VISRAG: VISION-BASED RETRIEVAL-AUGMENTED GENERATION ON MULTI-MODALITY DOCUMENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Retrieval-augmented generation (RAG) is an effective technique that enables large language models (LLMs) to utilize external knowledge sources for generation. However, current RAG systems are solely based on text, rendering it impossible to utilize vision information like layout and images that play crucial roles in real-world multi-modality documents. In this paper, we introduce VisRAG, which tackles this issue by establishing a vision-language model (VLM)-based RAG pipeline. In this pipeline, instead of first parsing the document to obtain text, the document is directly embedded using a VLM as an image and then retrieved to enhance the generation of a VLM. Compared to traditional text-based RAG, VisRAG maximizes the retention and utilization of the data information in the original documents, eliminating the information loss introduced during the parsing process. We collect both open-source and synthetic data to train the retriever in VisRAG and explore a variety of generation methods. Experiments demonstrate that VisRAG outperforms traditional RAG in both the retrieval and generation stages, achieving a 25–39% end-to-end performance gain over traditional text-based RAG pipeline. Further analysis reveals that VisRAG is effective in utilizing training data and demonstrates strong generalization capability, positioning it as a promising solution for RAG on multi-modality documents. Our code and data will be made publicly available.

1 INTRODUCTION

Trained on massive data, large language models (LLMs) like GPT-4 (Achiam et al., 2023) have shown strong abilities in common NLP tasks using their parametric knowledge (Wei et al., 2022; Zhao et al., 2023). However, the issue of hallucination (Ji et al., 2023; Bang et al., 2023) and the challenge of updating the parametric knowledge limit their real-world application in specific domains. Retrieval-augmented generation (RAG) alleviates this problem by using a knowledge retriever, which has access to a custom outer knowledge base, to supply the LLM with the necessary information for generating outputs (Guu et al., 2020; Lewis et al., 2020; Yu et al., 2023). Open-source RAG frameworks like llamaindex (Liu, 2022) have been developed to facilitate the research and deployment of common RAG pipelines.

Typical retrieval-augmented generation (RAG) pipelines are *text-based*, operating on segmented texts as retrieval units (Yu et al., 2023; Asai et al., 2024; Yan et al., 2024), which we refer to as TextRAG. In real-world scenarios, knowledge is often presented in multi-modality documents such as textbooks and manuals which may have texts and figures intersected together. To acquire texts from such data sources, a *parsing* stage is often employed, which typically involves a cascade of processes, including layout recognition, optical character recognition (OCR), and post-processing steps like text joining (Zhang et al., 2024). While effective in most scenarios, the parsing process inevitably introduces errors, which can negatively impact the retrieval and generation phases. Moreover, text-based RAG utilizes only textual information, overlooking potential information present in other modalities like images. Although research has been conducted on image retrieval and multi-modal RAG, these approaches primarily focus on predefined scenarios wherein images and descriptive texts are properly extracted and paired (Wei et al., 2023; Sharifymoghaddam et al., 2024; Zhou et al., 2024), differing from real-world scenarios where texts and images (including figures) are often interleaved within a single document page.

The recent development of vision-language models (VLMs) has introduced a promising approach to understanding complex visual cues in images and documents (OpenBMB, 2024b; Wang et al., 2024). A VLM typically integrates a vision encoder, a language model, and a connector that bridges the two, showing superior abilities in applications such as describing pictures (Alayrac et al., 2022), explaining figures (Bavishi et al., 2023), and transcribing (printed and handwritten) text from document images (Laurençon et al., 2024). Given the robust capabilities of VLMs in capturing multi-modal information present in images, an intriguing question arises: can the basic language model in the retrieval and generation components of TextRAG be substituted with a VLM, thus the parsing stage is bypassed and all the information of the document is preserved?

In this paper, we present **Vis**ion-based **R**etrieval-**a**ugmented **G**eneration (VisRAG), to study the feasibility of building a pure-vision RAG pipeline using VLMs. VisRAG is built with a VLM-based retriever VisRAG-Ret and generator VisRAG-Gen. Inherited the bi-encoder of text-based dense retriever (Karpukhin et al., 2020), VisRAG-Ret maps the query and the document into an embedding space, but utilizing the document's image directly instead of relying on extracted textual content. The embedding is obtained by applying weighted mean pooling on the final hidden states of the input text or vision tokens. After retrieving top-k document images, VisRAG processes these images to generate the answer. While it is straightforward to use a VLM that supports multi-image input for generation, for VLMs that can only accept one single image, we propose page concatenation and weighted selection techniques to enable the handling of multiple documents. Throughout the process, VisRAG preserves all information in its original visual format, thereby preventing the potential information loss or distortion that might occur in traditional RAG pipelines.

To evaluate VisRAG on real-world multi-modal documents, we construct datasets from open-source visual question answering (VQA) datasets and synthetic query-document pairs derived from web-crawled PDFs. In terms of retrieval, VisRAG-Ret exhibits superior performance in retrieving multi-modal documents. It outperforms state-of-the-art text- and vision-centric retrievers and achieves better results than solely relying on its constituent vision encoder or language model under identical training conditions. For generation, VisRAG-Gen surpasses traditional text-based generators with open-source VLMs. With GPT-4o, capable of handling multiple images, VisRAG shows increasing performance gains with more retrieved documents, indicating the potential for improved multi-page reasoning in the future. As depicted in Figure 1, in a direct comparison of pipeline performances, VisRAG achieves a 39% relative improvement over TextRAG using MiniCPM-V 2.6 as the generator and a 25% relative improvement with GPT-4o as the generator, attributed to the cascade effect. Further analysis reveals that VisRAG possesses better training data efficiency and generalization ability than baseline models, and demonstrates robustness across both text-centric and vision-centric documents. VisRAG shows great promise in replacing TextRAG as the next-generation standard for RAG pipelines.

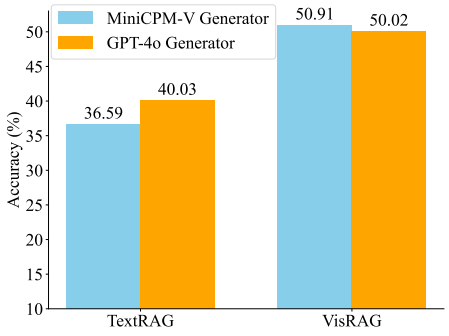


Figure 1: TextRAG vs. VisRAG on final generation accuracy. In TextRAG, parsed text serves as the basis for both retrieval and generation processes. In contrast, VisRAG leverages the original document image directly by using a VLM-based retriever and generator. Details can be found in Sec. 5.1.

2 Related Work

Retrieval-augmented Generation (RAG). RAG enhances large language models (LLMs) by incorporating retrieved information from external knowledge bases, which assists in addressing knowledge-intensive tasks (Guu et al., 2020), reducing hallucinations (Semnani et al., 2023), and acquiring new knowledge (Vu et al., 2023). An RAG pipeline typically comprises a text-based retriever that fetches relevant information from the knowledge base given the user query, and an LLM-based generator that reads the query along with the retrieved information to generate an answer (Shi et al., 2024b; Yu et al., 2023). Prior research on RAG primarily focuses on: a) improving the retriever, which is typically a text encoder producing text embeddings, through generator feed-

back (Yu et al., 2023; Shi et al., 2024b); b) enhancing the generator via supervised fine-tuning (Lin et al., 2024; Xu et al., 2024a), in-context pre-training (Shi et al., 2024a), or advanced prompting (Xu et al., 2024c); and c) developing advanced RAG pipelines to handle long-form or multi-hop question answering (Jiang et al., 2023; Asai et al., 2024). However, research on RAG has predominantly targeted cleaned text corpora like Wikipedia from an academic standpoint. Building effective RAG pipelines for real-world, multi-modal documents remains a challenge.

Vision-language Models. Recent advancements in vision-language models (VLMs) have greatly improved fine-grained multi-modal understanding. Since CLIP (Radford et al., 2021) pioneered contrastive visual-text alignment, models like Flamingo (Alayrac et al., 2022), LLaVA (Liu et al., 2023b), and BLIP (Li et al., 2022) have expanded LLMs to process visual inputs by connecting languages models with a CLIP-style vision encoder. Research has then shifted towards more advanced multi-task and multi-stage pre-training paradigms, enabling models to generalize across a wide range of vision-language tasks (Liu et al., 2024a; Bai et al., 2023; Wang et al., 2023; Dai et al., 2023). This is followed by notable advancements in high-resolution visual understanding (Xu et al., 2024b; Bavishi et al., 2023; Lin et al., 2023) and OCR capabilities (Kim et al., 2022; Lee et al., 2023; Hong et al., 2024; Chen et al., 2024b). Specifically, VLMs like the DocOwl series (Ye et al., 2023a; Hu et al., 2024b;a), UReader (Ye et al., 2023b), and TextMonkey (Liu et al., 2024b) are purpose-built to tackle OCR-free document understanding. More recently, breakthroughs have been made in multi-image understanding (Li et al., 2024a; Wang et al., 2024). Recent open-source VLMs like the MiniCPM-V (Yao et al., 2024) and Qwen2-VL (Wang et al., 2024) series combine the merits of recent techniques, achieving state-of-the-art performance. Those features of VLMs provide a foundation for our vision-based RAG pipeline, which requires multi-modal document understanding.

Multi-modality Retrieval and RAG. Multi-modal retrieval encompasses a wide range of tasks, such as retrieving a matching image given the text (Han et al., 2017), retrieving a text-image pair to answer a question (Chang et al., 2022), and retrieving texts that answer the given query about a provided image (Hu et al., 2023a; Luo et al., 2023), etc. Wei et al. (2023) propose UniIR, a universal multi-modal retrieval model capable of addressing the aforementioned multiple tasks. The retrieved information is then employed for incorporating knowledge (Hu et al., 2023b; Luo et al., 2021) or in-context learning (Tan et al., 2024; Liu et al., 2023a), with the aim of generating answers or images (Sharifymoghaddam et al., 2024). Prior research mentioned above is conducted on academic datasets, where texts and images are meticulously extracted from raw data and paired (e.g., images with their captions), to make it feasible to do separate encoding of data in different modalities. This hinders their applicability in real-world RAG scenarios, as real-world multi-modal documents are often presented in mixed modalities, and information may be distributed across various combinations of modalities. Concurrent works DSE (Ma et al., 2024) and ColPali (Faysse et al., 2024) address this issue by directly encoding the image of a document for retrieval. However, as these studies focus on retrieval, they lack a comprehensive comparison of their approaches with text-based retrieval in both in-domain and out-of-domain settings, and do not conduct an end-to-end RAG evaluation.

3 METHODOLOGY

In this section, we first recap the typical RAG pipeline (Sec. 3.1), then present our VisRAG framework (Sec. 3.2) and the construction of our training and evaluation data (Sec. 3.3).

3.1 PRELIMINARY: RETRIEVAL-AUGMENTED GENERATION

A typical retrieval-augmented generation (RAG) pipeline consists of a retriever and a generator, both built on large language models (LLMs)[1]. This pipeline operates on a knowledge corpus $\mathcal{D}$, which is processed into units for retrieval and generation, denoted as $\mathcal{D} = \{d_1, \ldots, d_n\}$, where n is the number of retrieval units. Given a text query q and the retrieval corpus $\mathcal{D}$, the retriever functions as $\mathcal{R} : (q, \mathcal{D}) \rightarrow \mathcal{D}_{\mathcal{R}}$, taking q and $\mathcal{D}$ as inputs and producing a candidate set $\mathcal{D}_{\mathcal{R}} \subset \mathcal{D}$. To enable efficient search, the units in the knowledge corpus $\mathcal{D}$ are pre-encoded into embeddings. During RAG pipeline inference, approximate nearest neighbor (ANN) search is applied to retrieve

[1]In many cases, the retriever uses language models smaller than 1B parameters, which may not be considered "large", but we use the term LLM for simplicity.

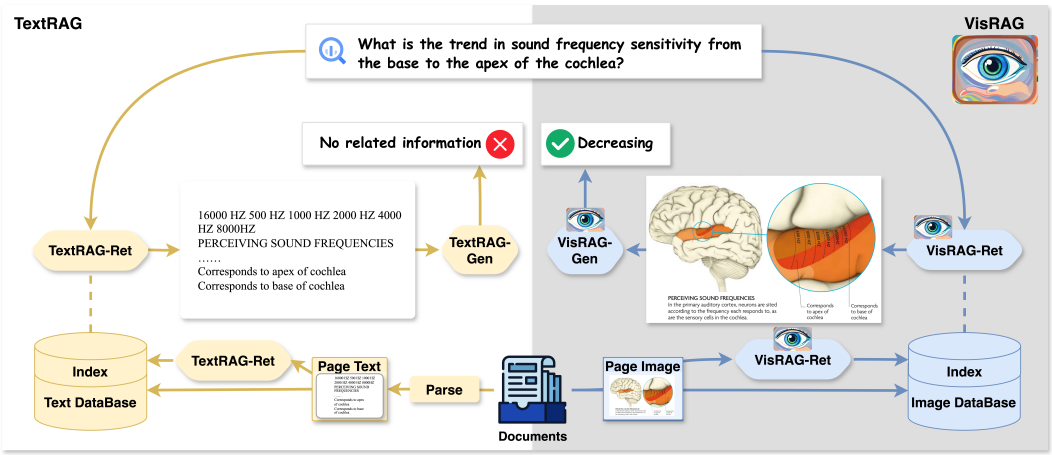


Figure 2: TextRAG (left) vs. VisRAG (right). Traditional text-based RAG (TextRAG) relies on parsed texts for retrieval and generation, losing visual information in multi-modal documents. Our vision-based RAG (VisRAG) employs a VLM-based retriever and generator to directly process the document page's image, thereby preserving all information in the original page.

$\mathcal{D}_{\mathcal{R}}$, which serves as the knowledge source for generation. The generation process can be defined as a function $\mathcal{G} : (q, \mathcal{D}_{\mathcal{R}}) \rightarrow a$, where a represents the answer and $\mathcal{G}$ denotes the LLM generator. This is achieved by prompting the LLM with the query and the retrieved units $\mathcal{D}_{\mathcal{R}}$ to generate an answer.

As shown in Figure 2 (left), traditional RAG frameworks (TextRAG) typically utilize text-based units for retrieval and generation. However, in real-world scenarios, data often appear in complex, multi-modal documents, requiring an additional parsing step to obtain text. In this paper, we propose to use the *page* as the fundamental unit for retrieval and generation, which is directly processed by vision language models (VLMs) as an image without further processing during retrieval and generation. In subsequent sections, we use the terms "page" and "document" interchangeably.

3.2 VisRAG: Vision-based Retrieval-augmented Generation

In this section, we present **Vis**ion-based **R**etrieval-**a**ugmented **G**eneration (VisRAG), as shown in Figure 2 (right). In contrast to traditional RAG frameworks which use text segments for both retrieval and generation, VisRAG leverages the image of the document to preserve all information.

3.2.1 Retrieval

The first stage of VisRAG, VisRAG-Ret, aims to retrieve a set of pages from the corpus $\mathcal{D}$ given q. We follow the dual-encoder paradigm in text-based dense retrieval models (Karpukhin et al., 2020) but employ a VLM rather than an LLM to encode the query and page. Specifically, the query and page are encoded separately as text and image in the VLM, producing in a sequence of hidden states. To derive the final embedding, and given that we use generative VLMs with causual attention, we adopt the position-weighted mean pooling over the last-layer VLM hidden states (Muennighoff, 2022), giving higher weights to later tokens:

$$\mathbf{v} = \sum_{i=1}^{S} w_i \mathbf{h}_i, \tag{1}$$

where $\mathbf{h}_i$ is the i-th hidden state, S is the sequence length, $w_i = \frac{i}{\sum_{j=1}^{S} j}$ is the i-th weight, and $\mathbf{v}$ is the query or page embedding. The similarity score is calculated by the cosine similarity of the query and page embedding. VisRAG-Ret is optimized using the InfoNCE loss:

$$l(q, d^+, D^-) = -\log \frac{\exp(s(q, d^+)/\tau)}{\exp(s(q, d^+)/\tau) + \sum_{d^- \in D^-} \exp(s(q, d^-)/\tau)}, \tag{2}$$

where d^+, D^- are positive document and the negative document set of q, respectively, $s(q, d)$ is the similarity score between q and d, and τ is the temperature.

3.2.2 GENERATION

The second stage of VisRAG, VisRAG-Gen, focuses on generating the answer according to the user query and retrieved pages using a VLM. We propose the following mechanisms to enable VisRAG-Gen to handle multiple retrieved pages in $\mathcal{D}_\mathcal{R}$ for generation. The prompts used for generation is presented in Appendix E.

Page Concatenation. A straightforward approach is to concatenate all pages in $\mathcal{D}_\mathcal{R}$ into a single image to accommodate most VLMs that are trained to accept a single image. Formally,

$$a \leftarrow \text{VLM-Single}(q, \text{Concat}(\{d | d \in \mathcal{D}_\mathcal{R}\})), \tag{3}$$

where VLM-Single is a VLM that accepts a single image with text prompt and Concat is the image concatenation operation. In this paper, we experiment with horizontal concatenation.

Weighted Selection. Another approach is to ask the VLM to generate an answer for every page from top-k, and select a final one with the highest confidence (Lewis et al., 2020; Shi et al., 2024b). The final confidence is defined as the weighted generation probability of the answer:

$$P(a|q, \mathcal{D}_\mathcal{R}) = P(a|q, d) \cdot \lambda(q, d), \tag{4}$$

where $P(a|d, q)$ is calculated as the reciprocal of the perplexity of generating the answer a conditioned on the single document d, and $\lambda(d, q)$ is the normalized retrieval score:

$$\lambda(q, d) = \frac{e^{s(q,d)}}{\sum_{d' \in \mathcal{D}_\mathcal{R}} e^{s(q,d')}}. \tag{5}$$

VLMs Accepting Multiple Images. Some recent VLMs like MiniCPM-V 2.6 (OpenBMB, 2024b) and Qwen-VL 2 (Wang et al., 2024) are designed and trained to accept multiple images as input to perform cross-image reasoning. This capability may be useful for the generation as the required information could be located on a single page from the retrieved document set $\mathcal{D}_\mathcal{R}$ for single-hop questions or spread across multiple pages for multi-hop questions. Formally, we have

$$a \leftarrow \text{VLM-Multi}(q, \{d | d \in \mathcal{D}_\mathcal{R}\}), \tag{6}$$

where VLM-Multi is the VLM that accepts multiple images with text prompt.

3.3 DATA CONSTRUCTION

To effectively build and evaluate RAG pipelines on multi-modal documents, we construct our datasets using a combination of visual question answering (VQA) datasets and synthetic data. The statistics of our constructed dataset are provided in Table 1.

Data Sources. We collect question-document pairs from a series of VQA datasets, targeting different document types: MP-DocVQA (Tito et al., 2023) for industrial documents, ArXivQA (Li et al., 2024b), ChartQA (Masry et al., 2022), InfographicsVQA (Mathew et al., 2022), and PlotQA (Methani et al., 2020) for various figure types, and SlideVQA (Tanaka et al., 2023) for presentation slides. All datasets feature questions that can be answered using a single document (page), except for SlideVQA, which includes multi-hop questions requiring information from multiple pages. We follow the original datasets' train-test splits, except for MP-DocVQA and InfographicsVQA, where the validation split serves as our evaluation set. Additionally, we enhance our training set by collecting openly available PDFs from online sources and generating queries using GPT-4o (OpenAI, 2024), with details presented in Appendix A.1. We assemble the retrieval corpus by gathering the positive document associated with each query from the training and evaluation sets.

Query Filtering. Some queries extracted from VQA datasets are *context-dependent*, which lack specificity to a certain entity. For instance, the response to "Where was <u>the conference</u> held?" varies based on the contextual document. Using such context-dependent queries in open retrieval tasks is ineffective because they lack strong document specificity. To address this, we implement an additional filtering stage to remove these context-dependent questions, where we prompt llama-3-8b-instruct (AI@Meta, 2024) with human-annotated in-context samples to generate the classification label. Table 1 shows a substantial reduction in context-dependent questions across data sources. The details of filtering are presented in Appendix A.2.

Table 1: Dataset statistics. We collect data from visual question answering (VQA) datasets for training and evaluation and synthetic additional query-document pairs for training. We apply filtering on VQA datasets to remove context-dependent queries that are not suitable for retrieval.

Source	Document Type	% Filtered	Train	Evaluation		
			# Q-D Pairs	# Q	# D	# Pos. D per Q
ArXivQA (2024b)	Arxiv Figures	14%	25,856	8,640	8,066	1.00
ChartQA (2022)	Charts	41%	4,224	718	500	1.00
MP-DocVQA (2023)	Industrial Documents	69%	10,624	1,879	741	1.00
InfoVQA (2022)	Infographics	26%	17,664	2,046	459	1.00
PlotQA (2020)	Scientific Plots	44%	56,192	11,307	9,593	1.00
SlideVQA (2023)	Slide Decks	22%	8,192	1,640	1,284	1.34
Synthetic	Various	-	239,358	-	-	-

Evaluation Metrics. We report retrieval and generation performance on the evaluation sets of datasets sourced from VQA datasets. For retrieval, we use MRR@10 and Recall@10 as the metrics. For generation, consistent with methods applied to the source datasets, we report the answer accuracy, employing a relaxed exact match metric which allows a 5% error margin for numeric responses (Masry et al., 2022; Methani et al., 2020).

4 Experimental Methodology

In this section, we introduce our setup for experiments. Descriptions of the LLMs/VLMs used in our experiments can be found in Appendix C.

Document Parsing. To assess the performance of VisRAG in comparison to TextRAG, we employ specific text extraction methods. The first approach, referred to as "(OCR)" in subsequent text, is a pipeline that initially leverages PPOCR (Du et al., 2020) to identify text regions, then combines vertically aligned and horizontally proximate text boxes to reduce fragmentation. The second approach, termed "(Captioner)", is an end-to-end model-based method. In this approach, we apply MiniCPM-V 2.0 (OpenBMB, 2024a; Yao et al., 2024), fine-tuned on paired (document image, extracted text) data, to directly parse text from the document image. Details of the parsing processes are presented in Appendix B.

Retrieval Experiments. VisRAG-Ret is a document embedding model built on MiniCPM-V 2.0, a vision-language model that integrates SigLIP (Zhai et al., 2023) as the vision encoder and MiniCPM (Hu et al., 2024c) as the language model. To ensure fair comparisons, we organize experiments into three settings: off-the-shelf, out-of-domain, and in-domain, as depicted below.

- Off-the-shelf: We directly evaluate popular text and image retrieval models on extracted texts, including BM25 (OCR), a lexical model; bge-large-en-v1.5 (Xiao et al., 2023) (OCR) and NV-Embed-v2 (Lee et al., 2024) (OCR), state-of-the-art text embedding models with sizes 335M and 7.85B, respectively; and SigLIP, a CLIP-style (Radford et al., 2021) vision model serving as the encoder for MiniCPM-V series.
- Out-of-domain: Models in this category are trained solely on synthetic data and evaluated on the VQA datasets, lacking in-domain supervision, in order to show the models' generalization capabilities. These models include textual models MiniCPM (OCR), MiniCPM (Captioner), and vision model SigLIP. MiniCPM (OCR) and (Captioner) are MiniCPM-based text embedding models trained and evaluated on extracted text.
- In-domain: Models in this category are trained on the blend of the VQA training data and synthetic data. We evaluate the same set of models as in the out-of-domain setting to show model performance when supervised labels are available. We also report the performance of ColPali (Faysse et al., 2024) on our evaluation data. ColPali is a page embedding model that encodes a screenshot of a page into multiple vectors. We train ColPali on our dataset using the official code and hyper-parameters provided in its paper.

We report VisRAG-Ret's performance in both out-of-domain and in-domain settings.

Table 2: Overall retrieval performance in MRR@10. The best retrieval performance in each group is marked in **bold**, and the second best performance is underlined. We train ColPali (Faysse et al., 2024) on our dataset. Corresponding Recall@10 performance can be found in Table 6.

Model	# Para.	ArxivQA	ChartQA	DocVQA	InfoVQA	PlotQA	SlideVQA	Average
(a) Off-the-shelf Models								
BM25 (OCR)	n.a.	32.30	41.64	71.92	63.99	35.32	84.57	54.96
bge-large (2023) (OCR)	335M	27.63	39.92	50.21	67.01	30.95	79.94	49.28
NV-Embed-v2 (2024) (OCR)	7.85B	**45.64**	**52.58**	**73.01**	**81.70**	**36.10**	**91.73**	**63.46**
SigLIP (2023)	883M	30.56	49.38	49.70	63.77	21.90	81.17	49.41
(b) Out-of-domain: *Models Fine-tuned on Synthetic Data*								
MiniCPM (OCR)	2.72B	37.19	43.34	68.57	75.36	21.93	86.80	55.53
MiniCPM (Captioner)	2.72B	34.59	46.65	64.25	71.84	18.85	82.96	53.19
SigLIP (2023)	883M	37.01	49.34	58.32	63.55	21.23	85.32	52.46
VisRAG-Ret	3.43B	**59.03**	**51.18**	**73.28**	**81.17**	**28.70**	**90.38**	**63.96**
(c) In-domain: *Models Fine-tuned on Synthetic and In-domain data*								
MiniCPM (OCR)	2.72B	47.36	54.43	74.13	80.11	**41.33**	90.49	64.64
MiniCPM (Captioner)	2.72B	47.53	54.28	68.87	76.46	35.93	85.45	61.42
SigLIP (2023)	883M	50.22	**61.44**	66.01	72.87	40.15	89.51	63.37
ColPali (2024)	2.92B	64.61	59.18	**82.17**	79.19	33.49	**93.08**	68.62
VisRAG-Ret	3.43B	**67.00**	59.34	77.65	**84.05**	40.26	91.71	**70.00**

Generation Experiments. To evaluate generation performance, we fix the retrieval model to VisRAG-Ret and report the performance of various generation models and methods. For VisRAG-Gen, we compare the performance of the single-image VLM MiniCPM-V 2.0, which only accepts a single image, against the multi-image VLM MiniCPM-V 2.6 (OpenBMB, 2024b; Yao et al., 2024) and GPT-4o (OpenAI, 2024). MiniCPM-V 2.6 is an upgrade of MiniCPM-V 2.0, incorporating Qwen2-7B (Yang et al., 2024) as the language model and supporting multi-image input. We evaluate the performance of page concatenation and weighted selection on the single-image VLM. Additionally, we report the performance of text-based generation baselines, including MiniCPM (OCR) and GPT-4o (OCR), where only extracted texts are used for generation. For all experiments, we report results using the top-1, top-2, and top-3 retrieved documents, as well as an "Oracle" condition where the model is provided with only the positive document(s) to show the performance upper bound.

Implementation Details. VisRAG-Ret is fine-tuned using in-batch negatives (Karpukhin et al., 2020) for one epoch with a batch size of 128 on 8 NVIDIA A100 80GB GPUs. The temperature parameter in Equation 2 is set to 0.02. Baseline retrievers are fine-tuned with the same hyper-parameters, and textual baselines utilize extracted text data as document-side input. The generation part does not use any fine-tuning; we directly use off-the-shelf LLMs/VLMs for generation.

5 Evaluation Results

5.1 Overall Performance

Retrieval Performance. In this experiment, we compare VisRAG-Ret with (a) off-the-shelf models, and trained baselines in (b) out-of-domain setting where we only leverage synthetic data, and in (c) in-domain setting where we leverage both in-domain and synthetic training data.

As shown in Table 2(a)(b), VisRAG-Ret, trained on out-of-domain data, outperforms all off-the-shelf baselines, including both text and vision models. It significantly outperforms both BM25 and bge-large, and surpasses NV-Embed-v2, a state-of-the-art text retrieval model with 7.85B parameters. Note that bge-large and NV-Embed-v2 are trained on millions of query-doc pairs (Xiao et al., 2023; Lee et al., 2024), which are 10x more than our training data. Although bge-large outperforms BM25 on benchmarks like MTEB (Muennighoff et al., 2023), it fails on our datasets, indicating text-based embedding models trained on clean text struggle with texts parsed from real-world documents.

When trained with the same data setup, as demonstrated in Table 2(b)(c), VisRAG-Ret outperforms text models MiniCPM (OCR) & (Captioner) and the vision model SigLIP by a significant margin. The advantage is more pronounced in the out-of-domain setting, where VisRAG-Ret achieves 15% and 22% gains over MiniCPM (OCR) and SigLIP, respectively, compared to 8% and 10% in the in-domain setting. This indicates that VisRAG-Ret has better generalization capability compared to text- and vision-centric models. Notably, despite utilizing the same VLM MiniCPM-V 2.0 for

Table 3: Overall generation performance in accuracy (%). Different generation models and methods utilize the same retriever, VisRAG. Performance relative to Oracle (using the ground-truth document(s) for generation) is colored in blue.

Model / Method	Input	ArxivQA	ChartQA	DocVQA	InfoVQA	PlotQA	SlideVQA	Average
(a) TextRAG-Gen: *Text-based Generation*								
MiniCPM (OCR)	top-1	42.87 (96.0%)	24.79 (83.6%)	27.99 (79.1%)	16.86 (88.5%)	21.97 (97.8%)	25.43 (91.2%)	26.65 (89.3%)
	top-2	42.57 (95.4%)	25.07 (84.5%)	28.21 (79.7%)	16.32 (85.6%)	22.18 (98.8%)	26.83 (96.3%)	26.86 (90.0%)
	top-3	42.63 (95.5%)	24.65 (83.1%)	25.49 (72.0%)	15.30 (80.3%)	21.52 (95.8%)	27.38 (98.2%)	26.16 (87.7%)
	Oracle	44.64 (100%)	29.67 (100%)	35.39 (100%)	19.06 (100%)	22.46 (100%)	27.87 (100%)	29.85 (100%)
GPT-4o (OCR)	top-1	58.51 (94.8%)	39.69 (65.8%)	44.86 (77.1%)	37.34 (84.5%)	25.70 (94.4%)	41.95 (83.0%)	41.34 (82.1%)
	top-2	58.00 (94.0%)	41.36 (68.6%)	47.74 (82.1%)	39.00 (88.3%)	18.42 (67.6%)	45.85 (90.7%)	41.73 (82.9%)
	top-3	58.54 (94.8%)	40.25 (66.7%)	48.43 (83.3%)	39.69 (89.8%)	17.87 (65.6%)	45.12 (89.3%)	41.65 (82.7%)
	Oracle	61.72 (100%)	60.31 (100%)	58.17 (100%)	44.18 (100%)	27.24 (100%)	50.55 (100%)	50.36 (100%)
(b) VisRAG-Gen: *Single-image VLM (MiniCPM-V 2.0)*								
Page Concatenation	top-1	56.82 (94.5%)	28.97 (76.8%)	36.83 (76.1%)	24.29 (85.7%)	25.95 (84.1%)	33.78 (88.5%)	34.44 (84.8%)
	top-2	56.22 (93.5%)	25.63 (67.9%)	27.83 (57.5%)	18.96 (66.9%)	25.73 (83.4%)	31.40 (82.3%)	30.96 (76.3%)
	top-3	55.49 (92.3%)	25.35 (67.2%)	20.81 (43.0%)	16.42 (57.9%)	24.37 (79.0%)	29.27 (76.7%)	28.62 (70.5%)
	Oracle	60.10 (100%)	37.74 (100%)	48.38 (100%)	28.35 (100%)	30.84 (100%)	38.17 (100%)	40.60 (100%)
Weighted Selection	top-1	56.82 (94.5%)	28.97 (76.8%)	36.83 (76.1%)	24.29 (85.7%)	25.95 (84.1%)	33.78 (86.3%)	34.44 (84.5%)
	top-2	56.67 (94.3%)	29.25 (77.5%)	37.15 (76.8%)	24.24 (85.5%)	25.98 (84.3%)	34.33 (87.7%)	34.60 (84.9%)
	top-3	57.12 (95.0%)	29.53 (78.2%)	35.76 (73.9%)	24.00 (84.7%)	28.53 (92.5%)	34.57 (88.3%)	34.92 (85.7%)
	Oracle	60.10 (100%)	37.74 (100%)	48.38 (100%)	28.35 (100%)	30.84 (100%)	39.15 (100%)	40.76 (100%)
(c) VisRAG-Gen: *Multi-image VLM*								
MiniCPM-V 2.6	top-1	65.30 (90.4%)	41.50 (63.1%)	61.36 (74.8%)	46.63 (83.9%)	44.22 (74.3%)	46.46 (81.8%)	50.91 (77.9%)
	top-2	65.14 (90.1%)	41.23 (62.7%)	65.57 (80.0%)	46.29 (83.3%)	35.24 (59.2%)	48.60 (85.5%)	50.34 (77.1%)
	top-3	65.45 (90.6%)	42.76 (65.0%)	66.05 (80.5%)	45.45 (81.8%)	37.67 (63.3%)	49.45 (87.0%)	51.14 (78.3%)
	Oracle	72.27 (100%)	65.74 (100%)	82.01 (100%)	55.57 (100%)	59.51 (100%)	56.83 (100%)	65.32 (100%)
GPT-4o	top-1	62.94 (93.1%)	43.45 (66.0%)	59.50 (75.9%)	54.01 (83.0%)	29.74 (75.8%)	50.49 (80.5%)	50.02 (79.2%)
	top-2	62.19 (92.0%)	43.59 (66.2%)	63.54 (81.0%)	56.99 (87.5%)	26.42 (67.4%)	56.04 (89.3%)	51.46 (81.5%)
	top-3	62.26 (92.1%)	46.10 (70.0%)	64.93 (82.8%)	56.45 (86.7%)	27.26 (69.5%)	56.34 (89.8%)	52.22 (82.7%)
	Oracle	67.58 (100%)	65.88 (100%)	78.45 (100%)	65.10 (100%)	39.22 (100%)	62.74 (100%)	63.16 (100%)

parsing, MiniCPM (Captioner) performs worse than VisRAG-Ret, indicating that directly encoding with VLMs works better than using VLMs for parsing. This can be attributed to the inevitable information loss when multi-modality information is transcribed into text.

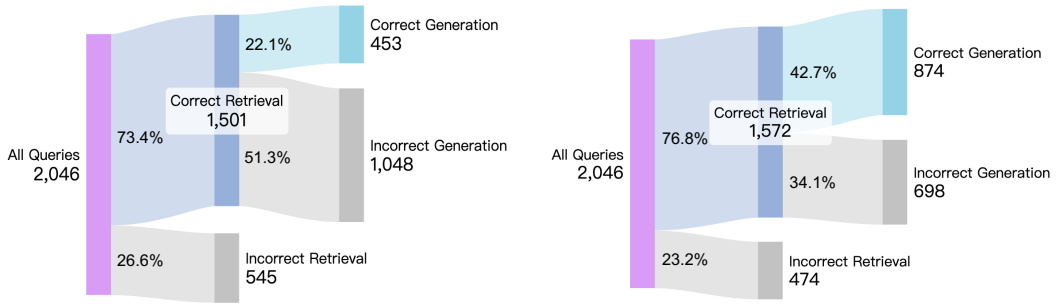
Further analysis reveals that MiniCPM (OCR) and SigLIP perform differently across datasets: SigLIP excels in ArxivQA and ChartQA, while MiniCPM (OCR) significantly outperforms SigLIP in DocVQA and InfographicsVQA. This may be due to the different focuses of the two models: MiniCPM focuses on text, while SigLIP focuses on visual signals. VisRAG-Ret, built on top of MiniCPM-V 2.0, with a SigLIP encoder and a MiniCPM language model, combines the merits of both and performs well across all datasets, capturing more holistic information from a document.

Compared to ColPali, a multi-vector document page embedding model, VisRAG-Ret not only maintains superior performance but also achieves much better memory efficiency. ColPali represents a page with 256KB of data distributed across 1030 128-dim vectors (Faysse et al., 2024), whereas VisRAG-Ret uses just 4.5KB in a single 2304-dimensional vector. This makes VisRAG-Ret more suitable for scaling to millions or billions of documents in real-world applications.

Generation Performance. In this experiment, we apply a series of text- and vision-based generators and methods on top of the same retriever VisRAG-Ret to study their effectiveness in generating the answer given the query and retrieved documents. Table 3 shows the performance of (a) text-based generation (TextRAG-Gen), (b) generation using the VLM MiniCPM-V 2.0 which only accepts a single image as input, and (c) generation using VLMs which accept multiple images as input.

When models are provided with only the ground-truth documents ("Oracle"), VisRAG-Gen models, which process the document image directly, significantly outperform RAG-Gen models, which rely solely on extracted text. For instance, MiniCPM-V 2.0 achieves 36% higher performance than MiniCPM (OCR) when using ground-truth documents. This underscores the importance of visual clues in extracting answers from documents and indicates that VisRAG-Gen has a higher performance upper bound than TextRAG-Gen.

In practical scenarios where models receive the top-1 to 3 retrieved documents, which may include noise, VisRAG-Gen consistently outperforms TextRAG-Gen within the same model series. Specifically, for MiniCPM-V 2.0, capable of processing only a single image, the weighted selection approach demonstrates better performance than page concatenation when handling 2 or 3 retrieved documents. Simple concatenation may overwhelm the VLM with unnecessary information, while



(a) TextRAG with MiniCPM (OCR) as the retriever and MiniCPM-V 2.6 (OCR) as the generator.

(b) VisRAG with VisRAG-Ret as the retriever and MiniCPM-V 2.6 as the generator.

Figure 3: Pipeline performance of (a) TextRAG and (b) VisRAG on InfographicsVQA. We visualize the portion of queries that have the positive document retrieved at the top-1 position ("Correct Retrieval"), and that are answered correctly given the top-1 retrieved document ("Correct Generation").

weighted selection filters answers based on multiple VLM outputs conditioned on individual documents, thus reducing the information burden.

TextRAG pipelines usually benefit from an increased number of retrieved documents due to better information coverage (Zhu et al., 2024). However, while weighted selection enhances robustness in performance, there is no significant performance boost with a higher count of retrieved documents using this approach. Notably, only the most advanced VLMs, such as GPT-4o, which handle multiple images, show a marked performance increase as the number of retrieved documents rises. This suggests that reasoning over multiple images remains a challenging task for current VLMs.

End-to-end Performance. In this experiment, we study the effectiveness of the VisRAG *pipeline*, by comparing it with the TextRAG pipeline. We construct TextRAG using MiniCPM (OCR) and MiniCPM-V 2.6 (OCR) for retrieval and generation, respectively, and VisRAG using VisRAG-Ret for retrieval and MiniCPM-V 2.6 for generation. The performance on InfographicsVQA is visually represented in Figure 3. Notebly, VisRAG achieves a higher rate of accurately retrieving documents than TextRAG, and demonstrates a significantly improved rate of correct answer generation from accurately retrieved documents. The cumulative improvements in both retrieval and generation phases result in an overall accuracy increment from 22.1% to 42.7%. Across the six evaluation datasets, VisRAG shows a 39% relative accuracy increment on average, as illustrated in Figure 1. The case study of VisRAG and TextRAG is presented in Appendix F.

5.2 Training Data Efficiency

As retrieval acts as the bottleneck in an RAG pipeline, it is crucial to have an effective retrieval component to maintain optimal performance. In this experiment, we study the training data efficiency of VisRAG-Ret by evaluating the performance of VisRAG-Ret trained under different amounts of synthetic training data, i.e. in the out-of-domain setting. As shown in Figure 4, when only trained on 20k q-d pairs, VisRAG can surpass bge-large (OCR). After training on 150k pairs, it can further surpass NV-Embed-v2 (OCR), the SOTA 8B-sized text embedding model trained on millions of curated text pairs. This highlights VisRAG-Ret's high training data efficiency and strong generalization capability, as all models are evaluated out-of-domain. When compared with MiniCPM (OCR), which uses extracted text for training, VisRAG-Ret consistently achieves

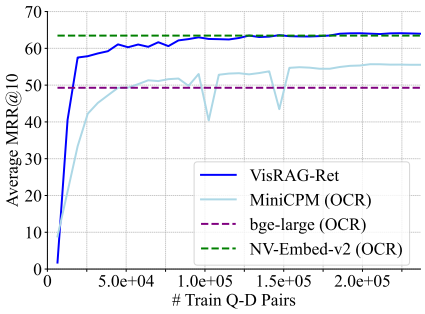


Figure 4: Average retrieval performance of VisRAG-Ret vs. MiniCPM (OCR) trained with different numbers of training examples.

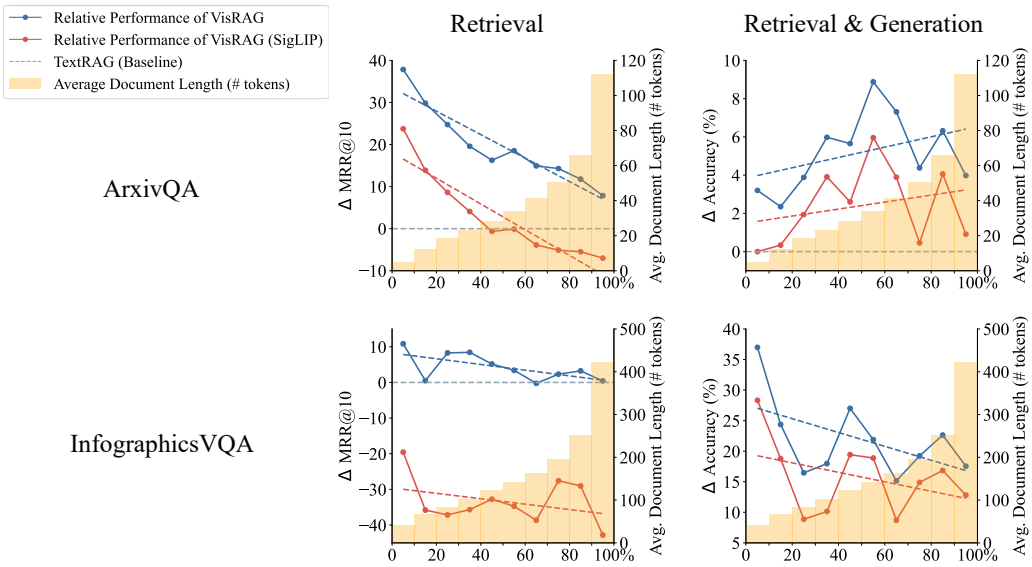


Figure 5: Relative retrieval and generation performance of VisRAG, VisRAG (SigLIP), and TextRAG on different subsets of queries. The X-axes represent the query subsets where the lengths of the positive documents fall within specific percentile ranges. For comparative analysis, we set TextRAG's performance to zero and show the performance differences of other models from TextRAG.

a performance gain of about 17% and exhibits a more stable training process. The results show VisRAG-Ret's potential for further performance improvements by scaling up the training data.

5.3 Performance on Different Data Subsets

In this experiment, we assess the retrieval and generation performance of VisRAG and TextRAG defined in Figure 3, as well as VisRAG (SigLIP), which replaces the retriever in VisRAG with SigLIP. We report their performance across different data subsets by categorizing queries based on the lengths of their positive documents, measured by the number of tokens of the extracted text. Documents with a higher volume of extracted text may prioritize textual information over visual content. As illustrated in Figure 5, queries in ArxivQA and InfographicsVQA are divided into equal-sized bins according to the lengths of their relevant documents. For each bin, we calculate and plot the average performance differences between VisRAG and TextRAG, as well as between VisRAG (SigLIP) and TextRAG, to compare how each model performs relative to TextRAG. We observe that, in general, the relative performance of VisRAG and VisRAG (SigLIP) improves as the length of the relevant document decreases. This suggests that models with vision encoders can better understand documents that emphasize visual information. However, VisRAG (SigLIP) consistently underperforms VisRAG across all data subsets and, in some cases, even performs worse than TextRAG. In contrast, VisRAG consistently outperforms TextRAG, indicating that the underlying language model in VisRAG is crucial for better understanding the semantics conveyed through visual cues.

6 Conclusion

In this paper, we propose VisRAG, a novel retrieval-augmented generation (RAG) paradigm that utilizes vision-language models (VLMs) to facilitate retrieval and generation within an RAG pipeline, thereby eliminating the parsing stage required in traditional text-based RAG. Our empirical results demonstrate that VisRAG consistently outperforms text-based RAG on retrieval and generation while maintaining a simpler pipeline. We hope that VisRAG will inspire future RAG development to incorporate VLMs for handling multi-modal documents.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

AI@Meta. Llama 3 model card. 2024. URL https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *Proceedings of NeurIPS*, volume 35, pp. 23716–23736, 2022.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *Proceedings of ICLR*, 2024.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. In *Proceedings of AACL/IJCNLP 2023*, pp. 675–718, 2023.

Rohan Bavishi, Erich Elsen, Curtis Hawthorne, Maxwell Nye, Augustus Odena, Arushi Somani, and Sağnak Taşırlar. Introducing our multimodal models, 2023. URL https://www.adept.ai/blog/fuyu-8b.

Yingshan Chang, Mridu Narang, Hisami Suzuki, Guihong Cao, Jianfeng Gao, and Yonatan Bisk. Webqa: Multihop and multimodal qa. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16495–16504, 2022.

Guiming Hardy Chen, Shunian Chen, Ruifei Zhang, Junying Chen, Xiangbo Wu, Zhiyi Zhang, Zhihong Chen, Jianquan Li, Xiang Wan, and Benyou Wang. Allava: Harnessing gpt4v-synthesized data for a lite vision-language model. *arXiv preprint arXiv:2402.11684*, 2024a.

Wentong Chen, Junbo Cui, Jinyi Hu, Yujia Qin, Junjie Fang, Yue Zhao, Chongyi Wang, Jun Liu, Guirong Chen, Yupeng Huo, et al. Guicourse: From general vision language models to versatile gui agents. *arXiv preprint arXiv:2406.11317*, 2024b.

Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of SIGIR*, 2009.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven C. H. Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. In *Proceedings of NeurIPS*, 2023.

Yuning Du, Chenxia Li, Ruoyu Guo, Xiaoting Yin, Weiwei Liu, Jun Zhou, Yifan Bai, Zilin Yu, Yehua Yang, Qingqing Dang, and Haoshuang Wang. Pp-OCR: A Practical Ultra Lightweight OCR System. *arXiv*, abs/2009.09941, 2020.

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

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Retrieval augmented language model pre-training. In *Proceedings of ICML*, pp. 3929–3938, 2020.

Xintong Han, Zuxuan Wu, Phoenix X Huang, Xiao Zhang, Menglong Zhu, Yuan Li, Yang Zhao, and Larry S Davis. Automatic spatially-aware fashion concept discovery. In *Proceedings of ICCV*, pp. 1463–1471, 2017.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. Cogagent: A visual language model for gui agents. In *Proceedings of CVPR*, pp. 14281–14290, 2024.

Anwen Hu, Haiyang Xu, Jiabo Ye, Ming Yan, Liang Zhang, Bo Zhang, Ji Zhang, Qin Jin, Fei Huang, and Jingren Zhou. mplug-docowl 1.5: Unified structure learning for ocr-free document understanding. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of EMNLP*, pp. 3096–3120, 2024a.

Anwen Hu, Haiyang Xu, Liang Zhang, Jiabo Ye, Ming Yan, Ji Zhang, Qin Jin, Fei Huang, and Jingren Zhou. mplug-docowl2: High-resolution compressing for ocr-free multi-page document understanding. *arXiv preprint arXiv:2409.03420*, 2024b.

Hexiang Hu, Yi Luan, Yang Chen, Urvashi Khandelwal, Mandar Joshi, Kenton Lee, Kristina Toutanova, and Ming-Wei Chang. Open-domain visual entity recognition: Towards recognizing millions of wikipedia entities. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12065–12075, 2023a.

Shengding Hu, Yuge Tu, Xu Han, Ganqu Cui, Chaoqun He, Weilin Zhao, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Xinrong Zhang, Zhen Leng Thai, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, dahai li, Zhiyuan Liu, and Maosong Sun. Minicpm: Unveiling the Potential of Small Language Models with Scalable Training Strategies. In *First Conference on Language Modeling*, volume abs/2404.06395, 2024c.

Ziniu Hu, Ahmet Iscen, Chen Sun, Zirui Wang, Kai-Wei Chang, Yizhou Sun, Cordelia Schmid, David A Ross, and Alireza Fathi. Reveal: Retrieval-augmented visual-language pre-training with multi-source multimodal knowledge memory. In *Proceedings of CVPR*, pp. 23369–23379, 2023b.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, (12):248:1–248:38, 2023.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of EMNLP*, pp. 7969–7992, 2023.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of EMNLP*, pp. 6769–6781, 2020.

Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. Ocr-free document understanding transformer. In *Proceedings of ECCV*, pp. 498–517. Springer, 2022.

Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models? *arXiv preprint arXiv:2405.02246*, 2024.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Nv-Embed: Improved Techniques for Training LLMs as Generalist Embedding Models. *arXiv*, abs/2405.17428, 2024.

Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *Proceedings of ICML*, pp. 18893–18912, 2023.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of NeurIPS*, 2020.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024a.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *Proceedings of ICML*, pp. 12888–12900, 2022.

Lei Li, Yuqi Wang, Runxin Xu, Peiyi Wang, Xiachong Feng, Lingpeng Kong, and Qi Liu. Multimodal ArXiv: A Dataset for Improving Scientific Comprehension of Large Vision-Language Models. In *Proceedings of ACL*, pp. 14369–14387, 2024b.

Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Richard James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, et al. Ra-dit: Retrieval-augmented dual instruction tuning. In *Proceedings of ICLR*, 2024.

Ziyi Lin, Chris Liu, Renrui Zhang, Peng Gao, Longtian Qiu, Han Xiao, Han Qiu, Chen Lin, Wenqi Shao, Keqin Chen, et al. Sphinx: The joint mixing of weights, tasks, and visual embeddings for multi-modal large language models. *arXiv preprint arXiv:2311.07575*, 2023.

Bingshuai Liu, Chenyang Lyu, Zijun Min, Zhanyu Wang, Jinsong Su, and Longyue Wang. Retrieval-augmented multi-modal chain-of-thoughts reasoning for large language models. *arXiv preprint arXiv:2312.01714*, 2023a.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Proceedings of NeurIPS*, volume 36, pp. 34892–34916, 2023b.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of CVPR*, pp. 26296–26306, 2024a.

Jerry Liu. LlamaIndex, 11 2022. URL `https://github.com/jerryjliu/llama_index`.

Yuliang Liu, Biao Yang, Qiang Liu, Zhang Li, Zhiyin Ma, Shuo Zhang, and Xiang Bai. Textmonkey: An ocr-free large multimodal model for understanding document. *arXiv preprint arXiv:2403.04473*, 2024b.

Man Luo, Yankai Zeng, Pratyay Banerjee, and Chitta Baral. Weakly-supervised visual-retriever-reader for knowledge-based question answering. In *Proceedings of EMNLP*, pp. 6417–6431, 2021.

Man Luo, Zhiyuan Fang, Tejas Gokhale, Yezhou Yang, and Chitta Baral. End-to-end knowledge retrieval with multi-modal queries. In *Proceedings of ACL*, pp. 8573–8589, 2023.

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

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq R. Joty, and Enamul Hoque. Chartqa: A Benchmark for Question Answering about Charts with Visual and Logical Reasoning. In *Proceedings of ACL*, pp. 2263–2279, 2022.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V. Jawahar. Infographicvqa. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2582–2591, 2022.

Nitesh Methani, Pritha Ganguly, Mitesh M. Khapra, and Pratyush Kumar. Plotqa: Reasoning over scientific plots. In *The IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2020.

Niklas Muennighoff. Sgpt: Gpt sentence embeddings for semantic search. *arXiv preprint arXiv:2202.08904*, 2022.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. Mteb: Massive text embedding benchmark. In *Proceedings of EACL*, pp. 2014–2037, 2023.

OpenAI. Hello, gpt-4o — openai, 2024. URL https://openai.com/index/hello-gpt-4o/.

OpenBMB. openbmb/minicpm-v-2, 2024a. URL https://huggingface.co/openbmb/MiniCPM-V-2.

OpenBMB. openbmb/minicpm-v-2_6, 2024b. URL https://huggingface.co/openbmb/MiniCPM-V-2_6.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of ICML*, pp. 8748–8763, 2021.

Sina J Semnani, Violet Z Yao, Heidi C Zhang, and Monica S Lam. Wikichat: A few-shot llm-based chatbot grounded with wikipedia. *arXiv preprint arXiv:2305.14292*, 2023.

Sahel Sharifymoghaddam, Shivani Upadhyay, Wenhu Chen, and Jimmy Lin. Unirag: Universal retrieval augmentation for multi-modal large language models. *arXiv preprint arXiv:2405.10311*, 2024.

Weijia Shi, Sewon Min, Maria Lomeli, Chunting Zhou, Margaret Li, Xi Victoria Lin, Noah A Smith, Luke Zettlemoyer, Wen-tau Yih, and Mike Lewis. In-context pretraining: Language modeling beyond document boundaries. In *Proceedings of ICLR*, 2024a.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. Replug: Retrieval-augmented black-box language models. In *Proceedings of NAACL*, pp. 8364–8377, 2024b.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of CVPR*, pp. 8317–8326, 2019.

Cheng Tan, Jingxuan Wei, Linzhuang Sun, Zhangyang Gao, Siyuan Li, Bihui Yu, Ruifeng Guo, and Stan Z Li. Retrieval meets reasoning: Even high-school textbook knowledge benefits multimodal reasoning. *arXiv preprint arXiv:2405.20834*, 2024.

Ryota Tanaka, Kyosuke Nishida, Kosuke Nishida, Taku Hasegawa, Itsumi Saito, and Kuniko Saito. Slidevqa: A Dataset for Document Visual Question Answering on Multiple Images. In *Proceedings of AAAI*, pp. 13636–13645, 2023.

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

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Proceedings of NeurIPS Datasets and Benchmarks Track*, 2021.

Rubèn Tito, Dimosthenis Karatzas, and Ernest Valveny. Hierarchical multimodal transformers for Multipage DocVQA. *Pattern Recognition*, 144:109834, 2023.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, et al. Freshllms: Refreshing large language models with search engine augmentation. *arXiv preprint arXiv:2310.03214*, 2023.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution, 2024.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*, 2023.

Cong Wei, Yang Chen, Haonan Chen, Hexiang Hu, Ge Zhang, Jie Fu, Alan Ritter, and Wenhu Chen. Uniir: Training and benchmarking universal multimodal information retrievers. *arXiv preprint arXiv:2311.17136*, 2023.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *TMLR*, 2022.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-pack: Packaged resources to advance general chinese embedding. *arXiv preprint arXiv:2309.07597*, 2023.

Peng Xu, Wei Ping, Xianchao Wu, Zihan Liu, Mohammad Shoeybi, and Bryan Catanzaro. Chatqa 2: Bridging the gap to proprietary llms in long context and rag capabilities. *arXiv preprint arXiv:2407.14482*, 2024a.

Ruyi Xu, Yuan Yao, Zonghao Guo, Junbo Cui, Zanlin Ni, Chunjiang Ge, Tat-Seng Chua, Zhiyuan Liu, Maosong Sun, and Gao Huang. Llava-uhd: an lmm perceiving any aspect ratio and high-resolution images. *arXiv preprint arXiv:2403.11703*, 2024b.

Zhipeng Xu, Zhenghao Liu, Yibin Liu, Chenyan Xiong, Yukun Yan, Shuo Wang, Shi Yu, Zhiyuan Liu, and Ge Yu. Activerag: Revealing the treasures of knowledge via active learning. *arXiv preprint arXiv:2402.13547*, 2024c.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*, 2024.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, Qianyu Chen, Huarong Zhou, Zhensheng Zou, Haoye Zhang, Shengding Hu, Zhi Zheng, Jie Zhou, Jie Cai, Xu Han, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. Minicpm-V: A GPT-4v Level MLLM on Your Phone. *arXiv*, abs/2408.01800, 2024.

Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Yuhao Dan, Chenlin Zhao, Guohai Xu, Chenliang Li, Junfeng Tian, et al. mplug-docowl: Modularized multimodal large language model for document understanding. *arXiv preprint arXiv:2307.02499*, 2023a.

Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Guohai Xu, Chenliang Li, Junfeng Tian, Qi Qian, Ji Zhang, et al. Ureader: Universal ocr-free visually-situated language understanding with multimodal large language model. In *Findings of EMNLP*, pp. 2841–2858, 2023b.

Zichun Yu, Chenyan Xiong, Shi Yu, and Zhiyuan Liu. Augmentation-adapted retriever improves generalization of language models as generic plug-in. In *Proceedings of ACL*, pp. 2421–2436, 2023.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid Loss for Language Image Pre-Training. In *Proceedings of ICCV*, pp. 11941–11952, 2023.

Ge Zhang, Scott Qu, Jiaheng Liu, Chenchen Zhang, Chenghua Lin, Chou Leuang Yu, Danny Pan, Esther Cheng, Jie Liu, Qunshu Lin, et al. Map-neo: Highly capable and transparent bilingual large language model series. *arXiv preprint arXiv:2405.19327*, 2024.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *ArXiv preprint*, 2023.

Tianshuo Zhou, Sen Mei, Xinze Li, Zhenghao Liu, Chenyan Xiong, Zhiyuan Liu, Yu Gu, and Ge Yu. MARVEL: unlocking the multi-modal capability of dense retrieval via visual module plugin. In *Proceedings of ACL*, pp. 14608–14624, 2024.

Kunlun Zhu, Yifan Luo, Dingling Xu, Ruobing Wang, Shi Yu, Shuo Wang, Yukun Yan, Zhenghao Liu, Xu Han, Zhiyuan Liu, et al. Rageval: Scenario specific rag evaluation dataset generation framework. *arXiv preprint arXiv:2408.01262*, 2024.

A DATA CONSTRUCTION DETAILS

A.1 SYNTHETIC DATA

Table 4: Statistics of crawled documents. We prompt GPT-4o to generate queries on these documents.

Name	Source	Description	# Pages
Textbooks	https://openstax.org/	College-level textbooks including various subjects	10,000
ICML Papers	ICML 2023	ICML papers on various topics	5,000
NeurIPS Papers	NeurIPS 2023	NeurIPS papers on various topics	5,000
Manuallib	https://www.manualslib.com/	Manuals of various kinds of products	20,000

To augment the training dataset of VisRAG, we gather additional documents from the web and utilize GPT-4o to generate queries based on these documents. The sources of the collected documents are listed in Table 4. The prompt employed is shown in Figure 6.

```
Hello, I have a super rich document library. Assume you are a curious but very ignorant
      human. You often ask me questions (queries) to seek a precise document as a
     reference for your question or request.
- Now, you have received another task:
    - Here is a document image. This is a reference (target) that I provided from the
     rich document library based on your query. Your task now is to imagine various
     different angles of questions that I might ask.
### Your goal is to accurately find this document target as a potential reference
     document candidate through queries in a very rich document library.
### The questions I ask might need references from the text, images, charts, or
     implicit meanings in the document.
### Maximum number of query-answer pairs is 6.

Below is your output format:
```json
{
    "result":[
        {
            "answer": "",
            "query" : ""
        },
        {
            "answer": "",
            "query" : ""
        },
    ...
        {
            "answer": "",
            "query" : ""
        }
    ]
}
```
{{ document }}
```

Figure 6: Prompt for GPT-4o to generate queries, where {{ `document` }} is the document page.

A.2 QUERY FILTERING

As mentioned in Sec. 3.3, a significant portion of queries in VQA datasets are context-dependent that are unsuitable for retrieval. We prompt llama-3-8b-instruct (AI@Meta, 2024) to filter out such queries using the prompt in Figure 7, which includes human-annotated samples from DocVQA.

B DOCUMENT PARSING

In this paper, we experiment with two categories of document parsing strategies: pipeline-based parsing and model-based parsing.

```
I have some QA data here, and you can observe that the questions can be divided into
     two categories:

The category #A: When you see this question alone without a given document, you are
     sure to find a unique document in a corpus to provide a unique answer.

The category #B: When you see this question alone without a given document, you will
     find hard to locate a document to give a deterministic answer for this question,
     because you will find multiple candidate documents in a corpus, which may lead to
     different answers for this question.

The number mentioned on the right of the leftside margin? #B
What is the date mentioned in the second table? #B
What is the full form of PUF? #A
What is the number at the bottom of the page, in bold? #B
Who presented the results on cabin air quality study in commercial aircraft? #A
What is the name of the corporation? #B
To whom this is addressed? #B
How many one-on-one interviews were completed during April 10th through the April 12th?
      #A
What is the subject of the document/letter? #B
Who sent the letter? #B
Heading of the document? #B
What is the slope mentioned in the first table? #B
Where were the two magnesium containing papers made at? #A
what is the date in the letter? #B
What is the date mentioned in the letter? #B
Which part of Virginia is this letter sent from? #B
who were bothered by cigarette odors? #A
which cigarette would be better if offered on a thicker cigarette? #A
Cigarettes will be produced and submitted to O/C Panel for what purpose? #A
What is the heading of first table? #B
What is RIP-6 value for KOOL KS? #A
Which hetero-atoms does polar compounds contain? #A
One variable that has implicitly not been controlled? #B
Which corporation's letterhead is this? #B
what is the contact person name mentioned in letter? #B
what is the date mentioned in this letter? #B
Another model of the 83mm with zero ventilation will be made at Semiworks within how
     many weeks? #A
Hand sheets were made utilizing a 30% level of which component? #A
What is the source? #B
What is the heading of the document? #B
What is the subject? #B
Which test is used to evaluate ART menthol levels that has been shipped? #A
How much percent had not noticed any difference in the odor of VSSS? #A
What is the cigarette code of RIP-6(W/O Filter) 21/4SE? #A
What is the meeting date? #B
what is the subject of this letter? #B
what is the index for Retention of Franchise? #B
What is the heading of second table? #B
What is the full form of POVC? #A
what mm Marlboro Menthol were subjectively smoked by the Richmond Panel? #A
What sort of communication/letter is this? #B
According to the listed requirements, what must be the age group of female smokers? #A
How many one-on-one interviews were completed during April 10th through the April 12th?
      #A
During the process of prototype production and ringtipping, some cigarettes were
     observed to have burn holed in which paper? #A
How many distinct mechanisms appear to play a role in the breakup of a smoke column
     into a multi-dimensional flowfield? #A
Where was the conference held? #B
Who is in cc in this letter? #B
Under BOLD, primary production of Blend #24- will be completed by which date? #A
{{ query }} #
```

Figure 7: Prompt for llama3-8b-instruct to classify queries, where {{ query }} is the query to be classified. Label B denotes context-dependent queries.

B.1 PIPELINE-BASED PARSING

We consider the following document parsing pipelines:

Pytesseract. Pytesseract is a Python wrapper for Google's Tesseract OCR engine, offering a straightforward interface for text extraction from images. Unlike more complex methods, Pytesseract requires minimal pre-processing. By invoking the `image_to_string` function, OCR is performed in a single step, directly returning the extracted text. Tesseract internally handles bounding boxes, confidence scores, and orientation correction.

PPOCR-based Methods. PaddlePaddle OCR (PPOCR) (Du et al., 2020) is widely used for document text extraction, covering text detection, classification, and recognition. First, a text detection model identifies text regions and generates bounding boxes. These regions are then processed by a classification model to correct orientation issues like rotation or flipping. Next, a recognition model extracts the textual content from the corrected bounding boxes, returning recognized text with confidence scores. Only results with confidence scores above 0.6 are retained, and the bounding box coordinates, along with the recognized text, are stored for further processing. We apply the following strategies to obtain the final parsing result:

- Adjacent Merging: To enhance text coherence, this policy combines adjacent text boxes based on vertical proximity (within 15 pixels) and horizontal alignment (within 100 pixels), reducing text fragmentation. This iterative merging process consolidates eligible text boxes into unified bounding boxes with concatenated text. Finally, the text from the remaining bounding boxes is combined with line breaks to produce the final result.
- Layout Preserving: This policy maintains the original document structure by ordering text boxes based on their spatial positions. Spaces and line breaks are dynamically inserted to reflect horizontal and vertical gaps between text regions. This approach ensures that the extracted text mirrors the original document layout, preserving its formatting in the final result.

We run the aforementioned pipelines on our dataset to obtain text-based training and evaluation data, and fine-tune a MiniCPM retriever to assess performance. The results are presented in Table 5. Methods based on PPOCR demonstrate significantly better performance compared to pytesseract, with adjacent merging and layout preserving yielding similar results. Consequently, we opt to use the adjacent merging policy for our "(OCR)" runs.

Table 5: Overall retrieval performance of different document parsing pipelines.

	ArxivQA	**ChartQA**	**DocVQA**	**InfoVQA**	**PlotQA**	**SlideVQA**	**Average**
(c) In-domain: *Models Fine-tuned on Synthetic and In-domain data*							
MiniCPM (Pytesseract)	33.70	49.69	70.43	74.07	36.11	80.40	57.40
MiniCPM (Adjacent Merging)	**47.36**	54.43	**74.13**	80.11	**41.33**	**90.49**	**64.64**
MiniCPM (Layout Preserving)	45.26	**55.78**	73.75	**80.22**	40.97	90.22	64.37

B.2 MODEL-BASED PARSING

In addition to pipeline-based methods, we also employ a model-based parsing approach using MiniCPM-V 2.0 to directly transcribe document images into text. This method is referred to as "(Captioner)".

To train this model, we collect data from two sources: a) ALLaVA (Chen et al., 2024a) (image, caption) pairs, and b) VQA documents with descriptions generated by GPT-4V. We use the prompt in Figure 8 to instruct GPT-4V to generate detailed descriptions of documents from DocVQA, ChartQA, SlideVQA, InfographicsVQA, TextVQA (Singh et al., 2019), and ArxivQA.

We train MiniCPM-V 2.0 with a batch size of 2048 and a learning rate of 5e-6 for 1 epoch.

```
Based on the layout information, output the text in the image. Try not to modify the
     text, but you need to indicate the structure such as title, body text, subtitle,
     table, etc.
Note:
If there are charts or graphs, they should be described in detail.
If you feel that there are more than 4000 words or most of the text in the image is
     unclear or most of the text contents in the image are not written in English, then
      directly return <none>.
{{ document }}
```

Figure 8: Prompt for GPT-4V to generate page description, where {{ `document` }} is the document page.

C MODELS USED IN THIS PAPER

MiniCPM (Hu et al., 2024c) is a large language model (LLM) with 2.4 billion non-embedding parameters, demonstrating capabilities comparable to much larger models, such as Llama2-7B (Touvron et al., 2023) and Gemma-7B (Team et al., 2024). In this paper, we employ MiniCPM to construct the baseline text-based retriever (Table 2) and generator (Table 3).

SigLIP (Zhai et al., 2023) is a CLIP-style multi-modal model designed to align text and vision representations. We utilize SigLIP-400m, released by Hugging Face[2], which incorporates Flash Attention 2, increases maximum resolution to 980x980, and adopts the NaViT strategy to allow (a) variable resolution images and (b) aspect ratio preserved images. In this paper, SigLIP is used to develop the baseline vision-based retriever (Table 2).

MiniCPM-V 2.0 (OpenBMB, 2024a; Yao et al., 2024) is a vision-language model (VLM) with 2.8 billion non-embedding parameters, built upon SigLIP-400m and MiniCPM. It can process single images up to 1.8 million pixels (e.g., 1344x1344) at any aspect ratio. We use MiniCPM-V 2.0 to build VisRAG-Ret (Table 2) and VisRAG-Gen (Table 3(b)), as well as the document parsing model.

MiniCPM-V 2.6 (OpenBMB, 2024b; Yao et al., 2024) is an upgrade of MiniCPM-V 2.0 and MiniCPM-Llama3-V 2.5 (Yao et al., 2024). It is built upon SigLIP-400M and Qwen2-7B (Yang et al., 2024) with a total of 8.5B parameters, exihibiting a significant performance improvement over MiniCPM-Llama3-V 2.5 (Yao et al., 2024). Different from previous models, MiniCPM-V 2.6 can accept multiple images as the input and perform multi-modal in-context learning. It also demonstrates stronger OCR capabilities. We use MiniCPM-V 2.6 to build VisRAG-Gen (Table 3) and a text-based generation baseline MiniCPM-V 2.6 (OCR) (Figure 3, Figure 5).

Note that, MiniCPM-Llama3-V 2.5 (Yao et al., 2024) is not used in this paper.

GPT-4o (OpenAI, 2024) is OpenAI's latest multi-modal model, capable of processing any combination of text, audio, image, and video inputs and generating outputs in text, audio, and image formats. We use GPT-4o to construct VisRAG-Gen (Table 3) and to synthesize training data.

D RETRIEVAL PERFORMANCE IN RECALL@10

Table 6 presents the retrieval performance in Recall@10.

E PROMPTS FOR GENERATION

We present the prompts of VisRAG-Gen and TextRAG-Gen in Table 7.

[2]https://huggingface.co/HuggingFaceM4/siglip-so400m-14-980-flash-attn2-navit

Table 6: Overall retrieval performance in Recall@10.

Model	# Para.	ArxivQA	ChartQA	DocVQA	InfoVQA	PlotQA	SlideVQA	Average
(a) Off-the-shelf Models								
BM25 (OCR)	n.a.	42.22	56.69	84.67	79.77	50.51	89.34	67.20
bge-large (2023) (OCR)	335M	35.84	49.58	68.71	84.65	46.65	88.35	62.30
NV-Embed-v2 (2024) (OCR)	7.85B	**56.11**	**65.32**	**89.52**	**93.89**	**52.42**	**96.54**	**75.63**
SigLIP (2023)	883M	42.27	62.12	71.95	83.63	40.28	91.33	65.26
(b) Out-of-domain: *Models Fine-tuned on Synthetic Data*								
MiniCPM (OCR)	2.72B	47.06	56.69	86.06	90.13	39.52	93.74	68.87
MiniCPM (Captioner)	2.72B	47.27	60.86	81.32	88.91	37.11	91.25	67.79
SigLIP (2023)	883M	49.51	63.65	78.61	82.16	41.36	93.03	68.05
VisRAG-Ret	3.43B	**72.52**	**64.76**	**90.10**	**95.21**	**47.42**	**95.71**	**77.62**
(c) In-domain: *Models Fine-tuned on Synthetic and In-domain data*								
MiniCPM (OCR)	2.72B	59.21	69.22	89.36	92.72	61.94	95.74	78.03
MiniCPM (Captioner)	2.72B	60.81	67.83	85.68	91.25	58.42	93.50	76.25
SigLIP (2023)	883M	67.12	**75.91**	84.62	90.81	**62.04**	95.00	79.25
ColPali (2024)	2.92B	76.46	70.19	**96.06**	93.65	48.62	96.97	80.33
VisRAG-Ret	3.43B	**80.41**	72.98	92.97	**96.33**	61.47	**97.03**	**83.53**

Table 7: Prompt templates for generation. "Others" refers to all VQA datasets except ArxivQA.

	TextRAG	VisRAG
ArxivQA	Hint: {{ parsed document(s) }} Question: {{ query }} Options: A. {{ Option 1 }} B. {{ Option 2 }} C. {{ Option 3 }} D. {{ Option 4 }} Answer directly with the letter of the correct option as the first character.	{{ document(s) }} Question: {query }} Options: A. {{ Option 1 }} B. {{ Option 2 }} C. {{ Option 3 }} D. {{ Option 4 }} Answer directly with the letter of the correct option as the first character.
Others	Image:{{ parsed document(s) }} Answer the question using a single word or phrase. Question:{{ query }} Answer:	{{ document(s) }} Answer the question using a single word or phrase. Question:{{ query }} Answer:

F CASE STUDY

We show two cases in Table 8 and Table 9. In both instances, we compare VisRAG with TextRAG, maintaining the same setup as described in the "End-to-end Performance" paragraph in Sec. 5.1.

In the first case from DocVQA, the user queries about "Club Jetty," however, the term "Club Jetty" in the relevant document is not successfully extracted due to its decorative font. This leads to TextRAG failing to retrieve the document, while VisRAG successfully retrieves it.

In the second case from InfographicsVQA, although both TextRAG and VisRAG successfully retrieve the document, TextRAG generates an incorrect response due to the loss of layout information, making it unclear which number (53% or 49%) pertains to Europe. VisRAG effectively utilizes the layout information and generates the correct answer.

Table 8: Case study from DocVQA. In this case, VisRAG successfully retrieves the ground-truth document, while TextRAG fails, leading to VisRAG's correct generation and TextRAG's incorrect generation.

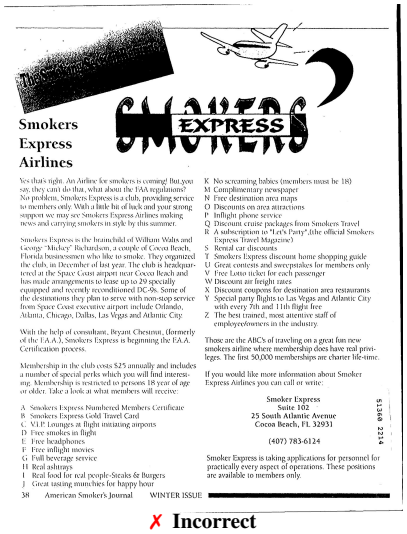
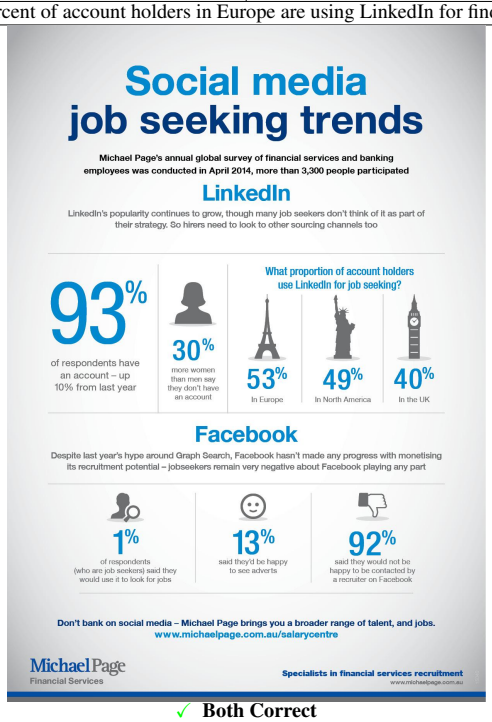
	TextRAG	VisRAG
Query	On which day is Club Jetty closed?	
Retrieved Top-1 Document	 ✗ **Incorrect**	✓ **Correct**
Document Parsing Result	SMOKERS↩ EXPRESS↩ Express↩ Airlines↩ Yes that's right. An Airline for↩ smokers is coming! But you↩ say, they can't do that, what about↩ the FAA regulations?↩ No problem. Smokers Express is↩ a club, providing service↩ to members only: With a little bit↩ of luck and your strong↩ support we may see Smokers↩ Express Airlines making↩ news and carrying smokers↩ in style by this summer.↩ K No screaming babies↩ (members must be 18)↩ M Complimentary newspaper↩ N Free destination area maps↩ O Discounts on area attractions↩ p Inflight phone service↩ Q Discount cruise packages↩ from Smokers Travel↩ R A subscription to "Let's Party"↩ the official Smokers↩ Smokers Express is the brainchild↩ of William Walts and↩ George "Mickey" Richardson, a↩ couple of Cocoa Beach,↩ Florida businessmen who like to↩ smoke. They organized↩ the club, in December of last year.↩ The club is headquartered↩ at the Space Coast airport↩ near Cocoa Beach and↩ has made arrangements to lease↩ up to 29 specially equipped↩ and recently reconditioned DC-9s.↩ Some of the destinations they↩ plan to serve with non-stop service↩ from Space Coast executive airport↩ include Orlando, Atlanta, Chicago,↩ Dallas, Las Vegas, and Atlantic City↩ (Express Travel Magazine)↩ S Rental car discounts↩ T Smokers Express discount home↩ shopping guide↩ U Great contests and sweepstakes↩ for members only↩ V Free Lotto ticket for each passenger↩ W Discount air freight rates↩ X Discount coupons for destination↩ area restaurants↩ Y Special party flights to Las Vegas↩ and Atlantic City with every 7th and↩ 11th flight free↩ Z The best trained, most attentive↩ staff of employee/owners↩ in the industry.↩ With the help of consultant,↩ Bryant Chestnut (formerly of the↩ FAA), Smokers Express is↩ beginning the FAA↩ Certification process.↩ Those are the ABC's of traveling↩ on a great fun new↩ smokers airline where membership↩ does have real privileges.↩ The first 50,000 memberships are↩ charter life-time.↩ Membership in the club costs↩ $25 annually and includes↩ a number of special perks↩ which you will find interesting.↩ Membership is restricted↩ to persons 18 years of age↩ or older. Take a look at↩ what members will receive:↩ If you would like more↩ information about Smokers↩ Express Airlines you can call or↩ write:↩ Smokers Express↩ Suite 102↩ 25 South Atlantic Avenue↩ Cocoa Beach, FL 32931↩ (407) 783-6124↩ A Smokers Express Numbered↩ Members Certificate↩ B Smokers Express Gold Travel↩ Card↩ C V.I.P. Lounges at flight initiating↩ airports↩ D Free smokes in flight↩ E Free headphones↩ F Free inflight movies↩ G Full beverage service↩ H Real ashtrays↩ Smoker Express is taking↩ applications for personnel↩ for practically every aspect of↩ operations. These positions↩ are available to members only.↩ t Real food for real people—Steaks↩ & Burgers↩ Great tasting munchies for happy↩ hour.↩ American Smoker's Journal↩ 38 WINTER ISSUE	FXPLOREKAUAI↩ (We mail gift paks)↩ Windsurfing↩ KAUAIWINDSURFING↩ EXPERIENCEIS↩ NOW OPEN↩ Learn to Windsurf↩ (certified instruction)↩ Special introductory↩ Lesson Rate↩ on your way↩ fresh↩ from the roaster↩ fern grotto↩ WAILUA↩ MARINA↩ RESTAURANT↩ On the banks of the Wailua River↩ to you↩ COFFEE↩ & NUT↩ ROASTING↩ CENTER↩ "HOME STYLE COOKING"↩ famous baked stuffed pork chops↩ and 28 other entrees↩ EASY LEARNING↩ EXCURSIONS↩ RENTALS↩ Phone: 245-9290↩ or Kauai Surf ext. 7830↩ The Market Place-shop 39↩ at the Coconut Plantation↩ Waipouli, Kauai↩ coffee tea nuts spices herbs↩ Complimentary transportation↩ (from Wailua area Hotels-dinner only)↩ Phone: 822-4311↩ NOW! lunch daily from 11 a.m.↩ PAPERBACK↩ HUT↩ Hi, my name is Sunny ...↩ and I own one of the most↩ unique restaurants in the world↩ in Lihue, Kauai.↩ It's called the Casa Blanca,↩ and we offer Kauai's only late↩ gourmet dining service in a very↩ friendly and casual atmosphere.↩ We're open every night from↩ 5:30-10:30 for dinner with↩ Brunch on Sundays and live↩ entertainment in our OASIS↩ lounge until the wee small↩ hours. Oh Yes, we specialize↩ in Italian and French↩ cuisine with lots of fresh↩ local seafood and Kauai's↩ only Fresh Fruit Daquiris.↩ Call us for reservations at 245-9181↩ and free hotel pickup↩ from most resorts.↩ I know you'll love↩ Kauai and have the↩ time of your life↩ at the Casa Blanca.↩ the↩ Bestsellers↩ Games↩ Hawaiiana↩ We have the most complete selection↩ of paperback books on the island.↩ Over 5,000 books in stock.↩ OPEN EARLY-CLOSE LATE↩ The Market Place at Coconut Plantation↩ Waipouli, Kauai↩ 822-3216↩ CLUBIETTY↩ Restaurant and Cabaret↩ Nawiliwili Bay↩ CANTONESE FOOD↩ a specialty of the house↩ COMPLETE MENU-including↩ STEAK-LOBSTER-MAHIMAHI↩ DINNER: 5:30-9:45 p.m.↩ Closed TUESDAYS↩ MUSIC to Dine & Dance by-7:30 p.m.↩ After dinner Dance Band & DISCO↩ Courtesy pick-up-Lihue area↩ 245.4970....after hours 245.3856↩ 2989 HALEKO ROAD↩ 245-9181↩ SUGAR MILL SNACKS↩ ASIAJOE↩ .MUUMUUS. SOUVENIRS↩ HANDICRAFTS IMPORTS↩ COCONUT↩ PLANTATION-↩ MARKET PLACE↩ 3↩ o Fresh Fruit↩ Drinks↩ e Cold↩ Drinks↩ e Sandwiches↩ Macadamia↩ Nut Waffle↩ Fresh Fruit↩ o Ice Cream↩ c Berry↩ VELVET PAINTINGS. T-SHIRTS↩ The Market Place At Coconut Plantation↩ 484 Kuhio Hwy. at Waipouli, Kapaa, Kauai↩ OPEN 7 AM M-S; Sun. 8 AM↩ 822-9981↩ 36↩ Latitude 20/November 1978
Answer	Mondays ✗ **Incorrect**	Tuesdays ✓ **Correct**

Table 9: Case study from InfographicsVQA. In this case, both VisRAG and TextRAG successfully retrieve the correct document; however, only VisRAG effectively leverages the layout information, enabling accurate generation. In contrast, TextRAG suffers from information loss of the layout, resulting in incorrect responses.

	TextRAG	**VisRAG**
Query	What percent of account holders in Europe are using LinkedIn for finding job?	
Retrieved Top-1 Document	 ✓ **Both Correct**	
Document Parsing Result	Social media↩ job seeking trends↩ Michael Page's annual global survey of financial services and banking↩ employees was conducted in April 2014,more than 3,300 people participated↩ Linkedln↩ Linkedin's popularity continues to grow, though many job seekers don't think of it as part of↩ their strategy.So hirers need to look to other sourcing channels too↩ What proportion of account holders↩ use Linkedin for job seeking?↩ 93↩ %↩ 30%↩ of respondents have↩ anaccount-up↩ 10% from last year↩ more women↩ than men say↩ they don't have↩ an account↩ 53%↩ In Europe↩ 49%↩ In North America↩ 40%↩ In the UK↩ Facebook↩ Despite last year's hype around Graph Search,Facebook hasn't made any progress with monetising↩ its recruitment potential -jobseekers remain very negative about Facebook playing any part↩ 13%↩ said they'd be happy↩ to see adverts↩ 92%↩ said they would not be↩ happy to be contacted by↩ a recruiter on Facebook↩ 1%↩ Don't bank on social media – Michael Page brings you a broader range of talent, and jobs↩ www.michaelpage.com.au/salarycentre↩ of respondents↩ (who are job seekers) said they↩ would use it to look for jobs↩ MichaelPage↩ Financial Services↩ Specialists in financial services recruitment↩ www.michaelpage.com.au↩	
Answer	49% ✗ **Incorrect**	53% ✓ **Correct**

G ADDITIONAL RETRIEVAL AND GENERATION RESULTS

Table 10: Additional retrieval performance in MRR@10.

Model	ArxivQA	ChartQA	DocVQA	InfoVQA	PlotQA	SlideVQA	Average
(b) Out-of-domain: *Models Fine-tuned on Synthetic Data*							
MiniCPM (OCR)	37.19	43.34	68.57	75.36	21.93	86.80	55.53
SigLIP (2023)	37.01	49.34	58.32	63.55	21.23	85.32	52.46
MiniCPM (OCR) + SigLIP (RRF)	41.87	49.88	67.19	72.13	24.38	87.79	57.21
(c) In-domain: *Models Fine-tuned on Synthetic and In-domain data*							
MiniCPM (OCR)	47.36	54.43	74.13	80.11	41.33	90.49	64.64
SigLIP (2023)	50.22	61.44	66.01	72.87	40.15	89.51	63.37
MiniCPM (OCR) + SigLIP (RRF)	54.60	61.20	73.31	78.62	42.84	91.80	67.06

Table 11: Additional generation performance in accuracy (%). Different generation models and methods utilize the same retriever, VisRAG. Performance relative to Oracle (using the ground-truth document(s) for generation) is colored in blue.

Model / Method	Input	ArxivQA	ChartQA	DocVQA	InfoVQA	PlotQA	SlideVQA	Average
(b) VisRAG-Gen: *Single-image VLM (MiniCPM-V 2.0)*								
Page Concatenation	top-6	56.20 (93.5%)	25.63 (67.9%)	11.44 (23.7%)	12.90 (45.5%)	17.10 (55.5%)	25.67 (67.3%)	24.83 (58.9%)
	top-10	55.69 (92.7%)	24.23 (64.2%)	7.82 (16.2%)	12.32 (43.4%)	20.08 (65.1%)	23.05 (60.4%)	23.87 (57.0%)
	Oracle	60.10 (100%)	37.74 (100%)	48.38 (100%)	28.35 (100%)	30.84 (100%)	38.17 (100%)	40.60 (100%)
(c) VisRAG-Gen: *Multi-image VLM*								
MiniCPM-V 2.6	top-6	66.08 (91.4%)	42.90 (65.3%)	64.34 (78.5%)	43.94 (79.1%)	34.85 (58.6%)	50.00 (88.0%)	50.35 (76.8%)
	top-10	63.84 (88.3%)	42.48 (64.6%)	48.91 (59.6%)	30.60 (55.1%)	33.28 (55.9%)	50.67 (89.2%)	44.96 (68.8%)
	Oracle	72.27 (100%)	65.74 (100%)	82.01 (100%)	55.57 (100%)	59.51 (100%)	56.83 (100%)	65.32 (100%)
Qwen2-VL	top-1	62.50 (89.1%)	44.99 (67.3%)	63.97 (76.0%)	48.92 (82.6%)	49.21 (70.7%)	51.28 (80.7%)	53.48 (77.8%)
	top-2	63.97 (91.2%)	45.13 (67.5%)	68.44 (81.3%)	48.78 (82.3%)	47.02 (67.6%)	55.85 (87.9%)	54.86 (79.7%)
	top-3	63.97 (91.2%)	45.82 (68.5%)	68.92 (81.9%)	47.12 (79.5%)	46.41 (66.7%)	56.52 (89.0%)	54.79 (79.5%)
	Oracle	70.12 (100%)	66.85 (100%)	84.14 (100%)	59.24 (100%)	69.56 (100%)	63.54 (100%)	68.91 (100%)

In this section, we present supplementary evaluation results for both retrieval and generation on our dataset.

Table 10 shows additional retrieval results obtained by applying reciprocal rank fusion (RRF) (Cormack et al., 2009) to combine the outputs of MiniCPM (OCR) and SigLIP. It is a straightforward method to integrate textual information extracted from the page with its visual clues. The results indicate that fusing text and image modalities provides a meaningful performance boost over individual modality baselines. However, this approach still falls short of the performance achieved by our VisRAG-Ret model (63.96 for out-of-domain, 70.00 for in-domain). This underscores the superior capability of VisRAG-Ret in understanding both modalities within a unified architecture.

Table 11 provides additional generation results using top-6 and top-10 retrieved documents from VisRAG-Ret. For these experiments, we evaluate the performance of MiniCPM-V 2.0 using the page concatenation method and MiniCPM-V 2.6 with direct feeding. We also report the performance of another SOTA VLM, Qwen2-VL-7B-Instruct (Wang et al., 2024). The results indicate significant performance degradation when handling a larger number of retrieved pages, for both page concatenation (MiniCPM-V 2.0) and multi-page input (MiniCPM-V 2.6). MiniCPM-V 2.6 exhibits greater robustness to increasing context compared to MiniCPM-V 2.0. Open-source VLMs still face challenges in reasoning over multiple pages and extracting relevant information from noisy retrieved data. Results for Qwen2-VL demonstrate stronger document understanding capabilities, outperforming MiniCPM-V 2.6 in these tasks.

H RETRIEVAL EFFICIENCY

In this experiment, we evaluate the retrieval efficiency of VisRAG-Ret and MiniCPM (OCR) by measuring two key components: offline document parsing and encoding latency, and online query encoding and search latency. Query and document encoding are conducted on an NVIDIA A100 40G GPU with a batch size of 1, while document parsing is performed on a single core of an Intel Xeon Platinum 8350C CPU. The reported latencies are averaged over 11,307 queries and 9,593 documents from the PlotQA dataset. The results are summarized in Table 12.

Table 12: Retrieval efficiency (ms). We report offline latencies per document, including document parsing and encoding latencies, as well as online latencies per query, including query encoding and search latencies.

	Offline Latency per Document			**Online Latency per Query**		
	Parsing	**Encoding**	**Total**	**Encoding**	**Search**	**Total**
MiniCPM (OCR)	284	28	312	28	26	54
VisRAG-Ret	–	121	121	28	26	54

As shown in the table, although VisRAG-Ret, a VLM-based model, requires more time for document encoding compared to MiniCPM (OCR), it bypasses the time-consuming parsing stage required by MiniCPM (OCR). This leads to a 58% reduction in total document processing time for VisRAG-Ret. For online query processing, the latencies of VisRAG-Ret and MiniCPM (OCR) are nearly identical, as the queries consist solely of textual inputs.

I RETRIEVAL PERFORMANCE ON TEXT RETRIEVAL BENCHMARKS

Table 13: Retrieval performance on subsets of the text retrieval benchmark BEIR (Thakur et al., 2021) in NDCG@10. VisRAG-Ret performs retrieval on rendered document screenshots.

Model	**SciFact**	**NFCorpus**	**Scidocs**
MiniCPM (OCR)	61.04	14.12	13.01
VisRAG-Ret	62.47	27.02	16.25

To evaluate how VisRAG-Ret performs in retrieval scenarios involving only textual data, we conduct an experiment using the BEIR (Thakur et al., 2021) text retrieval benchmark. To evaluate VisRAG-Ret, we convert the document texts into rendered screenshots and apply VisRAG-Ret to this modified dataset. We use the Pillow[3] library to convert text documents into screenshots, setting a width of 800px, a font size of 24px, and the DejaVuSans font. The height of each screenshot varies depending on the document length, with a margin of 20px and a line spacing of 4px. For comparison, we include MiniCPM (OCR) in the evaluation, utilizing raw textual data directly available in BEIR. Note that the term "OCR" in MiniCPM (OCR) is used solely for naming consistency.

As shown in Table 13, VisRAG-Ret, relying only on the rendered screenshots, significantly outperforms MiniCPM (OCR) which uses textual information. This result highlights that VisRAG-Ret's pooling-based representation effectively captures textual details and is well-suited for text-heavy document retrieval.

[3]https://python-pillow.org/