Evaluating Modern RAG: Textual, Multimodal, Dense, and Late Interaction Pipelines

Emre Kuru
Özyeğin University
Istanbul, Türkiye
emre.kuru@ozu.edu.tr

Mehmet Onur Keskin
Özyeğin University
Istanbul, Türkiye
onur.keskin@ozu.edu.tr

Abstract

Retrieval-augmented generation (RAG) systems have traditionally relied on text-based pipelines that extract and retrieve information from documents. While efficient and lightweight, these approaches often struggle with documents where meaning is conveyed through layout, tables, and visual elements. Recent advances in multimodal pipelines, powered by advanced vision-language models (VLMs), offer significantly improved retrieval quality by jointly encoding visual and textual signals. However, these gains come with increased memory requirements and higher indexing and retrieval latency. To navigate these trade-offs, we propose a quantitative, data-driven selection methodology, guiding practitioners in identifying the most suitable RAG pipeline for a given document corpus based on empirical performance and resource constraints. We evaluate leading contemporary textual and multimodal pipelines, including dense and late-interaction architectures, analyze their strengths and limitations, and examine the types of documents they are equipped to handle. Our study provides practical guidance on selecting retrieval architectures that balance retrieval effectiveness with system-level efficiency, offering clear criteria for when the benefits of sophisticated multimodal systems outweigh their operational costs.

CCS Concepts

• **Information systems** → **Evaluation of retrieval results**; **Multimedia and multimodal retrieval**; *Question answering*.

Keywords

Retrieval-Augmented Generation, Multimodal Retrieval

ACM Reference Format:
Emre Kuru and Mehmet Onur Keskin. 2025. Evaluating Modern RAG: Textual, Multimodal, Dense, and Late Interaction Pipelines. In *Information Retrieval's Role in RAG Systems (IR-RAG '25) at SIGIR 2025, July 17, 2025, Padova, Italy*. ACM, New York, NY, USA, 10 pages. https://doi.org/TBD

1 Introduction

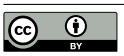
Retrieval-Augmented Generation (RAG) systems have emerged as a powerful solution for grounding large language models in external document corpora, particularly in knowledge-intensive domains such as finance, healthcare, and law [7, 9, 12]. Early RAG pipelines operated on purely textual representations in such domains, extracting content from scanned documents using Optical Character Recognition (OCR) [13]. These *text-only pipelines* proved effective for linear documents with clean structure, offering fast and lightweight retrieval.

Although widely adopted due to their simplicity and efficiency, traditional text-only pipelines face significant limitations when applied to visually complex documents. OCR tends to flatten structural and spatial information, such as table layouts, section headers, or figure references, into unstructured text, often leading to the loss of semantic cues critical for accurate retrieval [18, 19]. As a result, such pipelines frequently underperform tasks that require an understanding of document layout, tabular reasoning, or visual context.

Several enhanced text-based methods have been proposed to mitigate these limitations, such as layout-aware approaches. That incorporates structural information by applying document layout parsers [8, 15] to segment documents into semantically meaningful regions such as titles, captions, tables, and figures, and using these annotations to guide chunking strategies [17], providing a partial structural understanding of the document. Furthermore, image captioning techniques have been used to convert visual elements (e.g., charts or figures) that cannot be captured by OCR into descriptive text [20], allowing them to be indexed and retrieved within a traditional text-only framework. These enhancements extended the reach of text-based pipelines into visually rich domains while maintaining relatively low computational overhead.

Nonetheless, even with such augmentations, text-based approaches still operate on symbolic approximations of visual content. They remain constrained by the quality of the layout or captioning models and often struggle to preserve fine-grained spatial relationships and formatting nuances. This led to the emergence of *multimodal pipelines* [3, 6], which operate directly on document images using vision-language models (VLMs). By encoding full-page inputs, these models jointly capture textual content, spatial layout, and visual features in a single embedding, dramatically improving retrieval in visually complex settings. However, they come with steep system-level costs, including increased memory consumption, indexing, and retrieval latency.

In parallel to this modality shift, retrieval architectures have also evolved. Traditional retrieval architectures have embedded these documents (text or image) into a single vector, enabling fast similarity search. However, such representations compress the information, leading to diminished retrieval precision, especially when fine-grained semantic or structural alignment is required. To address this, *late interaction* architectures [4] have been proposed. Instead of encoding the input into a single vector, these models preserve multiple sub-representations and perform more expressive



IR-RAG @ SIGIR '25, Padova, Italy

ACM ISBN TBD
https://doi.org/TBD

similarity computations at retrieval time. While this boosts performance, particularly in layout-sensitive queries, it also introduces higher latency, increased memory usage, and larger index sizes.

Despite the growing variety of pipeline designs, there remains little guidance on selecting the best approach for a specific task. This work addresses this gap by comprehensively evaluating state-of-the-art RAG retrieval pipelines across the two key design axes discussed: modality (text vs. vision) and architecture (dense vs. late interaction). Our study benchmarks these pipelines on retrieval performance and across a range of Quality-of-Service (QoS) metrics on two challenging, real-world datasets that exhibit diverse document structures and layout complexity. Building on these findings, we propose a selection methodology for identifying the most suitable RAG pipeline for a given document corpus and computational constraints. This is a deployment-oriented framework for selecting the most efficient and accurate pipeline.

The rest of the paper is organized as follows. Section 2 reviews the related work. Section 3 describes the retrieval pipelines evaluated in this study. Section 4 presents the experimental setup and results across effectiveness, latency, and memory usage. Finally, Section 5 concludes the paper with key takeaways and future directions.

2 Related Work

Retrieval-augmented generation (RAG) has become a central paradigm for enhancing large language models with access to external knowledge sources. Early RAG pipelines operated primarily over textual inputs derived via Optical Character Recognition (OCR), with minimal attention paid to document layout or visual structure. Lin and Byrne introduced one of the earliest RAG architectures based on OCR outputs, integrating Dense Passage Retrieval (DPR) with a generative language model for open-domain question answering over scanned texts [5]. Their work demonstrated that standard dense retrieval methods could be extended to semi-structured documents with text extraction.

To better handle structurally rich documents, several works have incorporated layout-aware techniques. Yepes *et al.* proposed a structure-guided chunking strategy that segments documents into semantically distinct regions, such as titles, body text, and tables, using computer vision and NLP methods, thereby preserving some spatial information during retrieval [17]. However, OCR-based pipelines, even with such enhancements, remain constrained by transcription errors and the inherent flattening of spatial layout. Zhang *et al.* showed that such limitations can significantly degrade both retrieval accuracy and generation quality, especially in documents where meaning is intrinsically tied to structure [19].

Recognizing these persistent limitations, recent work has explored multimodal retrieval pipelines that directly encode visual signals. Chen *et al.* introduced MuRAG, a memory-augmented architecture capable of reasoning over both image and text embeddings [1]. Ma *et al.* proposed a unified embedding space for document screenshots, enabling dense retrieval over raw visual input [6]. Similarly, Riedler *et al.* applied CLIP-based retrieval to visually complex documents, demonstrating the benefits of joint vision-language representations [10]. Furthermore, Faysse *et al.* introduced ColPali, a visual late interaction architecture that extends the ColBERT paradigm to full-page images [3]. By retaining patch-level representations and applying token-wise similarity scoring at query time, ColPali achieves state-of-the-art retrieval performance on layout-sensitive benchmarks, exemplifying the performance gains achievable with sophisticated multimodal, late-interaction approaches.

While this body of work has significantly advanced retrieval performance across various modalities, most evaluations have focused narrowly on effectiveness metrics such as nDCG and Recall. As RAG systems become more powerful and complex, understanding their system-level implications—such as indexing latency, retrieval latency, and memory footprint—becomes paramount for practical deployment, especially at scale. In contrast to prior studies, our work offers a holistic evaluation of retrieval pipelines. We systematically compare dense and late interaction methods across both text and visual modalities, not only in terms of retrieval effectiveness but also with respect to these crucial Quality-of-Service (QoS) metrics, which are essential for real-world deployment scenarios. To our knowledge, this is one of the first comprehensive analyses directly comparing these four pipeline archetypes—Text-Dense, Text-Late Interaction, Visual-Dense, and Visual-Late Interaction—across both retrieval quality and detailed operational costs.

3 Proposed Approach

This work proposes a data-driven selection methodology for identifying the most suitable RAG retrieval pipeline given a document corpus and its computational constraints. This approach is grounded in a comprehensive evaluation of the state-of-the-art pipelines that represent key axes in modern retrieval system design: input modality (textual vs. visual) and retrieval architecture (dense vs. late interaction). This section outlines the design decisions behind these pipelines, detailing how they preprocess and embed documents and how their underlying architectures handle retrieval.

Figure 1 illustrates the differences between text-based and multimodal pipelines. Text-based pipelines typically follow a multi-stage preprocessing procedure. Raw document pages are first passed through Optical Character Recognition (OCR) to extract textual content. Afterward, the layout analysis step is applied, segmenting the document into semantically meaningful regions such as tables, titles, and figures. These regions inform layout-aware chunking strategies that divide the extracted text into retrieval-ready text chunks. An image captioning module is applied to generate natural language descriptions that can be indexed alongside the text for chunks containing non-textual content, such as images or tables. Finally, each chunk is embedded using a language model and stored in a vector database.

In contrast, multimodal pipelines operate directly on full-page document images. These systems leverage a vision-language model (VLM) to process the entire page holistically, generating embeddings that capture textual, visual, and spatial cues in a unified representation. This end-to-end approach eliminates the need for OCR, chunking, or captioning, enabling a more direct and layout-aware document encoding. The resulting embeddings are stored directly in the vector database.

Figure 2 illustrates the architectural differences between dense and late interaction retrieval mechanisms. In dense retrieval, the

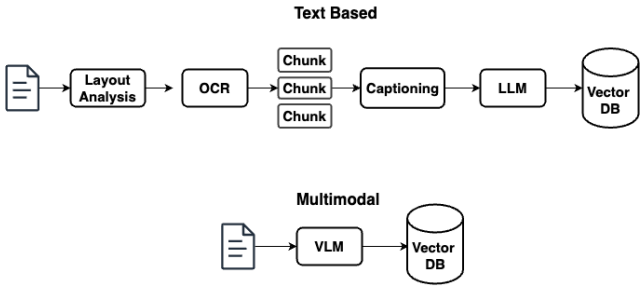


Figure 1: Indexing Workflows of Text & Multimodal Pipelines

document and query are encoded into single vector representations using a language or vision-language model. Retrieval is performed by computing the similarity between these vectors using a distance metric such as cosine similarity. This highly efficient approach scales well, making it well-suited for latency-sensitive applications. However, its compressed representation can lose fine-grained alignment, particularly important in structured or layout-dependent documents. Late interaction architectures address this limitation by preserving token-level or patch-level granularity during indexing. Instead of compressing the input into a single embedding, these models retain multiple embeddings per document, enabling more detailed and expressive similarity calculations at query time. A common scoring mechanism is MaxSim, where each query token is compared with all document tokens, and the maximum similarity scores are aggregated. While this approach improves retrieval precision, it also increases storage requirements and retrieval latency.

3.1 Proposed Pipelines

As discussed, we evaluate four retrieval pipelines that span the key design dimensions in modern RAG systems: input modality (text vs. vision) and retrieval architecture (dense vs. late interaction).

Text-Dense. This pipeline implements a dense retrieval architecture of over-extracted text. As is common practice, we rely on the `Unstructured`[1] off-the-shelf tool in the highest resolution settings for OCR, layout analysis, and chunking (by-title). Afterward, for chunks that include non-textual elements, we set up a full-fledged captioning strategy, in which we feed the visual elements to a state-of-the-art Vision Language Model (Google-Gemini 2 0 Flash[2]) to obtain highly detailed textual descriptions of the elements. Each resulting chunk is then embedded into a dense vector using a text encoder.

Text-Late Interaction. This pipeline retains the same preprocessing strategy as the dense variant but uses a late interaction architecture to improve token-level alignment. Instead of embedding each chunk into a single vector, it preserves multiple sub-token embeddings, allowing fine-grained comparisons between query and document terms.

Visual-Dense. This pipeline skips OCR and processes full-page document images directly using a vision-language model. Each image is encoded into a dense vector that jointly captures visual and textual signals. This design allows for compact, layout-sensitive representations with minimal preprocessing.

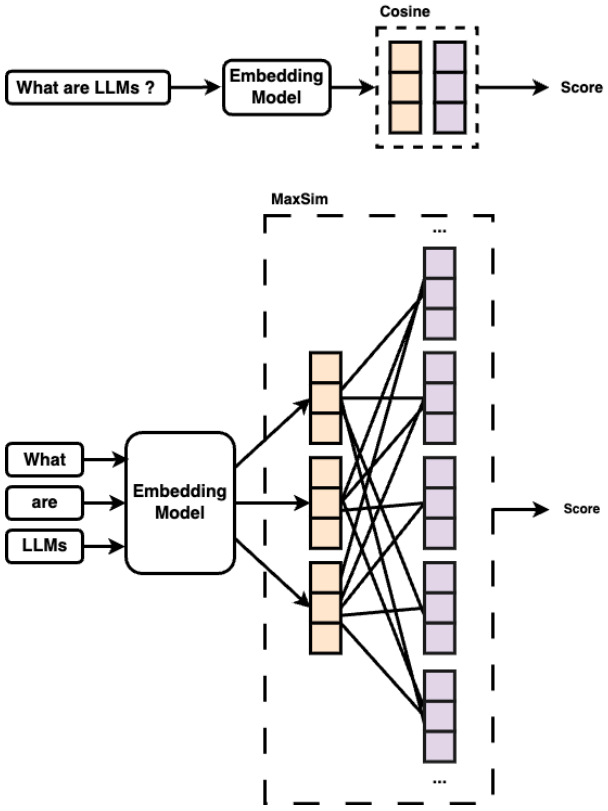


Figure 2: Dense vs Late Interaction Retrieval Architectures

Visual-Late Interaction. This pipeline operates on full-page images like the dense variant but uses a late interaction retrieval mechanism. It produces multi-vector embeddings for each document patch and performs expressive similarity scoring with patch-level granularity. This improves retrieval precision for layout-sensitive queries at the cost of greater computational overhead.

4 Evaluation

This section evaluates the proposed pipelines regarding retrieval performance across various QoS metrics, including memory consumption, indexing time, and retrieval latency.

The remainder of this section is organized as follows. Section 4.1 describes the experiment setting.Section 4.2 reports on retrieval effectiveness using standard ranking metrics to compare the pipelines' retrieval performances. Finally, Section 4.3 presents our quality of service (QoS) evaluation, analyzing memory consumption, indexing, and retrieval latency.

4.1 Expiriment Setting

[1] www.unstructured.io
[2] https://ai.google.dev/gemini-api/docs/models
[3] https://qdrant.tech
[4] https://huggingface.co/Linq-AI-Research/Linq-Embed-Mistral
[5] https://huggingface.co/lightonai/GTE-ModernColBERT-v1

All experiments were conducted on a single NVIDIA H100 GPU with 80GB of memory. Each retrieval pipeline was evaluated independently using the same document corpus and query set to ensure a fair comparison. We implemented each pipeline under the same codebase and hardware to eliminate performance differences due to infrastructure. To ensure consistency across pipelines, all vector indices were stored using identical Qdrant[7] clusters.

4.1.1 Datasets. We evaluate our retrieval pipelines using the REAL-MM-RAG benchmark [14], a recent multimodal dataset designed for realistic document retrieval in structured domains. Unlike earlier benchmarks, REAL-MM-RAG focuses on long, scanned documents with complex layouts, incorporating visual elements such as tables, charts, and diagrams. The queries are generated to reflect natural user questions rather than direct surface-level matches, providing a more rigorous test of retrieval models. To assess robustness under linguistic variation, each query in the dataset is rewritten up to three times using a large language model, resulting in four levels of query phrasing. These rephrased versions introduce increasing syntactic and lexical variation, allowing us to measure how well retrieval pipelines generalize beyond superficial token overlap and capture true semantic intent. We focus on two challenging subsets from this benchmark.

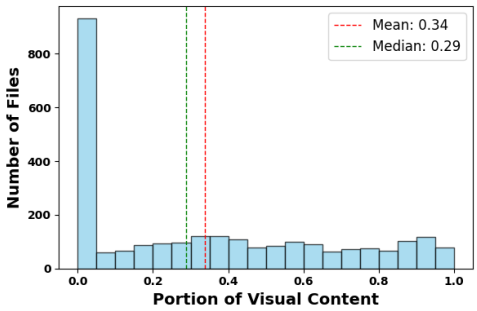
FinReport contains annual financial reports from IBM between 2005 and 2023. These documents include dense narrative content, structured financial tables, and visually segmented layouts. Queries typically require extracting specific values, understanding table structures, and reasoning over multiple modalities. This dataset is ideal for evaluating performance in table-heavy, layout-sensitive retrieval tasks.

TechReport consists of IBM technical documentation across systems like FlashSystem and Power Systems. These documents are primarily textual but include frequent visual components such as block diagrams, system schematics, and formatted tables. The queries in this domain target procedural details and technical descriptions, testing a model's ability to retrieve relevant explanatory content embedded in text and images.

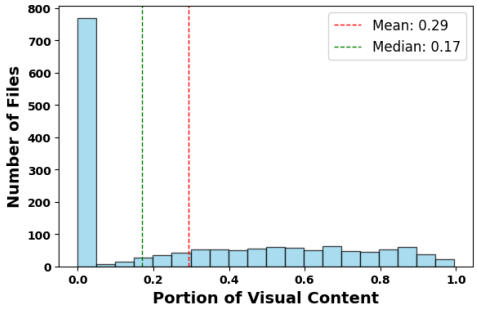
These datasets were chosen because they enable a fine-grained evaluation of how visual modality influences retrieval performance. Specifically, each document's proportion of visual content varies in both datasets, providing a balanced distribution of visually sparse and rich documents. This makes it possible to observe how retrieval pipelines perform under different visual conditions. As shown in Figure 5, both datasets exhibit a broad range of visual content levels.

4.1.2 Models. Each pipeline is built to represent the state-of-the-art, open-access models in their respective configurations. Specifically, text-based models were chosen from the top entries on the MTEB leaderboard [2], while vision-based models were selected based on rankings in the ViDoRe leaderboard [3]. The following describes the specific models used for each pipeline and their underlying architectures:

- **Baseline**: BM25 [11], a sparse retrieval baseline widely used in information retrieval. BM25 scores documents based on exact token matches, using term frequency and inverse document frequency (TF-IDF) weighting. It does not use any learned parameters or embeddings.



(a) FinReport



(b) TechReport

Figure 3: Visual Component Rate Histogram across Datasets

- **Text-single**: Linq-Embed-Mistral[4], is built upon the Mistral-7B architecture, fine-tuned for text retrieval tasks. It generates dense embeddings of 768 dimensions and supports a context window of 4096 tokens. The model size is approximately 7.11 billion parameters.
- **Text-multi**: GTE-ModernColBERT-v1[5], a ColBERT-style embedding model optimized for fine-grained token matching. It produces 128-dimensional embeddings per token, resulting in multi-vector representations. The model comprises around 110 million parameters and supports a context window of 8192 tokens.
- **Visual-single**: nomic-embed-multimodal-7b[6], a dense multimodal retriever based on the Bi-Qwen 2.5 architecture, an extension of the Qwen2.5-VL-7B vision-language model [16]. It generates 768-dimensional dense embeddings per page.
- **Visual-multi**: ColQwen2.5-7b-multilingual-v1.0[7], a ColPali-style retriever built on the ColQwen 2.5 architecture, another extension of the Qwen2.5-VL-7B vision-language model [16]. Unlike its dense counterpart, ColQwen produces multi-vector representations for text and images, enabling more fine-grained matching between queries and document regions.

4.2 Retrieval Performance

Retrieval pipelines must ultimately be judged by their ability to return relevant content in response to a query. To quantify retrieval

[6]https://huggingface.co/nomic-ai/nomic-embed-multimodal-7b
[7]https://huggingface.co/Metric-AI/ColQwen2.5-7b-multilingual-v1.0

effectiveness, this section evaluates each pipeline using four widely adopted top-k ranking metrics (in our case, we chose $k = 5$):

- **nDCG@5 (Normalized Discounted Cumulative Gain):** Captures graded relevance by rewarding highly ranked relevant documents more than later in the result list.
- **MRR@5 (Mean Reciprocal Rank):** Reflects how early the first relevant document is retrieved, favoring pipelines that return relevant results near the top.
- **Precision@5:** Measures the proportion of retrieved documents within the top 5 that are relevant.
- **Recall@5:** Estimates the fraction of all relevant documents that appear in the top 5 results.

These metrics provide a comprehensive view of retrieval performance, balancing ranking position, accuracy, and completeness. Table 1 presents retrieval performance across four rephrasing levels for each pipeline and dataset. To assess the statistical significance of these results, we adopt a two-stage testing procedure. First, we apply group-level significance tests to determine whether any pipeline significantly differs from the rest. If the group contains three or more pipelines, we apply a *Repeated Measures ANOVA* when normality is satisfied, and fall back to the *Friedman test* otherwise. For pairwise comparisons, we apply either the *Paired t-test* (for normally distributed and homogeneous samples) or the *Wilcoxon Signed-Rank Test* (for non-normal or heterogeneous samples). Normality is tested using the *Kolmogorov-Smirnov test*, and homogeneity using *Levene's test*.

The *visual-multi* pipeline significantly outperforms all other methods ($p < 0.01$) across all metrics and rephrasing levels, highlighting its robustness and superior generalization. Notably, even the *visual-single* (dense vision-language) pipeline significantly outperforms all text-based pipelines across all rephrasing levels and metrics, confirming that multimodal pipelines; regardless of representation granularity, offer a clear and statistically significant advantage over purely textual models.

Furthermore, while the baseline *BM25* performs competitively on the original query set (occasionally surpassing some text-based pipelines) its performance degrades very sharply under rephrasing. This consistent drop underscores *BM25's* reliance on exact token matches and its vulnerability to lexical variation.

Conversely, the proposed pipelines exhibit moderate performance drops as query rephrasing increases, reflecting the increased difficulty of matching semantically similar but lexically varied queries. However, this degradation is relatively consistent across architectures, with no single pipeline type disproportionately affected. These results suggest that vision-based retrieval maintains a consistent advantage even under paraphrasing stress, while all pipelines are sensitive to linguistic variation.

All pipelines exhibit a moderate performance drop as query rephrasing increases, reflecting the increased difficulty of matching semantically similar but lexically varied queries. However, this degradation is relatively consistent across architectures, with no single pipeline type disproportionately affected. These results suggest that vision-based retrieval maintains a consistent advantage even under paraphrasing stress, while all pipelines are sensitive to linguistic variation.

To illustrate the strengths and limitations of different retrieval pipelines, two qualitative case studies are examined. These examples highlight how performance is affected across modality (text vs. vision) and retrieval architecture (dense vs. late interaction). The first case study comes from our FinReport dataset and focuses on the impact of modality choice, particularly how OCR-induced errors can compromise retrieval accuracy in text-based pipelines. In this example, the query asks about the change in IBM's total company debt between 2004 and 2005. However, the years in the corresponding financial table are entirely omitted during OCR preprocessing. As a result, both text pipelines fail to locate the correct context, retrieving irrelevant passages instead. In contrast, the visual pipelines operate directly on the document image and successfully place the correct page at the top of the ranking. This example underscores how early-stage OCR failures can cascade through text-based retrieval systems and highlights the robustness of vision-based models in preserving layout-dependent information.

Case 1: Correct Context

Q: What changed IBM's company debt from **2004 to 2005?**

Year	**2005**	**2004**
Total company debt	$22,641	$22,927
Non-Global Financing debt	$2,142	$607
Non-Global Financing debt capitalization	6.7%	2.1%

Case 1: OCR output

Total company debt	$22,641	$22,927
Non-Global Financing debt	$2,142	$607
Non-Global Financing debt capitalization	6.7%	2.1%

The second case study examines the effects of retrieval architecture, particularly the distinction between dense and late interaction models in the face of linguistic variation. In this example, all pipelines successfully retrieve the correct document when the query explicitly includes the term bandwidth, which aligns closely with the vocabulary used in the document. However, when the query is rephrased using the semantically equivalent phrase "maximum amount of data that can be transferred per unit of time", only the late interaction models retain the correct context at the top rank. The dense models, in contrast, fail to match the rephrased version due to their reliance on surface-level term overlap in fixed vector representations. This case highlights a key limitation of dense retrieval in real-world scenarios where query phrasing is highly variable. It demonstrates how late interaction architectures offer greater robustness by enabling finer-grained token-level comparisons that better capture semantic nuance.

Table 1: Retrieval Results Across Rephrase Levels Per Dataset

Level	Pipeline	FinReport				TechReport			
		nDCG@5	MRR@5	Precision@5	Recall@5	nDCG@5	MRR@5	Precision@5	Recall@5
0	BM25	0.43 ± 0.43	0.39 ± 0.43	0.11 ± 0.10	0.53 ± 0.50	0.61 ± 0.42	0.57 ± 0.43	0.14 ± 0.09	0.72 ± 0.45
	text-single	0.46 ± 0.42	0.41 ± 0.42	0.12 ± 0.10	0.60 ± 0.49	0.61 ± 0.39	0.56 ± 0.41	0.16 ± 0.08	0.78 ± 0.42
	text-multi	0.32 ± 0.43	0.30 ± 0.42	0.08 ± 0.10	0.40 ± 0.49	0.53 ± 0.45	0.50 ± 0.46	0.12 ± 0.10	0.62 ± 0.48
	visual-single	0.66 ± 0.39	0.61 ± 0.41	0.16 ± 0.08	0.80 ± 0.40	0.73 ± 0.37	0.69 ± 0.39	0.17 ± 0.07	0.85 ± 0.36
	visual-multi	**0.74 ± 0.33**	**0.69 ± 0.37**	**0.18 ± 0.06**	**0.89 ± 0.31**	**0.85 ± 0.28**	**0.82 ± 0.32**	**0.19 ± 0.05**	**0.94 ± 0.24**
1	BM25	0.31 ± 0.40	0.28 ± 0.39	0.08 ± 0.10	0.41 ± 0.49	0.45 ± 0.44	0.42 ± 0.44	0.11 ± 0.10	0.56 ± 0.50
	text-single	0.43 ± 0.42	0.39 ± 0.41	0.12 ± 0.10	0.58 ± 0.49	0.56 ± 0.40	0.50 ± 0.41	0.14 ± 0.09	0.72 ± 0.45
	text-multi	0.28 ± 0.41	0.26 ± 0.41	0.07 ± 0.09	0.34 ± 0.47	0.48 ± 0.45	0.45 ± 0.45	0.12 ± 0.10	0.58 ± 0.49
	visual-single	0.59 ± 0.42	0.54 ± 0.43	0.14 ± 0.09	0.72 ± 0.45	0.63 ± 0.40	0.59 ± 0.42	0.15 ± 0.08	0.77 ± 0.42
	visual-multi	**0.68 ± 0.37**	**0.63 ± 0.39**	**0.17 ± 0.07**	**0.84 ± 0.37**	**0.78 ± 0.33**	**0.75 ± 0.37**	**0.18 ± 0.06**	**0.89 ± 0.31**
2	BM25	0.22 ± 0.37	0.20 ± 0.36	0.06 ± 0.09	0.29 ± 0.45	0.38 ± 0.42	0.34 ± 0.41	0.10 ± 0.10	0.49 ± 0.50
	text-single	0.40 ± 0.41	0.35 ± 0.41	0.11 ± 0.10	0.54 ± 0.50	0.53 ± 0.41	0.48 ± 0.42	0.14 ± 0.09	0.69 ± 0.46
	text-multi	0.27 ± 0.40	0.24 ± 0.39	0.07 ± 0.09	0.34 ± 0.47	0.46 ± 0.44	0.42 ± 0.44	0.11 ± 0.10	0.56 ± 0.50
	visual-single	0.52 ± 0.42	0.47 ± 0.43	0.13 ± 0.09	0.66 ± 0.47	0.58 ± 0.42	0.53 ± 0.43	0.14 ± 0.09	0.71 ± 0.45
	visual-multi	**0.61 ± 0.40**	**0.55 ± 0.42**	**0.15 ± 0.09**	**0.76 ± 0.43**	**0.75 ± 0.35**	**0.72 ± 0.38**	**0.17 ± 0.07**	**0.87 ± 0.34**
3	BM25	0.18 ± 0.34	0.16 ± 0.33	0.05 ± 0.09	0.24 ± 0.43	0.32 ± 0.41	0.29 ± 0.40	0.08 ± 0.10	0.40 ± 0.49
	text-single	0.38 ± 0.41	0.34 ± 0.40	0.10 ± 0.10	0.51 ± 0.50	0.51 ± 0.41	0.46 ± 0.42	0.13 ± 0.09	0.67 ± 0.47
	text-multi	0.23 ± 0.38	0.21 ± 0.37	0.06 ± 0.09	0.30 ± 0.46	0.42 ± 0.44	0.39 ± 0.43	0.11 ± 0.10	0.53 ± 0.50
	visual-single	0.48 ± 0.43	0.43 ± 0.43	0.12 ± 0.10	0.61 ± 0.49	0.55 ± 0.42	0.50 ± 0.43	0.14 ± 0.09	0.68 ± 0.47
	visual-multi	**0.57 ± 0.41**	**0.52 ± 0.42**	**0.14 ± 0.09**	**0.72 ± 0.45**	**0.73 ± 0.36**	**0.68 ± 0.39**	**0.17 ± 0.07**	**0.85 ± 0.36**

Case 2: Relevant Document Context

With this setting, the Bronze customers can reach up to 1,000 Mbps, but they have to share the maximum **bandwidth** with all the other customers at the Bronze level. The Silver customers can reach up to 1,000 Mbps and do not have to share their limit with other customers in their performance class. The Gold customers are unlimited because they are not part of any performance class.

Case 2: Original Query

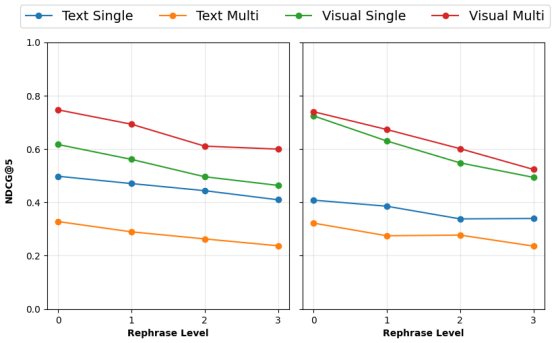
What is the **bandwidth** limit for Bronze customers in IBM FlashSystem A9000?

Case 2: Rephrased Query

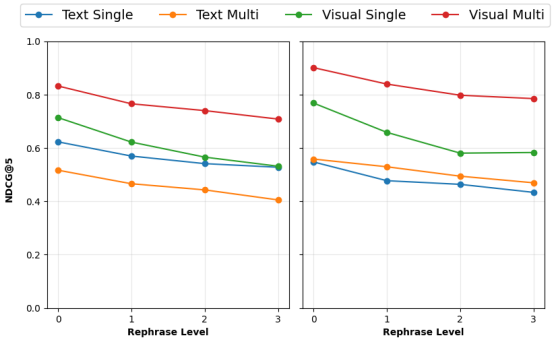
In the IBM FlashSystem A9000, what is the **maximum amount of data that can be transferred per unit of time** for users with a Bronze subscription?

Figure 4 breaks down retrieval performance by answer modality, referring to the type of document component in which the relevant information was embedded. Queries are labeled as *text-only* when the answer resides within textual elements such as paragraphs or titles, and as *visual-only* when the answer is found within visual elements like tables or images. This figure further dissects the pipeline strengths. For "text-only" queries (left column of subplots for each dataset), we anticipate that while all pipelines might perform reasonably well, the **Visual-multi** and **Visual-single** pipelines leveraging advanced VLMs may still hold an edge due to their holistic page understanding, potentially capturing contextual cues even from surrounding visual layout that text-only models might miss. The **Text-single** and **Text-multi** pipelines are expected to be strong contenders here, with `Text-multi` potentially showing greater robustness to rephrasing due to its late-interaction mechanism. Conversely, for "visual-only" queries (right column of subplots), a more significant performance gap is expected. **Visual-multi** should dominate, adeptly handling queries requiring interpretation of tables or diagrams directly. **Visual-single** is also expected to perform well, significantly outperforming text-based pipelines. The text-based pipelines (**Text-single** and **Text-multi**) will likely struggle considerably on "visual-only" queries, their success is highly dependent on the quality of OCR and any generated captions for those visual elements. Performance degradation across rephrasing levels (0 to 3) is anticipated for all query types and pipelines, but the visual pipelines, particularly **Visual-multi**, might exhibit more graceful degradation on "visual-only" queries due to their richer representations. Differences between FinReport and TechReport would also be telling: the more table-intensive FinReport dataset is likely to starkly highlight the superiority of visual pipelines for "visual-only" queries, whereas in TechReport, with its mix of text and diagrams, the distinctions might be nuanced but still favor visual approaches for visual queries.

Figure 5 breaks down retrieval performance (nDCG@5) by document visual content proportion, with subplots for rephrasing levels (0 to 3). Visual-multi leads, especially in TechReport. Text-single is comparable to Visual-single on mostly textual documents but degrades sharply with more visual content (especially in FinReport). Interestingly, Text-multi, despite underperforming Visual-single in



(a) FinReport



(b) TechReport

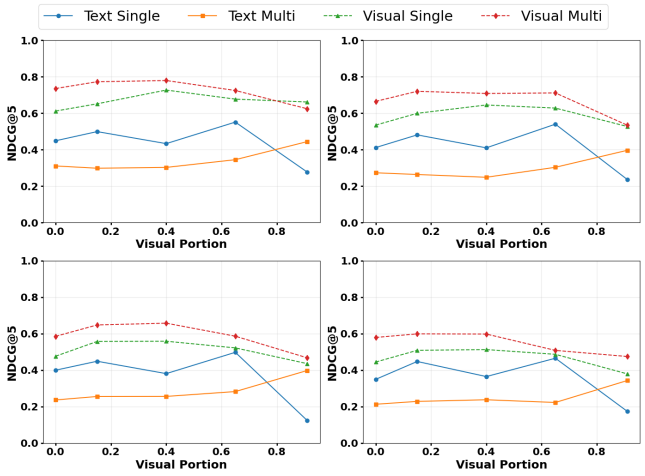
Figure 4: Retrieval Performance across Rephrase Levels for Text-Only (left) and Visual-Only (right) Queries for both Datasets

low-visual contexts, shows some resilience or even stronger results than Text-single in highly visual documents (FinReport, Level 0-1), suggesting its finer-grained token interactions might better navigate complex visual layouts indirectly via OCRed textual cues from those layouts.

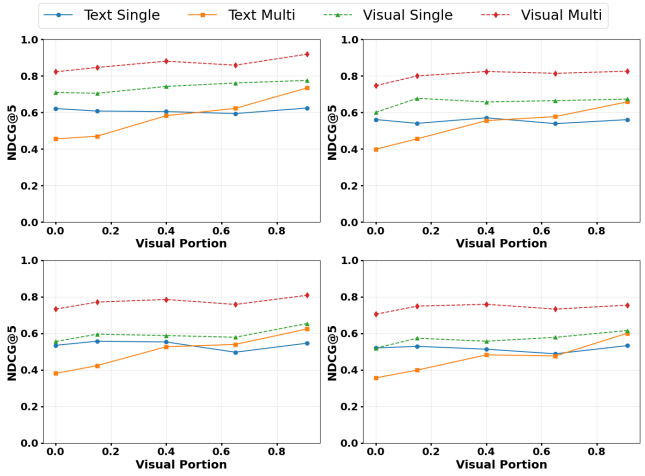
4.3 Quality of Service (QoS)

Deployment of RAG systems demands careful consideration of system-level efficiency. To capture these concerns, each pipeline is evaluated across three key Quality of Service (QoS) dimensions: memory consumption, indexing latency, and retrieval latency. Memory consumption refers to the total size of the vector index stored in the database. Indexing latency measures the end-to-end time required to preprocess, embed, and upsert documents into the retrieval system. Retrieval latency captures the time needed to embed a query and retrieve the top-k most relevant documents. Together, these metrics reflect real-world constraints such as hardware cost, scalability, and responsiveness, factors critical when deploying retrieval systems over large document corpora.

Table 2 presents the memory consumption of each pipeline. Due to their multi-vector representations, we can observe that late interaction models consume significantly more memory than their dense counterparts, which store embeddings at the token or patch level. This granularity allows for improved retrieval fidelity but comes at



(a) FinReport



(b) TechReport

Figure 5: Retrieval Performance across Visual Complexity for all Rephrase Levels (0-Top Left, 3-Bottom Right)

the cost of larger index sizes. Notably, *visual-multi* exceeds 5GB in memory usage, more than 10 times that of the compact *visual-dense* model. Interestingly, despite operating at higher embedding dimensionality, the visual-dense pipeline remains more storage-efficient overall. It encodes entire pages into a single vector, avoiding the cumulative overhead introduced by chunk-wise embeddings in text-based pipelines. Showcasing its scalability advantage in large-scale settings.

Table 2: Total Memory Consumption for both Datasets

Pipeline	Memory (MB)
text-single	390.00
text-multi	1079.00
visual-single	**340.00**
visual-multi	5068.80

Tables 3 and 4 show the average indexing workflows for both our datasets, broken down into preprocessing, embedding, and upserting stages. As expected, text-based pipelines incur substantial preprocessing overhead due to OCR, layout parsing, and captioning. Despite this, their embedding and upsert times remain relatively lower, particularly for dense variants. In contrast, visual pipelines skip preprocessing entirely, enabling faster end-to-end indexing, especially for the dense variant. However, due to the higher dimensionality, the visual multi-pipeline requires significantly longer upsert durations. These results highlight a key trade-off: text pipelines are preprocessing-heavy but compact, while visual pipelines are plug-and-play but slower to upload.

Table 3: Average Indexing Workflow in Seconds - FinReport

Stage	text-single	text-multi	visual-single	visual-multi
Preprocess	10.23	10.23	**0.00**	**0.00**
Embedding	**0.02**	0.18	0.72	0.71
Upsert	**1.24**	3.26	5.69	6.04
Total	11.49	13.67	**6.41**	6.75

Table 4: Average Indexing Workflow in Seconds - TechReport

Stage	text-single	text-multi	visual-single	visual-multi
Preprocess	10.70	10.70	**0.00**	**0.00**
Embedding	0.51	**0.19**	0.91	1.07
Upsert	**1.89**	2.03	6.93	9.00
Total	13.10	12.92	**7.84**	10.07

Table 5 reports the average retrieval latency across all pipelines, broken down into query embedding and retrieval components. As expected, dense pipelines, *text-single* and *visual-single*, exhibit significantly lower latency, with total times under one second. In contrast, late interaction pipelines introduce a significant performance gap. While *text-multi* remains within a negligible range, *visual-multi* incurs exceptionally high retrieval times, exceeding 20 seconds per query. This discrepancy reflects the computational overhead of multi-vector retrieval at scale, particularly for high-dimensional vision-based embeddings.

Table 5: Average Retrieval Latency in Seconds

Pipeline	TechReport			FinReport		
	Embedding	Retrieval	Total	Embedding	Retrieval	Total
text-single	0.04	**0.50**	**0.54**	0.03	**0.40**	**0.43**
text-multi	**0.01**	2.99	3.01	**0.01**	4.19	4.20
visual-single	0.05	0.58	0.63	0.05	0.55	0.60
visual-multi	0.05	22.53	22.58	0.03	27.35	27.39

5 Discussion

Our comprehensive evaluation of textual and multimodal RAG retrieval pipelines across dense and late interaction architectures yields several notable trends and practical insights.

First, the Text-multi (late interaction) pipeline consistently underperformed not only its visual counterparts but also the simpler Text-dense pipeline across most retrieval effectiveness metrics, while still incurring higher memory usage and slower retrieval latency than Text-dense. For instance, it achieved a mean nDCG@5 of only 0.32 on the FinReport dataset (Level 0 rephrase), significantly lower than Text-single (0.46) and Visual-single (0.66). In contrast, among the visual pipelines, Visual-multi (late interaction) achieved the highest retrieval performance overall, demonstrating the power of fine-grained, patch-level visual understanding. However, this superior effectiveness comes at a steep operational cost, particularly in terms of memory footprint (over 5GB) and retrieval latency (exceeding 20 seconds per query), making it potentially less suitable for resource-constrained or latency-sensitive scenarios.

Interestingly, our results challenge the common assumption that vision-based pipelines are invariably slower and more memory-intensive across the board. The Visual-single (dense) pipeline emerged as a compelling alternative, delivering significantly better retrieval performance than both text-based pipelines (e.g., 0.66 nDCG@5 on FinReport vs. 0.46 for Text-single and 0.32 for Text-multi) while also offering comparable or superior indexing throughput and the lowest memory consumption of all tested pipelines. This positions Visual-single as an excellent candidate for many real-world applications seeking a balance of effectiveness and efficiency.

Furthermore from our QoS results; we can observe that, embedding time itself was nearly negligible across all pipelines. The primary bottlenecks varied by modality: visual pipelines were predominantly constrained by the embedding and vector database upsert phase (especially Visual-multi), whereas text pipelines were bottlenecked by the extensive preprocessing phase (OCR, layout analysis, captioning).

6 Future Work

Our evaluation also revealed an important consideration for future benchmark development. Existing retrieval benchmarks, even those labeled as multimodal, often predominantly emphasize the retrieval of semantic answers that are textual, even if those answers are located within visual elements like charts or tables. This can create a bias where the unique challenges and capabilities of retrieving genuinely non-textual, visually grounded information are underrepresented with text retrieval from images. Future work should aim to address this gap by developing more rigid evaluation methodologies and datasets that specifically target the retrieval of diverse visual information, moving beyond text embedded in images.

Complementing this direction, we also plan to explore a hybrid retrieval strategy that allocates documents to the most suitable indexing pipeline based on their structure and retrieval needs. For example, visually complex documents containing tables or figures may be better indexed using late-interaction visual models to improve retrieval performance, while simpler documents may be handled more efficiently with dense pipelines. At query time, the system retrieves results from all pipelines and fuses the outputs, aiming to reduce memory usage and retrieval latency without significantly compromising retrieval quality.

In summary, this study provides a data-driven foundation for selecting appropriate RAG retrieval pipelines. Practitioners can use these findings—weighing the trade-offs between visual/textual modality, dense/late-interaction architecture, retrieval effectiveness, and QoS metrics—to make informed decisions tailored to their specific document corpora, task requirements, and operational constraints. These insights contribute to the ongoing effort to build more efficient, effective, and practically deployable RAG systems.

References

[1] Wenhu Chen, Hexiang Hu, Xi Chen, Pat Verga, and William W Cohen. 2022. Murag: Multimodal retrieval-augmented generator for open question answering over images and text. *arXiv preprint arXiv:2210.02928* (2022).

[2] Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Siblini, Dominik Krzemiński, Genta Indra Winata, et al. 2025. Mmteb: Massive multilingual text embedding benchmark. *arXiv preprint arXiv:2502.13595* (2025).

[3] Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. 2024. Colpali: Efficient document retrieval with vision language models. *arXiv preprint arXiv:2407.01449* (2024).

[4] Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 39–48.

[5] Weizhe Lin and Bill Byrne. 2022. Retrieval augmented visual question answering with outside knowledge. *arXiv preprint arXiv:2210.03809* (2022).

[6] Xueguang Ma, Sheng-Chieh Lin, Minghan Li, Wenhu Chen, and Jimmy Lin. 2024. Unifying multimodal retrieval via document screenshot embedding. *arXiv preprint arXiv:2406.11251* (2024).

[7] Karen Ka Yan Ng, Izuki Matsuba, and Peter Chengming Zhang. 2025. RAG in health care: a novel framework for improving communication and decision-making by addressing LLM limitations. *NEJM AI* 2, 1 (2025), AIra2400380.

[8] Birgit Pfitzmann, Christoph Auer, Michele Dolfi, Ahmed S Nassar, and Peter W J Staar. 2022. DocLayNet: A Large Human-Annotated Dataset for Document-Layout Analysis. (2022). doi:10.1145/3534678.353904

[9] Nicholas Pipitone and Ghita Houir Alami. 2024. Legalbench-rag: A benchmark for retrieval-augmented generation in the legal domain. *arXiv preprint arXiv:2408.10343* (2024).

[10] Monica Riedler and Stefan Langer. 2024. Beyond Text: Optimizing RAG with Multimodal Inputs for Industrial Applications. *arXiv preprint arXiv:2410.21943* (2024).

[11] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389.

[12] Spurthi Setty, Harsh Thakkar, Alyssa Lee, Eden Chung, and Natan Vidra. 2024. Improving retrieval for rag based question answering models on financial documents. *arXiv preprint arXiv:2404.07221* (2024).

[13] Ray Smith. 2007. An overview of the Tesseract OCR engine. In *Ninth international conference on document analysis and recognition (ICDAR 2007)*, Vol. 2. IEEE, 629–633.

[14] Navve Wasserman, Roi Pony, Oshri Naparstek, Adi Raz Goldfarb, Eli Schwartz, Udi Barzelay, and Leonid Karlinsky. 2025. REAL-MM-RAG: A Real-World Multi-Modal Retrieval Benchmark. *arXiv preprint arXiv:2502.12342* (2025).

[15] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. 2019. Detectron2. https://github.com/facebookresearch/detectron2.

[16] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115* (2024).

[17] Antonio Jimeno Yepes, Yao You, Jan Milczek, Sebastian Laverde, and Renyu Li. 2024. Financial report chunking for effective retrieval augmented generation. *arXiv preprint arXiv:2402.05131* (2024).

[18] Worawut Yimyam, Mahasak Ketcham, Tanapon Jensuttiwetchakult, Sansanee Hiranchan, Patiyuth Pramkeaw, and Narumol Chumuang. 2020. Enhancing and Evaluating an Impact of OCR and Ontology on Financial Document Checking Process. In *2020 15th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*. IEEE, 1–6.

[19] Junyuan Zhang, Qintong Zhang, Bin Wang, Linke Ouyang, Zichen Wen, Ying Li, Ka-Ho Chow, Conghui He, and Wentao Zhang. 2024. OCR Hinders RAG: Evaluating the Cascading Impact of OCR on Retrieval-Augmented Generation. *arXiv preprint arXiv:2412.02592* (2024).

[20] Ruochen Zhao, Hailin Chen, Weishi Wang, Fangkai Jiao, Xuan Long Do, Chengwei Qin, Bosheng Ding, Xiaobao Guo, Minzhi Li, Xingxuan Li, et al. 2023. Retrieving multimodal information for augmented generation: A survey. *arXiv preprint arXiv:2303.10868* (2023).