research, 2019. URL https://ai.google.com/research/NaturalQuestions/.

[41] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA [data set]. HotpotQA project, 2018. URL https://hotpotqa.github.io/.

[42] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2WikiMultiHopQA [data set]. Alab-NII, GitHub, 2020. URL https://github.com/Alab-NII/2wikimultihop.

[43] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue [data set]. StonyBrookNLP, GitHub, 2022. URL https://github.com/StonyBrookNLP/musique.