PAGEINDEX-GUIDED MULTIMODAL GRAPHRAG: A CRITICAL EVALUATION OF STRUCTURE-AWARE RETRIEVAL ON VISUALLY RICH FINANCIAL DOCUMENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Financial annual reports pack critical information into tables and charts, yet most retrieval-augmented generation (RAG) systems, graph based or not, only read the surrounding text. We evaluate whether adding a knowledge graph and multimodal (text plus vision) processing improves retrieval on real, visually rich documents, or whether it primarily adds complexity. We build **PIM-GraphRAG**, a system combining page-level hierarchical navigation with a schema-constrained multimodal knowledge graph and training-free cross-modal entity linking, and evaluate it against four baselines on ten real UK annual reports. The proposed system underperforms simple BM25 keyword search by 35 points on document accuracy (58.82% vs. 94.12%), and cross-modal linking recovers only 16 of 2,023 candidate entity pairs. A companion ablation across text-only, multimodal, and graph-augmented baselines shows why an unconstrained relation schema collapses distinct facts into one generic label, driving retrieval precision toward zero. We present these negative results as the paper's primary contribution, since they identify concretely where multimodal graph augmentation helps and where it does not, and we release the full pipeline for reproducibility.

1 INTRODUCTION

Large Language Models (LLMs) write fluently but only know what they saw during training, so they cannot reliably answer questions about a specific external document (3). Retrieval-Augmented Generation (RAG) fixes this by pulling relevant content from an outside corpus and feeding it to the model before generation. It has a well-known blind spot, though: standard RAG chops a document into a flat sequence of independently embedded chunks. Page structure disappears, and anything that is not plain text, a chart, a table, a diagram, is simply invisible to the pipeline.

Two separate lines of work have tried to patch different parts of this problem. Graph-based methods build an explicit relational structure over entities (2; 6), while hierarchical indexing schemes try to preserve how a document is actually organised across pages and sections (5). They rarely meet. GraphRAG-style methods operate almost exclusively on text and do not preserve page-level co-location, so a chart and its caption can end up nowhere near each other in the index. PageIndex keeps that co-location intact but has no relational graph at all, which limits reasoning that spans multiple pages. Systems that do try to bridge text and vision in a single graph, such as MMGraphRAG (1), need task-specific training to align entities across modalities. That training cost is exactly what makes such systems hard to point at a new document type without redoing the work.

Research Gap. No existing system does all three things at once: covers text, tables, and images; keeps page-level structure intact; and works without fine-tuning on the target domain (Table 1). This is not just an academic gap. A financial analyst asking about the year-on-year change in operating profit for one division might need a number sitting in a table, a chart, a paragraph, or some combination of the three. Keyword and vector search both miss anything locked inside a visual element, full stop, and manually flipping through a hundred-page PDF does not scale to dozens of reports. A

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

system that handles this reliably has direct value in financial services, audit, and compliance work, and that practical need is what motivates this study.

Approach Used. PIM-GraphRAG combines PageIndex-style hierarchical indexing with a schema-constrained multimodal knowledge graph, using GPT-4o Vision for visual extraction and Neo4j for storage. Its core piece is a training-free cross-modal entity alignment step: it spots when the same real-world entity, say "Operating Profit", shows up as running text, a chart axis label, and a table cell, and merges all three into one graph node using embedding similarity constrained by page proximity in the PageIndex tree. Section 4 shows this combination does not beat simpler baselines, and that is the point worth dwelling on rather than burying. Most papers in this space ask whether a multimodal graph-augmented system can be built. Fewer ask whether building one is worth it (8). This paper tries to answer the second question with numbers instead of intuition.

Four contributions come out of this work:

1. An end-to-end pipeline combining PageIndex hierarchical indexing, schema-constrained multimodal knowledge graph construction, and training-free cross-modal entity alignment, evaluated on ten real financial annual reports.
2. A controlled four-way ablation (vector RAG, text-only GraphRAG, multimodal RAG without a graph, and the full proposed system) that isolates what the graph adds versus what multimodality adds.
3. Two concrete failure modes, examined in detail: generic-relation schema collapse in unconstrained GraphRAG extraction, and how sparse recoverable cross-modal entity links actually are in real chart-heavy documents.
4. A verified 17-question evaluation set with document- and page-level ground truth, showing summary-based hierarchical navigation loses to direct lexical and dense retrieval on narrowly factual queries, and pinning down when that navigation style is likely to help instead.

2 Related Work

2.1 Standard and Graph-Based RAG

RAG (3) retrieves the top-k relevant chunks from a preprocessed corpus and conditions generation on the retrieved content plus the query. It grounds answers better than relying on parametric memory alone, but the corpus is still just unstructured text underneath, so inter-document reasoning and multi-hop inference stay weak (6). GraphRAG methods try to fix that by building a knowledge graph over the document content instead. Microsoft's implementation (2) extracts entities and relations with an LLM, clusters them with the Leiden community-detection algorithm, and supports both local fact-level search and global thematic search. The design is genuinely good at multi-hop reasoning through relational paths. It is also entirely text-only, ignores page-level document structure, and, as Section 4 shows, its usefulness depends heavily on how carefully the relation schema is designed.

2.2 Hierarchical and Chunk-Centric Indexing

RAPTOR-style methods take a different route: cluster text chunks into a hierarchy where leaves hold the original text and parent nodes hold progressively abstract summaries. Contextual completeness survives across levels of granularity this way, but explicit relationships between entities in different chunks are still lost. PageIndex (5) skips chunking-by-similarity entirely and instead builds a tree that mirrors the document's own page and section structure, so a table and the paragraph explaining it stay co-located because they were never separated in the first place. An LLM walks this tree at query time rather than running a brute-force vector search over everything. No vector database, no fine-tuning, which is appealing, but nothing here extracts entities or relations into a graph, so cross-page relational reasoning is still out of reach.

2.3 Multimodal Document Understanding and Multimodal GraphRAG

DocVQA (4) was one of the first benchmarks to show, concretely, that text-only systems fail once tables and charts enter the picture. Surveys since then have converged on multimodal knowledge

graphs, unified structures holding both text and visual entities, as the most promising direction, and also the hardest to build well (10). RAG-Anything (9) handles text, images, tables, equations, and code through a dual-graph setup, but the text graph and the visual graph stay structurally separate; cross-modal retrieval happens at query time rather than being baked into how the graph itself is built. MegaRAG adds visual cues to GraphRAG but drops page-level structure and never proposes an explicit way to align entities across modalities. The closest match to this work is MMGraphRAG (1), which builds visual scene graphs and links them to text graphs through SpecLink, a spectral-clustering alignment method. It performs well on benchmarks. It also needs task-specific training and, like the others, does not use page-level structure. Those two gaps, the training requirement and the missing structural awareness, are exactly what this system targets.

Table 1: Capability comparison of existing document retrieval systems.

System	Text	Tables	Images	Page-Level Indexing	Training-Free
Standard RAG (3)	Yes	No	No	No	Yes
Microsoft GraphRAG (2)	Yes	No	No	No	Yes
MegaRAG	Yes	No	Yes	No	No
MMGraphRAG (1)	Yes	No	Yes	No	No
RAG-Anything (9)	Yes	Yes	Yes	No	No
PageIndex (5)	Yes	No	No	Yes	Yes
This work	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**

3 METHOD

3.1 OVERVIEW

PIM-GraphRAG (PageIndex-guided Multimodal GraphRAG) runs in five stages, shown in Figure 1. Hierarchical page-level indexing comes first, then multimodal page assembly, then schema-constrained knowledge graph extraction, then training-free cross-modal entity alignment. Two-stage, PageIndex-guided answer generation comes last, and it is the only stage that runs online. Everything before it happens offline, during indexing. Figure 1 keeps these two phases visually separate for that reason, with a dashed arrow marking where the offline-built graph and tree feed into online retrieval.

3.2 HIERARCHICAL INDEX CONSTRUCTION

For a document corpus $D = \{d_1, \ldots, d_n\}$, each document d_i gets its own page-level navigation tree T_i, built with the PageIndex library from the table of contents where one exists, or from detected section boundaries where it does not. Every tree node spans a contiguous page range; the root is the whole document, and the leaves are individual pages or small groups of them. Node boundaries come from the document's own structure, not from clustering pages by similarity after the fact, which is the opposite of how RAPTOR builds its hierarchy.

3.3 MULTIMODAL PAGE ASSEMBLY

A subset of pages $P_i \subset d_i$ is sampled per document, weighted toward pages dense with tables and images, since those are exactly where a text-only pipeline breaks down. Each sampled page $p \in P_i$ becomes a triple:

$$p = \big(\text{text}(p),\ \text{tables}(p),\ \text{image}(p)\big) \tag{1}$$

Text comes from pdfplumber, tables from detected table structures, and the image is a 150-DPI render via PyMuPDF. Any page that either contains a detected table or yields fewer than 50 characters of extracted text gets flagged needs_vision$(p) = 1$ and routed to GPT-4o Vision instead of text-only extraction.

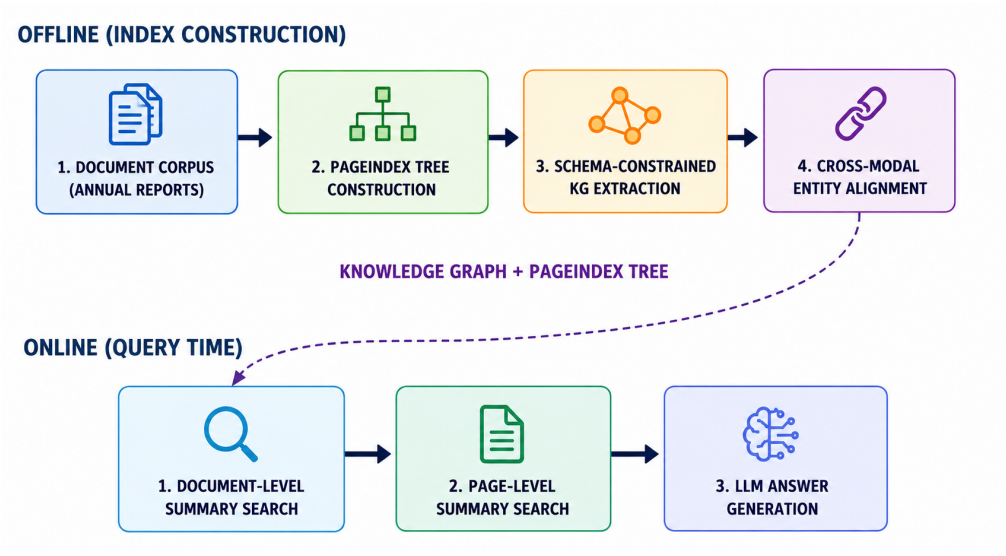

Figure 1: PIM-GraphRAG pipeline. Top row: offline index construction (document ingestion through cross-modal alignment). Bottom row: online query-time retrieval and answer generation. The dashed arrow indicates that the offline-built knowledge graph and PageIndex tree feed directly into online retrieval.

3.4 Schema-Constrained Knowledge Graph Construction

Nine entity types (Organisation, Person, FinancialMetric, MonetaryValue, Date, Location, Product, RiskFactor, BusinessSegment) and a matching relation type set $\mathcal{R}_{\text{types}}$ are fixed before any extraction happens. That decision was not arbitrary. It came directly out of a failure observed in the text-only GraphRAG baseline (Section 4.4), where letting the model invent its own relation labels collapsed almost everything into one dominant, near-useless label.

For each page p, an LLM, GPT-4o-mini normally, GPT-4o Vision when needs_vision$(p) = 1$, extracts a triple set:

$$T_p = \{(h, r, t) \mid h, t \in \mathcal{E}_p,\ r \in \mathcal{R}_{\text{types}}\} \tag{2}$$

constrained to that fixed schema. Every entity gets a modality tag, $m(e) \in \{\text{text}, \text{visual}\}$, recording whether it came from running text or from GPT-4o Vision reading the page image.

An early extraction pass turned up a specific problem: visual entities were often bare numbers, "24%", "2019", with no indication of what they referred to. Useless for retrieval. The vision prompt was rewritten to force every extracted value to carry its referent, "Operating Profit: 24%" instead of just "24%", and every vision-dependent page was reprocessed. The modality split moved from an initial 2,365 text / 43 visual entities to a corrected 1,752 / 595, and the corrected numbers are what the rest of this paper uses, since they reflect the actual text-to-visual entity density in the corpus far better than the first pass did.

3.5 Training-Free Cross-Modal Entity Alignment

Given the combined entity set $V = \bigcup_p \mathcal{E}_p$, the task is finding pairs (e_i, e_j) where $m(e_i) = \text{text}$ and $m(e_j) = \text{visual}$ but both describe the same real-world thing. OpenAI embeddings $\mathbf{z}_e$ are computed per entity name, and a pair becomes a candidate link when:

$$\text{sim}\big(\mathbf{z}_{e_i}, \mathbf{z}_{e_j}\big) > \tau \quad \text{and} \quad \big|\text{page}(e_i) - \text{page}(e_j)\big| \le w \tag{3}$$

with $\tau = 0.75$ as the similarity cutoff and w a page-proximity window taken from the PageIndex tree. A second filter strips out fiscal-year coincidences, "2024" matching "2023/24" by embedding similarity alone, which turned out to be a major source of false positives. What survives gets merged into a single graph node. None of this needs task-specific training, unlike SpecLink's spectral clustering (1), and nothing here is tied to financial documents specifically.

3.6 PageIndex-Guided Answer Generation

Retrieval at query time happens in two passes. First, a document-level search compares the query against short per-document summaries and keeps the top one or two matches:

$$D^* = \text{Top-}k\Big(\big\{\text{sim}(\mathbf{q}, \mathbf{s}_d) \;\big|\; d \in D\big\},\; k = 2\Big) \tag{4}$$

Then a page-level search inside those documents narrows things down to the most likely page range, using per-page summaries generated for every sampled page:

$$P^* = \text{Top-}k\Big(\big\{\text{sim}(\mathbf{q}, \mathbf{s}_p) \;\big|\; p \in \text{Pages}(D^*)\big\},\; k = 3\Big) \tag{5}$$

Whatever text, visual descriptions, and linked graph relations come out of that search get handed to the LLM for the final answer.

4 Experiments

Four questions drive the evaluation:

- **RQ1.** How does the proposed system perform relative to standard retrieval baselines (vector RAG, BM25, knowledge-graph-only) on verified, page-grounded financial QA?
- **RQ2.** How much does the design of the knowledge-graph relation schema affect retrieval quality, independent of whether a graph is present at all?
- **RQ3.** How effective is training-free, embedding-based cross-modal entity alignment on real financial documents with dense tabular and chart content?
- **RQ4.** Under what query conditions does summary-based hierarchical page navigation help or hurt, relative to direct lexical or dense search over page text?

4.1 Datasets and Metrics

DocVQA (4) serves as a secondary, motivational dataset here, publicly available at https://www.docvqa.org/. All 5,349 validation questions were sorted into four types by rule-based keyword matching: factual text extraction (87.8%), table reading (8.0%), chart reading (3.5%), and cross-modal reasoning (0.7%). Just over 12% require visual content to answer at all, which is a reasonable independent argument for a multimodal design before this paper's own results even come into it.

The primary dataset is ten UK FTSE-listed annual reports from 2024 and 2025, pulled directly as PDFs from each company's own investor-relations page: AstraZeneca, Barclays, BP, BT Group, HSBC, Marks & Spencer, National Grid, Rolls-Royce, Tesco, and Unilever. Together they run to 3,001 pages. All ten are publicly filed regulatory documents, so nothing proprietary, confidential, or personally identifiable enters this study anywhere. 68.3% of pages mix text and tables, 31.3% are text-only, and 0.4% are image-only. Visual density, the share of non-text-only pages, averages 67.9% and ranges from 34.1% in AstraZeneca's report up to 99.6% in Barclays's. Text extraction held up well: 99.0% of pages yielded more than 200 characters via pdfplumber.

Knowledge graph construction ran on a stratified sample of 150 pages, 15 per report, weighted toward table- and image-dense pages, rather than the full 3,001. The reason is money, not methodology: total OpenAI API spend across the whole project, covering the schema-constrained knowledge graph extraction, cross-modal alignment, and all baseline construction, came to approximately $50, and 150 pages was what this project could afford to process for the knowledge graph while still touching every report and both page types. Section 6 comes back to what this choice does and does not affect.

The baseline comparison set has 120 questions spanning all four DocVQA-style types, scored with RAGAS on faithfulness, answer relevancy, context precision, and context recall. The verified evaluation set is smaller and stricter: 17 questions, each built from a page with at least five extracted entities, each pointing at one specific entity or metric, each with a confirmed document and page as ground truth. It is scored on document accuracy, page precision at rank one (P@1), and page precision within the top-retrieved nodes (P@K).

4.2 BASELINES

Five baselines split across the two evaluation stages, and together they function as a component-wise ablation of PIM-GraphRAG rather than just a list of unrelated comparisons: each one removes exactly one of the proposed system's components, so the gap between adjacent rows in Tables 2 and 3 isolates what that component contributes. On the 120-question set: Standard RAG (512-token chunks, local embedding index, top-5 retrieval) removes both the graph and multimodality; Text GraphRAG (a Neo4j graph over the same chunks, but with unconstrained GPT-4o-mini relation extraction) adds an unconstrained graph back in, without multimodality or a fixed schema; Multimodal RAG without a graph (page text plus GPT-4o Vision descriptions, no graph layer at all) adds multimodality back in, without any graph. On the 17-question verified set: BM25 (pure keyword ranking) and Vector RAG (top-3 chunks by embedding similarity) both remove the graph, multimodality, and hierarchical navigation entirely; Knowledge-graph-only (direct entity and relation term matching, skipping both vector search and page-level navigation) isolates the schema-constrained graph on its own, without PageIndex navigation. Comparing each of these against the full PIM-GraphRAG result shows where the graph, the multimodality, and the hierarchical navigation each help or hurt in isolation.

4.3 MAIN RESULTS (RQ1)

Table 2: RAGAS scores by baseline system ($n = 120$).

System	Faithfulness	Answer Rel.	Context Prec.	Context Rec.
Standard RAG	0.784	0.872	0.591	0.534
Text GraphRAG	0.657	0.341	0.070	0.173
Multimodal RAG (no graph)	0.618	0.899	0.510	0.499

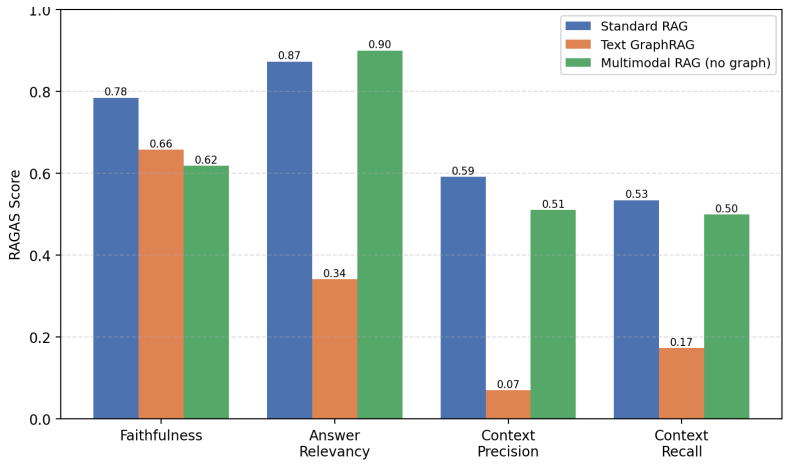

Figure 2: RAGAS scores across the three baseline systems, visualising the sharp collapse in Text GraphRAG's context precision and recall.

Table 3: Final system comparison on the verified 17-question evaluation set.

System	Doc. Accuracy (%)	Page P@1 (%)	Page P@Top-K (%)
PIM-GraphRAG (proposed)	58.82	23.53	29.41
Knowledge-graph-only	76.47	47.06	64.71
Vector RAG	88.24	64.71	76.47
BM25 keyword	94.12	70.59	88.24

Neither evaluation is kind to the proposed system. Standard RAG wins three of four RAGAS metrics outright, and on the verified set, all three simpler baselines beat PIM-GraphRAG on every single

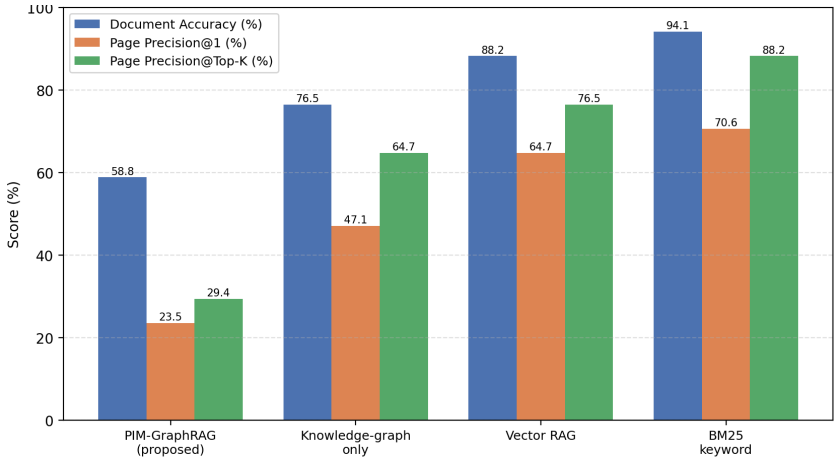

Figure 3: Document accuracy and page-level precision across all four systems on the verified evaluation set.

metric. Read as an ablation, adding an unconstrained graph (Text GraphRAG) makes things worse than Standard RAG on every metric, and adding the schema-constrained graph plus full navigation (PIM-GraphRAG) still underperforms simpler methods on the verified set. Sections 4.4 through 4.6 dig into why.

4.4 Why Unconstrained GraphRAG Collapses (RQ2)

Something specific went wrong with Text GraphRAG's extraction. Nearly every relation, regardless of whether it described a financial figure, a board appointment, or a sustainability partnership, ended up under one generic label: `reported_metric`. Retrieval was not empty; each query pulled back somewhere between 4 and 19 context items. The content of those items was the problem. A question asking which partnership a company uses for rainforest restoration came back as `Dove -> reported_metric -> product launch`, and the real answer, "Rimba Collective", sitting right there verbatim in the source text, never made it into a usable triple. That is why context precision and recall dropped to almost nothing (0.070 and 0.173), while answer relevancy held up a bit better (0.341): the model tended to admit it did not know rather than guess, and RAGAS rewards that honesty. Similar information loss during entity-centric extraction has been reported before, sometimes badly enough that GraphRAG loses to plain dense retrieval on factual QA. The schema-constrained design in Section 3.4 exists specifically because of this failure, and its nine fixed types kept the same collapse from happening in the proposed system's own graph.

4.5 Cross-Modal Linking Difficulty (RQ3)

The first linking pass, at a 0.75 similarity threshold, found nothing. Zero links. That traced straight back to the bare-numeric-label problem from Section 3.4. After the extraction prompt was fixed and every page reprocessed, a re-run across 2,023 valid entities (1,515 text, 508 visual) turned up only 6 candidate links. Filtering out fiscal-year coincidences left 16 verified links total, and 12 of those sit in a single report, Barclays, with 2 each in BP and National Grid and none anywhere else.

Why so few? Most numeric labels on a financial chart, an axis value, a percentage callout, do not have an independently named entity sitting nearby for an embedding model to latch onto. The chart's caption or title usually already absorbed whatever descriptive context existed during extraction, leaving the bare number with nothing left to match against. That looks like a structural property of financial-report charts specifically, not a bug in the linking code, and it has not been quantified before in work on embedding-based cross-modal alignment for chart-heavy enterprise documents.

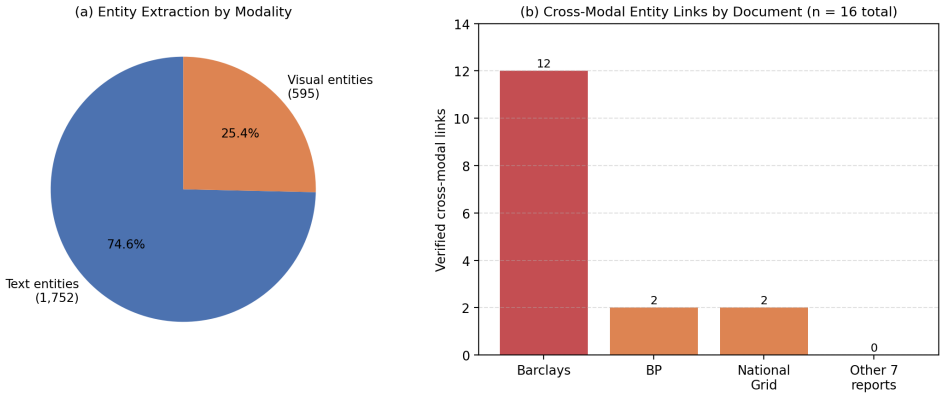

Figure 4: (a) Split between text and visual entities after prompt correction; (b) distribution of verified cross-modal links across the ten reports, showing extreme concentration in a single document.

4.6 Case Studies (RQ2, RQ4)

Case A: successful two-stage navigation. A query about a bank's risk-weighted asset (RWA) allocation to its Investment Bank division worked exactly as intended: document-level search found the right report, page-level search narrowed to the right page range, and the system answered "56% (p14)", which matches the source figure exactly. This is what two-stage summary-guided navigation looks like when the query matches how the page summaries were actually written.

Case B: schema collapse (Text GraphRAG).

Query	Which partnership does the company use to help restore rainforests?
Gold answer	Rimba Collective
Text GraphRAG output	`Dove -> reported_metric -> product launch` (answer not recovered)
Cause	Unconstrained relation schema collapsed a specific partnership fact into a generic catch-all label, discarding the entity needed to answer the question.

Case C: narrowly factual query favouring lexical search. Several of the 17 verified questions, a specific accounting policy statement, years of board tenure, were answered directly by BM25 and vector RAG simply because the question's own wording showed up verbatim on the source page. PIM-GraphRAG had no such shortcut. It worked from a compressed two-to-three-sentence summary at each of its two navigation stages, and that summary sometimes dropped the one figure or phrase the question actually needed, even though the full knowledge graph for that page still had the fact sitting in it correctly. This is not a defect so much as a trade-off: summary-based navigation gives up some single-fact precision in exchange for scalability and structural awareness, and it should do better on broader, multi-page questions than on the narrow single-fact lookups that dominate this particular evaluation set.

5 Discussion

Three things here generalise past this specific pipeline. Start with the relation schema: an unconstrained one that collapses into a single dominant label can make a graph-augmented system worse than having no graph at all, and retrieval volume alone will not tell you this is happening, since the volume can look completely normal while the content is useless. Schema design needs the same scrutiny that graph presence gets in this literature, maybe more.

Cross-modal linking is the second issue, and it is harder on real financial documents than clean benchmark corpora would suggest. Chart labels are frequently just decontextualised numbers, not entities with independent names, so embedding similarity between names alone runs out of signal

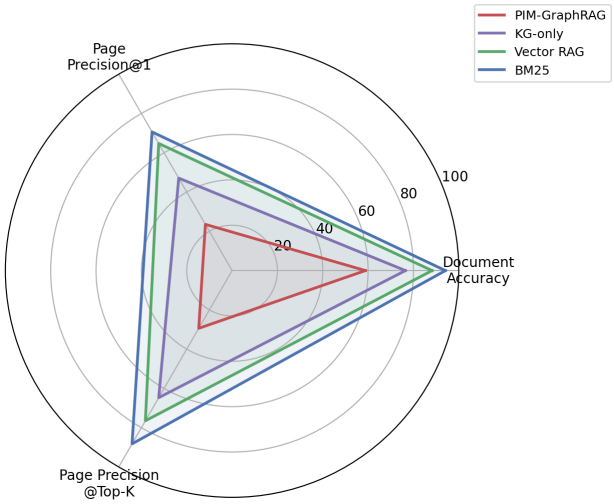

Figure 5: Overall performance profile across all four systems on the verified evaluation set. PIM-GraphRAG's smaller radar area reflects its consistent underperformance on narrowly factual queries.

quickly. Explicit chart-structure parsing, reading axes and legends directly rather than matching on text, is probably closer to what is actually needed.

The third is about navigation strategy. Structure-aware, summary-based retrieval is not automatically better than plain lexical or vector search; it depends entirely on the kind of question being asked. An evaluation set built mostly from single-fact lookups will always favour simpler baselines, no matter how carefully the structured system was designed. Testing hierarchical navigation properly means deliberately splitting single-fact questions from genuinely multi-page or cross-modal ones, rather than collapsing everything into one aggregate accuracy number and calling it a result.

6 Limitations

The knowledge graph and evaluation both rest on a stratified sample of 150 pages, not the full 3,001-page corpus. That was a budget decision: the entire project, including the knowledge graph extraction, cross-modal alignment, and every baseline reported in this paper, ran on approximately $50 of OpenAI API credit, and 150 pages was what could be processed for the knowledge graph within that budget while still touching every report. Scaling up would likely shift the absolute number of cross-modal links, but the underlying finding, that chart labels are hard to link, would probably survive intact. The 17-question verified set is small and leans heavily toward single-fact lookup; a bigger, more deliberately stratified set is needed before anyone can say with confidence whether the proposed system's structural design pays off on genuinely multi-page or cross-modal questions. Cross-modal linking here relied only on embedding similarity over entity names; a more structured, vision-grounded approach to parsing chart geometry directly was never attempted. Document and page selection used summary-based search only; a hybrid approach mixing summaries with direct keyword or entity matching did not get a full evaluation within the time and budget available. Finally, the ablation in Section 4.2 compares complete systems rather than removing one component from PIM-GraphRAG at a time; a within-system ablation, for instance PIM-GraphRAG with cross-modal linking disabled, would isolate each component's contribution more precisely than the current cross-system comparison does.

7 Future Work

A few directions stand out. Scaling knowledge graph extraction to the full corpus, budget permitting, is the most obvious; the roughly $50 total spend on this project suggests the full 3,001-page corpus is well within reach of a modestly larger budget. Building a larger evaluation set that deliberately separates single-fact lookup from multi-page and cross-modal reasoning would settle the open question

from Section 6. Hybridising summary-based document and page selection with direct keyword or entity matching is worth testing properly rather than leaving it as a hunch. Running a within-system ablation, disabling cross-modal linking or the fixed schema one at a time within PIM-GraphRAG itself, would give a cleaner attribution of each component's effect than the cross-system comparison in Section 4.2. And vision-grounded chart parsing, reading axis labels and legends structurally instead of relying on embedding similarity, seems like the more promising path for cross-modal linking specifically.

8 Conclusion

This paper built and tested an end-to-end pipeline: PageIndex-style hierarchical navigation, a schema-constrained multimodal knowledge graph, training-free cross-modal entity alignment, run against ten real financial annual reports. It lost to simpler baselines on a narrowly factual, verified evaluation set. A companion ablation showed something arguably more useful than a win would have been: naive relation extraction can make graph augmentation actively harmful, independent of how much content gets retrieved. Relation-schema design and cross-modal entity linking are both open problems in this field, and this paper attaches real numbers to both instead of leaving them as general warnings.

LLM Usage Disclosure

Generative AI tools were used only for editing, formatting, and code debugging assistance, not for research ideas, experiments, or results. All work is the author's own, who takes full responsibility for its content.

Reproducibility Statement

Section 3 covers the full pipeline architecture, with exact hyperparameters in Appendix B, the fixed entity/relation schema in Section 3.4, and the cross-modal alignment procedure with its similarity threshold in Section 3.5, Equation 3. Appendix C has pseudocode for both the offline index construction and the online query-time generation stages (Algorithms 1–2). Appendix A lists per-report corpus statistics and extraction outputs. DocVQA is a publicly released dataset (4); the ten annual reports making up the primary corpus are publicly filed regulatory documents, freely downloadable from each company's investor relations website (full list in Section 4.1). Total OpenAI API spend for the complete project, all baselines and the proposed system included, was approximately $50, reported here to support cost estimation for anyone attempting to reproduce or extend this work. Code, PageIndex trees, knowledge graph extraction outputs, RAGAS scores, and the Streamlit evaluation dashboard will be made available in a public repository upon publication.

References

[1] Chen, X. et al. 2025. MMGraphRAG: Bridging Vision and Language with Interpretable Multimodal Knowledge Graphs. *arXiv:2507.20804 [cs.AI]*.

[2] Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., Metropolitansky, D., Ness, R. O., & Larson, J. 2024. From Local to Global: A Graph RAG Approach to Query-Focused Summarization. *arXiv:2404.16130 [cs.CL]*.

[3] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS 2020)*, Article 793.

[4] Mathew, M., Karatzas, D., & Jawahar, C. V. 2021. DocVQA: A Dataset for VQA on Document Images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV 2021)*.

[5] PageIndex Documentation. 2024. PageIndex: Tree-Structured Page Indexing for Document Retrieval. Available at: https://pageindex.ai/blog/pageindex-intro.

[6] Peng, B., Zhu, Y., Liu, Y., Bo, X., Shi, H., Hong, C., Zhang, Y., & Tang, S. 2024. Graph Retrieval-Augmented Generation: A Survey. *arXiv:2408.08921 [cs.AI]*.

[7] Shearer, C. 2000. The CRISP-DM Model: The New Blueprint for Data Mining. *Journal of Data Warehousing*, 5(4).

[8] Xiang, Z., Wu, C., Zhang, Q., Chen, S., Hong, Z., Huang, X., & Su, J. 2025. When to use Graphs in RAG: A Comprehensive Analysis for Graph Retrieval-Augmented Generation. *arXiv:2506.05690 [cs.CL]*.

[9] Zhang, Y., Liu, X., Li, H. et al. 2025. RAG-Anything: All-Modality RAG System for Comprehensive Information Retrieval. *arXiv preprint arXiv:2505.09977*.

[10] Zhao, X., Chen, Y., Wang, Z. et al. 2025. Multimodal RAG: A Comprehensive Survey. *arXiv preprint arXiv:2510.15253*.

A Corpus and Extraction Statistics

Table 4: Per-report knowledge graph extraction statistics.

Report	Pages Proc.	Total Entities	Total Relations	Avg. Ent./Page
BP (2024)	15	323	233	21.5
Barclays (2024)	15	295	204	19.7
AstraZeneca (2025)	15	275	219	18.3
Marks & Spencer (2024)	15	243	178	16.2
Tesco (2025)	15	214	185	14.3
Rolls-Royce (2024)	15	212	155	14.1
HSBC (2024)	15	205	147	13.7
National Grid (2024)	15	204	173	13.6
Unilever (2024)	15	200	181	13.3
BT Group (2024)	15	176	142	11.7

Table 5: Cross-modal entity links by document.

Document	Cross-Modal Links Found
Barclays (2024)	12
BP (2024)	2
National Grid (2024)	2
Remaining seven reports	0

B Extraction and Alignment Hyperparameters

Parameter	Value
Entity type schema size	9 fixed types
Vision model	GPT-4o Vision
Extraction model	GPT-4o-mini
Chunk size (baseline RAG)	512 tokens
Cross-modal similarity threshold τ	0.75
Sampled pages per report	15
Total sampled pages	150
Vision-dependent pages	94
Text-only sampled pages	56
Total OpenAI API spend (project-wide)	$\sim$\$50

C Pipeline Pseudocode

Algorithm 1 PIM-GraphRAG Offline Index Construction

Require: Corpus D, entity schema $\mathcal{E}_{\text{types}}$, relation schema $\mathcal{R}_{\text{types}}$
Ensure: PageIndex trees $\{T_i\}$, knowledge graph G, cross-modal links L

for each document $d_i \in D$ **do**
  $T_i \leftarrow \text{PageIndexBuild}(d_i)$ ▷ Section 3.2
  $P_i \leftarrow \text{SamplePages}(d_i,\ \text{weight} = \text{table_or_image_density})$
  for each page $p \in P_i$ **do**
    $\text{needs_vision}(p) \leftarrow \text{HasTable}(p) \vee \text{len}(\text{text}(p)) < 50$
    if $\text{needs_vision}(p)$ **then**
      $\text{entities}_p, \text{relations}_p \leftarrow \text{GPT4oVision}(p, \mathcal{E}_{\text{types}}, \mathcal{R}_{\text{types}})$
    else
      $\text{entities}_p, \text{relations}_p \leftarrow \text{GPT4oMini}(\text{text}(p), \mathcal{E}_{\text{types}}, \mathcal{R}_{\text{types}})$
    end if
    $G.\text{add}(\text{entities}_p, \text{relations}_p)$
  end for
end for
for each pair $(e_i, e_j) \in G.\text{entities}$ with $m(e_i) = \text{text}$, $m(e_j) = \text{visual}$ **do**
  if $\text{sim}(\text{embed}(e_i), \text{embed}(e_j)) > \tau$ **and** $\text{page_distance}(e_i, e_j) \leq w$ **then**
    if $\neg\ \text{IsFiscalYearNoise}(e_i, e_j)$ **then**
      $L.\text{add}(\text{merge}(e_i, e_j))$
    end if
  end if
end for
return $\{T_i\}, G, L$

Algorithm 2 PIM-GraphRAG Query-Time Answer Generation

Require: Query q, PageIndex trees $\{T_i\}$, knowledge graph G, cross-modal links L
Ensure: Answer A

$\text{doc_summaries} \leftarrow \{\text{Summary}(d_i) \mid d_i \in D\}$
$\text{top_docs} \leftarrow \text{TopK}(\text{doc_summaries}, q, k{=}2)$ ▷ Stage 1: document-level
for each $d \in \text{top_docs}$ **do**
  $\text{page_summaries} \leftarrow \{\text{Summary}(p) \mid p \in \text{Pages}(d)\}$
  $\text{top_pages} \leftarrow \text{TopK}(\text{page_summaries}, q, k{=}3)$ ▷ Stage 2: page-level
end for
$\text{context} \leftarrow \text{Assemble}(\text{text}(\text{top_pages}), \text{visuals}(\text{top_pages}), G.\text{relations}(\text{top_pages}), L)$
$A \leftarrow \text{LLMGenerate}(q, \text{context})$
return A