Failure Modes in Parametric Graph Retrieval for Multi-Hop Question Answering

Anonymous Authors[1]

Abstract

Graph-based retrieval improves multi-hop question answering by exposing evidence paths across documents, but explicit graph traversal can introduce additional inference-time cost and failure modes. We study whether the routing behavior of a BFS-based graph retriever can be approximated by a compact MLP reranker. On a 400-question HotpotQA distractor val set, the MLP-pruned system remains competitive with full BFS, slightly improving F1, supporting-fact recall, bridge-node recovery, and latency, while giving up a small amount of exact match. We also report a 100-question pilot ablation used to study candidate caps, latency, hub penalties, and context compression. Across these runs, we identify hub overload as a sharp residual failure mode: when high-centrality distractor nodes dominate the retrieved context, performance collapses. A simple hub-penalty mitigation reduces hub overload but also hurts aggregate retrieval quality, showing that the failure cannot be repaired by naive post-hoc score adjustment alone.

1. Introduction

Graph-based retrieval has emerged as a powerful approach for multi-hop question answering, where answering a question requires aggregating evidence from multiple documents connected through intermediate entities (Yang et al., 2018).

Methods ranging from flat dense retrieval (Lewis et al., 2020) to hybrid graph-augmented systems (Sun et al., 2018) and full breadth-first graph traversal occupy a spectrum of increasing structural complexity and computational cost. Yet despite growing adoption, a systematic understanding of where these approaches break down, and under what conditions, remains underdeveloped. Prior work has catalogued failure points in retrieval-augmented generation at a broad level (Barnett et al., 2024) (Jiao et al., 2025), but have not pinpointed operationally grounded failure analyses that isolate specific graph-search pathologies. Full breadth-first traversal in particular over a large graph must be repeated for every query, with latency scaling with graph size and edge density.

MLP Memory (Wei et al., 2025) showed that this retrieval behaviour can be approximated by a lightweight parametric module: an MLP pretrained to imitate the kNN distribution eliminates the datastore lookup entirely while preserving much of the accuracy gain. Their central finding is that *if retrieval similarity patterns can be learned parametrically, explicit retrieval becomes optional at inference*.

We ask whether the same substitution is possible in the graph-memory setting. Specifically: if kNN retrieval can be distilled into an MLP, can graph-memory routing - the decisions about which nodes and edges constitute a relevant evidence subgraph for a given query - also be distilled into a smaller learned module?

We instrument three systems at increasing levels of graph awareness — a flat embedding retriever, a hybrid embedding-plus-graph system, and an unconstrained BFS traversal — and use HotpotQA's distractor setting (Yang et al., 2018) as a controlled environment in which retrieval failures have direct, measurable consequences for answer quality. The distractor setting is well-suited to this purpose: ten candidate paragraphs are provided per question, the correct evidence chains are known, and supporting-fact recall provides a precise signal for retrieval failure independently of downstream answer generation.

We investigate this hypothesis on HotpotQA (Yang et al., 2018), a multi-hop QA benchmark designed to require evidence chains across multiple Wikipedia articles. We build a heterogeneous document-entity-sentence graph and train an MLP reranker to approximate the scoring behavior of a BFS-based teacher retriever. At inference, the MLP operates over a capped candidate pool produced by BFS-based pruning. We compare this system against flat embedding retrieval, a hybrid BFS+embedding baseline, and full BFS traversal.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

Our results show that the MLP-pruned retriever can preserve much of the graph-retrieval benefit while exposing a clear residual failure mode: when hub distractors dominate the retrieved context, performance collapses sharply. This makes hub overload a reproducible failure boundary for graph-memory retrieval systems.

2. Results and Analysis

Our diagnostic framework defines three operational failure modes: support collapse, where no supporting document enters the top-5 retrieved set; hub overload, where high-betweenness nodes dominate the retrieved set and displace query-specific evidence; and high-hop reasoning stress, where evidence lies at graph distance two or more from the query anchor. These definitions are inspired by recent observations that static graph traversal is susceptible to hub nodes acting as semantic sinks that divert probability mass away from task-relevant evidence (Yun et al., 2025) (Gu, 2025), and by broader diagnostics of retrieval-reasoning misalignment in multi-hop settings (Ramnath, 2025)

2.1. Experimental Setup

We build the graph, (see Fig. 1) and training artifacts from a 2000-question HotpotQA distractor slice, using 1600 questions for training and 400 questions for evaluation. We compare four retrieval systems: a flat embedding baseline (`flat`), a hybrid BFS+embedding system (`hybrid`), a full BFS graph traversal baseline (`full_bfs`), and our MLP-reranked pruned system (`mlp_pruned`) [see Fig. 2]. We report exact match (EM), token-level F1, supporting-fact recall (SF recall), bridge-node recovery, hub-distractor rate, and mean per-question latency in milliseconds. We use the 400-question val set for the main evaluation and report the earlier 100-question run only as a pilot ablation for optimization and failure analysis.

2.2. Main Results

Graph construction. The rebuilt graph contains 217,090 nodes and 695,721 edges, with an average degree of 6.41. The graph contains question, document, sentence, entity, and value nodes, connected by typed edges such as `has_sentence`, `mentions`, `same_entity`, `bridge_to`, `compared_with`, and temporal links. Only 4 of the 2000 questions are unreachable from the constructed graph, suggesting that the graph build covers nearly all examples in the slice.

MLP training data. Feature generation produces 16,304 candidate rows with 781-dimensional features. Each question contributes between 2 and 20 candidate rows, with a mean of 8.2 rows per question and a positive rate of 0.251. The question split contains 1600 training questions and 400 validation questions. During training, the best validation loss is 0.2543 and the best validation supporting-fact recall@5 is 0.855.

On the 400-question val set, `full_bfs` obtains the highest EM, while `mlp_pruned` obtains the best F1 and supporting-fact recall. Compared with `full_bfs`, `mlp_pruned` improves F1 from 0.322 to 0.324, SF recall from 0.532 to 0.540, bridge-in-top5 from 0.107 to 0.138, and latency from 28.36 ms to 24.31 ms, while Exact Match (EM) decreases slightly from 0.247 to 0.242.

Table 1. Overall results on the 400-question HotpotQA val slice.

System	EM	F1	SF Rec.	Lat. (ms)
`flat`	0.190	0.264	0.329	6.26
`hybrid`	0.212	0.290	0.321	25.45
`full_bfs`	**0.247**	0.322	0.532	28.36
`mlp_pruned`	0.242	**0.324**	**0.540**	**24.31**

Question type. On bridge questions, graph-aware methods substantially outperform non-graph retrieval. `mlp_pruned` reaches bridge SF recall of 0.552, compared with 0.338 for `flat` and 0.346 for `hybrid`. On comparison questions, `mlp_pruned` obtains the highest EM, while `full_bfs` obtains the highest SF recall. This again suggests that the MLP reranker improves ranking in some cases, but does not replace the need for graph connectivity.

Table 2. Results by question type on the 400-question val set.

	Bridge (n=327)		Comparison (n=73)	
System	EM	SF Recall	EM	SF Recall
`flat`	0.205	0.338	0.123	0.288
`hybrid`	0.229	0.346	0.137	0.212
`full_bfs`	**0.269**	0.538	0.151	**0.507**
`mlp_pruned`	0.260	**0.552**	**0.164**	0.486

Latency. The optimized `mlp_pruned` path remains bounded at the larger evaluation scale. On the 400-question val set, `mlp_pruned` runs at 24.31 ms per question, compared with 28.36 ms for `full_bfs`. This indicates that teacher-score pruning and capped reranking can reduce online retrieval cost while preserving the main graph-retrieval benefit.

Pilot ablation on the 100-question slice. Before running the larger 400-question evaluation, we used a 100-question val slice to test whether capped MLP reranking preserves retrieval quality. On this smaller pilot, `mlp_pruned` achieved the strongest EM, F1, and supporting-fact recall among the compared systems, reaching EM 0.270, F1 0.355, and SF recall 0.625. We treat this result as an optimization

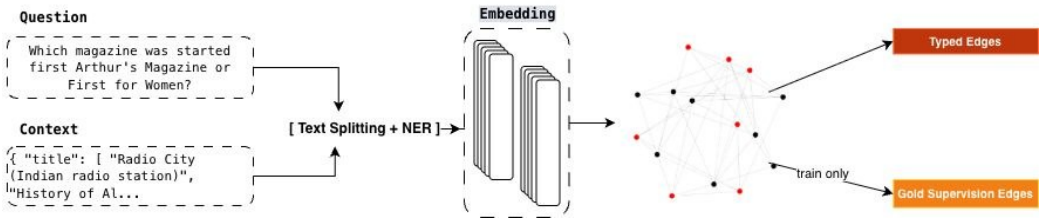

Figure 1. Offline graph construction

and ablation study rather than the primary evaluation, since the larger 400-question run gives a more stable estimate of aggregate performance.

Table 3. Pilot results on the earlier 100-question HotpotQA val slice. This table is used as an ablation reference, not as the main evaluation.

System	EM	F1	SF Rec.	Lat. (ms)
`flat`	0.170	0.246	0.340	2.04
`hybrid`	0.230	0.290	0.355	9.47
`full_bfs`	0.240	0.321	0.565	10.43
`mlp_pruned`	**0.270**	**0.355**	**0.625**	12.13

2.3. Failure Mode Analysis

The 100-question pilot and the 400-question val evaluation tell a consistent but not identical story. The pilot run suggests that capped MLP reranking can improve aggregate retrieval quality, while the larger run gives a more conservative estimate: `mlp_pruned` remains competitive with `full_bfs`, improves F1 and SF recall, but no longer wins EM. We therefore use the 400-question run as the main result and use the 100-question run to support ablation-level claims about latency, candidate caps, and hub penalties.

Aggregate metrics obscure sharp performance discontinuities that reveal the true operating boundaries of `mlp_pruned`. We define three operational failure modes and measure their prevalence and impact.

2.3.1. Failure Mode 1: Hub Overload

Definition. A retrieval is classified as hub-overloaded when at least three of the top-five retrieved items are hub distractors.

Observed impact. Hub overload is the clearest residual failure mode for `mlp_pruned`. On the hub-overload slice, `mlp_pruned` drops to EM 0.119 and SF recall 0.321. In contrast, on the clean slice with zero hub distractors, it reaches EM 0.336 and SF recall 0.617. This gap suggests that the method is not failing uniformly; rather, performance collapses under a specific graph-topology condition.

Mitigation attempt. We tested a simple hub-penalty mitigation that subtracts score mass from high-hub nodes during reranking. Although this reduced hub-overload frequency, it also reduced aggregate EM and supporting-fact recall. This suggests that hub nodes are not merely irrelevant noise: some carry useful bridge information, so suppressing them post-hoc creates a trade-off between reducing distractors and preserving evidence reachability.

Table 4. Hub overload vs. clean slice performance for `mlp_pruned`.

Condition	n	EM	SF Recall
Clean ($=0$ hub distractors)	107	0.336	0.617
Hub overload (≥ 3 hub distractors)	42	0.119	0.321

On the 100-question pilot slice, the same trade-off appears clearly. Increasing the hub penalty sharply reduces hub rate, but it also degrades supporting-fact recall. For example, increasing the penalty from 0 to 0.25 reduces hub rate from 0.404 to 0.182, but lowers SF recall from 0.625 to 0.605. Stronger penalties reduce hub rate further, but collapse SF recall. This makes the mitigation result useful as a negative finding: hub overload is real, but naive post-hoc suppression is too blunt.

Table 5. Hub-penalty ablation on the 100-question pilot slice. K denotes the MLP candidate cap and λ denotes the hub-penalty coefficient.

Variant	EM	SF Recall	Latency (ms)	Hub Rate
K=512, λ=0	**0.270**	**0.625**	9.27	0.404
K=256, λ=0	0.260	**0.625**	7.43	0.396
K=512, λ=0.25	0.260	0.605	9.75	0.182
K=512, λ=0.5	0.240	0.495	9.80	0.016
K=512, λ=1.0	0.210	0.330	9.08	0.002

2.3.2. Failure Mode 2: Support Collapse

Definition. A retrieval is classified as support-collapsed when the retrieved top-five contains zero supporting documents.

Observed impact. Support collapse is much more fre-

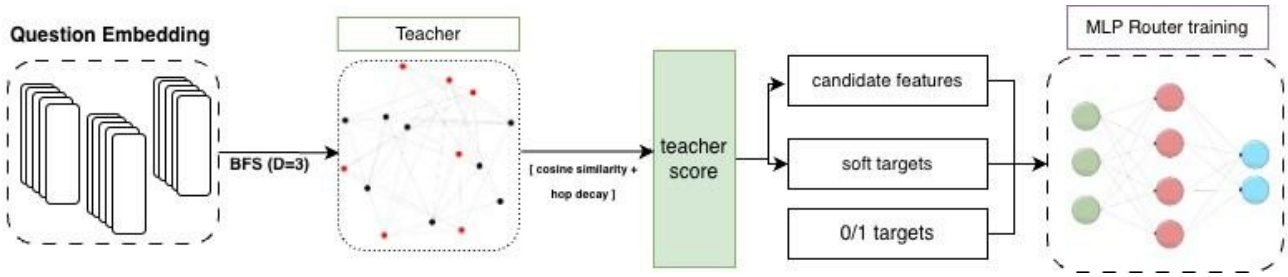

Figure 2. Teacher to router distillation

quent for non-graph retrieval than for graph-aware retrieval. It occurs in 153 questions for `flat` and 162 questions for `hybrid`, but only 62 questions for `full_bfs` and 60 questions for `mlp_pruned`. This supports the central role of graph connectivity: the main improvement comes from reaching the evidence subgraph, while the MLP reranker provides a smaller additional reduction in collapse frequency.

2.3.3. Failure Mode 3: High-Hop Reasoning Stress

Definition. A question is classified as high-hop when its supporting facts lie at graph hop distance ≥ 2 from the query node.

Observed impact. On the high-hop slice (n=179), both graph-aware methods outperform the non-graph baselines in supporting-fact recall. `flat` reaches SF recall of 0.313 and `hybrid` reaches 0.274, while `full_bfs` reaches 0.441 and `mlp_pruned` reaches 0.430. However, `mlp_pruned` does not improve supporting-fact recall over `full_bfs` on this slice, suggesting that the reranker mainly improves ranking within reachable candidates rather than adding deeper graph reasoning.

Table 6. High-hop slice performance, where supporting facts are at hop distance ≥ 2.

System	n	EM	F1	SF Recall
`flat`	179	0.151	0.232	0.313
`hybrid`	179	0.173	0.263	0.274
`full_bfs`	179	0.156	0.238	**0.441**
`mlp_pruned`	179	**0.168**	**0.253**	0.430

This pattern supports a limited interpretation of the MLP reranker. It is useful as a precision filter over BFS-generated candidates, but it does not reliably improve deeper-hop evidence recovery beyond what the graph traversal already provides.

2.4. Context Compression Pilot

We also tested whether the retrieved context can be compressed before being passed to the QA reader. This experiment was run on the 100-question pilot slice. Replacing full documents with the top-3 query-matched sentences slightly improves EM from 0.270 to 0.280, but reduces F1 from 0.355 to 0.342. Adding document titles hurts both metrics. This suggests that sentence-level compression may reduce context size, but it does not yet dominate full-document retrieval across both answer-level metrics.

Table 7. Context compression ablation for `mlp_pruned` on the 100-question pilot slice.

Context Mode	EM	F1
Full documents	0.270	**0.355**
Top-3 sentences	**0.280**	0.342
Title + top-3	0.250	0.334
Title + top-5	0.250	0.327

Conclusion and Impact Statement

This work studies retrieval mechanisms for multi-hop question answering. The main expected impact is improved understanding of when graph-based retrieval systems succeed or fail. The work does not introduce new data collection procedures or user-facing deployment mechanisms. Potential risks are similar to those of retrieval-augmented QA systems more broadly, including the possibility of retrieving incomplete, misleading, or weakly grounded evidence. Our failure analysis aims to make such risks easier to diagnose by identifying reproducible retrieval failure boundaries.

References

Barnett, S. et al. Seven failure points when engineering a retrieval augmented generation system. *arXiv preprint arXiv:2401.05856*, 2024.

Gu, e. a. Breaking the static graph: Context-aware traversal

for robust retrieval-augmented generation. *arXiv preprint arXiv:2602.01965*, 2025.

Jiao, S. et al. Doctor-RAG: Failure-aware repair for agentic retrieval-augmented generation. *arXiv preprint arXiv:2604.00865*, 2025.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., and Kiela, D. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Ramnath, e. a. When iterative RAG beats ideal evidence: A diagnostic study in scientific multi-hop question answering. *arXiv preprint arXiv:2601.19827*, 2025.

Sun, H., Dhingra, B., Zaheer, M., Mazaitis, K., Salakhutdinov, R., and Cohen, W. W. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Wei, R. et al. MLP memory: A retriever-pretrained memory for large language models. *arXiv preprint arXiv:2508.01832*, 2025.

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W. W., Salakhutdinov, R., and Manning, C. D. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2369–2380, 2018.

Yun, J., Lee, D., and Han, W.-S. Failure is feedback: History-aware backtracking for agentic traversal in multimodal graphs. *arXiv preprint arXiv:2602.03432*, 2025.