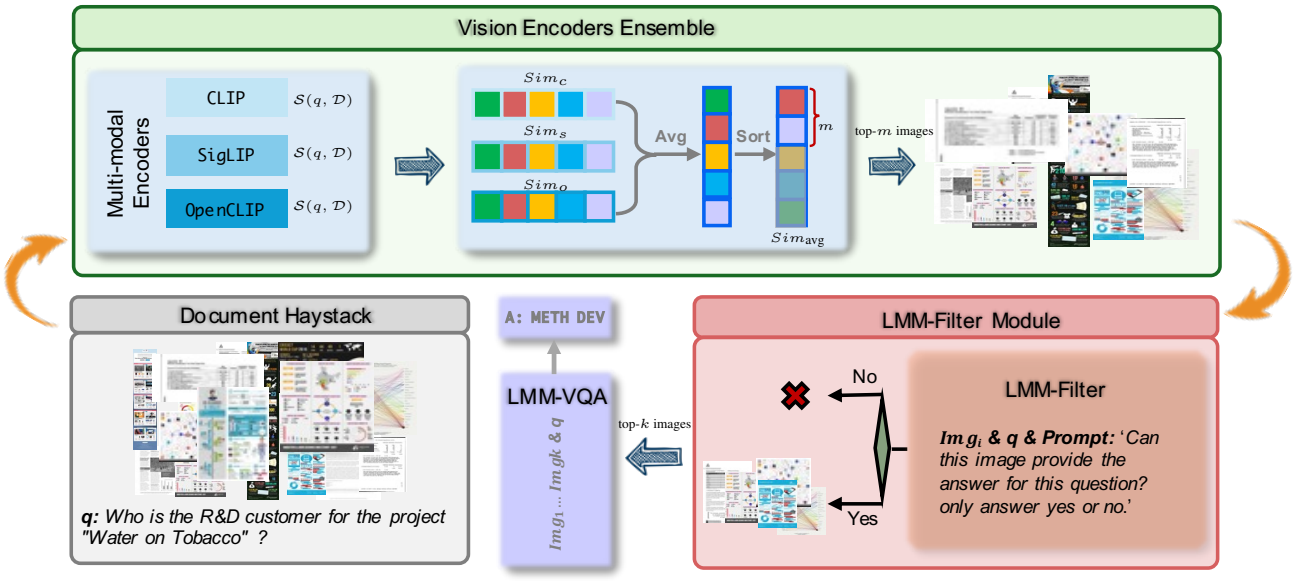

Figure 3. **The V-RAG pipeline workflow.** In the top section, a vision encoder ensemble is used, combining multiple vision models—CLIP, SigLIP, and OpenCLIP—to process a large document haystack. Each encoder computes similarity scores, which are averaged into Sim_{avg}. The top m documents, based on these scores, are selected for further analysis. In the bottom right, the LMM-Filter Module utilizes a pretrained LMM to assess whether each selected document can potentially answer the posed question. This filtering step removes documents that do not match, retaining only relevant ones. Finally, the top k most relevant images are input into the LMM along with the original question q to generate a specific answer.

2, to ensure high-quality, unique-answer questions. First, we employ a large language model (LLM) to filter out general questions that could generate multiple answers across documents. Next, a manual review step further checks the questions to ensure the data quality. Finally, a generic-knowledge filtering stage refines the dataset further, retaining only questions closely tied to specific document content.

This carefully designed pipeline, combining LLM-based filtering and human review, effectively curates questions that drive accurate, document-specific retrieval. By focusing on reducing answer ambiguity, DocHaystack and InfoHaystack enhance the precision of retrieval and reasoning in large-scale document processing tasks, providing a valuable tool for the evaluation of retrieval systems. We discuss this data curation pipeline in details as follows:

General-question LLM filtering. We begin by using the LLM, GPT-4o [30], to filter out general questions through a set of well-crafted instructions. Leveraging the LLM's strong contextual understanding, this initial filtering step allows us to efficiently process large volumes of data, identifying broad or ambiguous questions that may yield multiple answers across documents. This automated approach significantly enhances the benchmark construction's efficiency and quality.

To guide the LLM, we first define the task, providing clear distinctions between general and specific questions along with illustrative examples. With this framework, the LLM can then assess each question and determine if it is general or specific. The instructional format is as follows:LLM i

You are an evaluator tasked with identifying if a question is specific or general. A general question seeks commonly known or widely applicable information without unique identifiers, e.g., "Who is the person standing in the ground?" A specific question, however, requests unique information about a particular individual, event, or object, e.g., "What is the Social Security Number of Charles Yarbrough?" Based on these definitions, determine if the following question is general or specific: {question}.

General-question manual review. After the initial LLM filtering, we conduct a manual review of the questions that were classified as specific. This manual process involves two key steps to ensure answer uniqueness and benchmark quality.

In the first step, we examine each question to confirm it contains unique identifiers—such as names, dates, titles, or other specific attributes—suggesting a document-specific answer. This careful check helps identify questions with clear, unique markers that direct the retrieval process to a single document.

In the second step, we verify the uniqueness of each answer to eliminate any remaining ambiguity. Although specific identifiers are present, questions may still be prone

	GPT-4o	LLaVA-OneVision	Qwen2-VL
DocVQA	26.4%	4.7%	3.4%
InfographicVQA	54.9%	13.4%	11.3%

Table 1. **Percentage of questions answerable by LMMs without vision input.** We evaluate GPT-4o, LLaVA-Onevision, and Qwen2-VL on their ability to answer questions directly from our dataset without requiring vision input. The reported percentage reflects the proportion of examples that can be answered solely through language understanding.

to ambiguity, such as with common names or recurring book titles. To address this, we employ a refined verification process. First, we use Optical Character Recognition (OCR) [36] to extract all text from images in the dataset. We then search for occurrences of the unique identifiers retained from the first step across other documents. If matches are found, a manual review is conducted to ensure no alternative valid answers exist. This comprehensive approach minimizes the possibility of a single question mapping to multiple answers, enhancing the precision and reliability of our benchmarks.

Generic-knowledge filtering. In DocVQA and InfographicVQA tasks, certain questions—such as "`How many sports were in the 2008 Beijing Paralympic Games?`"—can be answered based on general knowledge accessible to a large language model, without relying on the image content. This introduces a language bias when using LMMs for visual question answering, as it shifts the focus away from image-based reasoning. To address this, we filter out these general-knowledge questions, ensuring that evaluation emphasizes vision-based understanding and that models rely primarily on visual content to generate accurate answers.

To implement this, we developed an LLM-based evaluation pipeline that detects and excludes such questions. For each question, we prompt an LLM with "`{question}, answer briefly.`". After receiving a response, we compare it to the ground-truth answer using another LLM. If the response matches the ground truth, we classify the question as general knowledge-related and remove it, thereby isolating questions that truly require visual document understanding. As shown in Table 1, GPT-4o accurately answers 26.4% of DocVQA questions and 54.9% of InfographicVQA questions directly, a rate significantly higher than that of open-source LLMs. Therefore, we select GPT-4o to filter out the questions that can be directly answered by the GPT-4o model. Overall, this process is to ensure that the evaluation reflects the necessity of vision-based comprehension.

Final dataset profile. After a rigorous three-stage data filtering process, we retained 109 questions from DocVQA and 155 questions from InfographicVQA, associated with

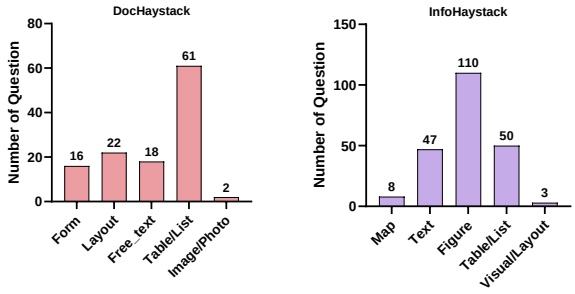

Figure 4. **Question type analysis.** We analyze the distribution of question types of DocHaystack and InfoHaystack. Each benchmark categorizes the data into 5 different types.

59 and 66 documents that provide the evidence, respectively. To assess retrieval performance at scale, we introduce two benchmarks: DocHaystack-1000 and InfoHaystack-1000, where each question requires retrieving relevant content from a set of 1,000 documents. Given the challenge this scale presents to current LMMs, particularly in terms of context length limitations, we also construct two smaller benchmarks: DocHaystack-100/200 and InfoHaystack-100/200. These benchmarks allow direct input of all associated images into the context, enabling evaluation of models' long-context comprehension ability. For training set, we also construct a dataset comprising 2,835 questions similarly, with 899 from DocVQA and 1,936 from InfographicVQA, to support robust learning and generalization for the multi-image reasoning tasks.

Question type analysis. The types of questions represent the types of the evidence required for accurate answers. In Figure 4, we illustrate the distribution of question types across our dataset to provide insights into its structure. Following the classification system used in DocVQA and InfographicVQA, we categorize questions accordingly (note that a single question may fall into multiple categories). As shown in the figure, the DocHaystack benchmark places a greater emphasis on Table/List and Layout understanding, whereas InfoHaystack primarily targets Figure, Text, and Table comprehension.

4. Method

Current large multimodal models (LMMs) face substantial challenges when reasoning across hundreds or thousands of images, due not only to context length limitations but also to the inherent complexity of the task. This issue is particularly pronounced in our benchmarks, which contain 1k document files requiring high-resolution images to capture and interpret small-font text effectively. To enable LMMs to perform reasoning over a substantial number of documents, we introduce a vision-centric retrieval-augmented generation (V-RAG) framework. V-RAG efficiently retrieves a re-

duced set of relevant documents, allowing the LMM to focus on a manageable subset for deeper understanding, as illustrated in Figure 3. In the following section, we provide a detailed description of the V-RAG pipeline.

Task definition. Given a question q and a collection of N documents $\mathcal{D} = \{D_1, \ldots, D_N\}$, the V-RAG framework aims to retrieve the top-k most relevant documents to support LMMs understanding and answering the question q. V-RAG accomplishes this through a two-step retrieval process designed to effectively identify and rank relevant documents for each question.

Vision encoder ensemble. Document files often contain a mix of text, symbols, and visual elements across various scales, requiring vision encoders to capture a comprehensive understanding of these complex structures. To efficiently handle this diversity, we represent each document as an image and utilize an ensemble of vision encoders, including CLIP [33], SigLIP [45], and OpenCLIP [16], each bringing distinct strengths to the image understanding, as depicted in Figure 3. For example, the ConvNext encoder [21] from OpenCLIP [16] is particularly effective for high-resolution image encoding. We compute the similarity score between each question q and all documents in the document set $\mathcal{D}$ according to Equation 1, with similarity scores from each encoder represented as Sim_c, Sim_o, and Sim_s respectively.

$$\mathcal{S}(q, \mathcal{D}) = cos(\phi_{\text{t}}(q), \phi_{\text{v}}(D_j)) \mid D_j \in \mathcal{D}, \tag{1}$$

where $\mathcal{S}$ denotes the computing the similarity between the query q and a collection of documents $\mathcal{D}$. cos denotes the cosine similarity. ϕ_{t} denotes the text encoder, and ϕ_{v} denotes the vision encoder.

To derive a final relevance score, we calculate the average similarity Sim_{avg} for each question-image pair by combining Sim_c, Sim_o, and Sim_s. We then rank the images based on Sim_{avg} in descending order, selecting the top-m most relevant images according to their similarity scores.

LMM-filter module. To refine the selection of top-m relevant images further, we introduce a LMM-based question-image relevance assessment module. This module evaluates the relevance between each question and the top-m images identified in the first filtering step. Specifically, we pair each image with the question text and input them into an open-source vision-language model, prompting, "Can this image provide answers to this question? Respond only with yes or no". We only retain the question-image pairs that are identified as "yes" from LMM, and remove other irrelevant images.

LMM-VQA module. Achieving high top-1 ranking accuracy in image retrieval is challenging, so we retain the top-k images from the LMM-filtered ranking list and present them to the LMM-VQA to improve the likelihood of including relevant images. We input these top-k images alongside the question into the LMM-VQA (see Figure 3), which then generates the answer directly. To enhance robustness against visual distractors, the LMM-VQA can be further optimized, as analyzed in the experiment section.

5. Experiments

In the experiments section, we will primarily describe our training setup, covering evaluation metrics, baseline models, and the fine-tuning procedure for the LMM-VQA model. We also present the main experimental results along with an ablation study to provide further insights.

5.1. Training setup

Metric. In our evaluation of the DocHaystack and InfoHaystack benchmarks, we employ a model-based assessment by leveraging GPT-4o-mini [30] to accurately determine whether the model predictions match target answers. This method uses a carefully structured prompt to facilitate GPT-4o-mini's evaluation of answer correctness. We empirically found that the model-based evaluation achieves higher consistency and alignment with human judgment. Additional details on the prompt design are provided in the Appendix.

For the document retrieval evaluation, we report the baseline results using Recall@1, Recall@3, and Recall@5 metrics. These metrics enable a thorough assessment of retrieval accuracy across varying levels of precision.

Baselines. In our experiment, we have evaluated several open and closed-sourced vision-language models on the retrieval and VQA performance. For the large multimodal model, we used the *gpt-4o-2024-08-06* version of GPT-4o [30], the *LLaVA-OneVision-Qwen2-7b-OV-HF* version of LLaVA-OneVision [20], and the *Qwen2-VL-7B-Instruct* version of Qwen2-VL [3]. For computing the text-to-image similarities, we employed the *Jina-CLIP-v1* [18] variant, *Nomic-Embed-Vision-v1.5* [29] variant, CLIP [33] ViT-L/14@336 variant, for SigLIP [45], the ViT-SO400M/14@384 variant, and for OpenCLIP [16], the ConvNeXt-XXL@1024 variant as well as text-based method, BM25.

In our V-RAG setting, we apply *LLaVA-OneVision-Qwen2-7b-OV-HF* for the LMM-filter module and *Qwen2-VL-7B-Instruct* for the LMM-VQA module. We select m as 60 and k as 5 in our experiment.

Optimizing the LMM-VQA module. To improve the robustness of the LMM-VQA model in handling visual question answering with multiple distractor images, we further fine-tune the model using our curated training data.

During this fine-tuning process, we introduce 1–10 randomly sampled distractor images for each question, creating a challenging setting that encourages the model to focus on relevant content amid a mix of positive and negative images. The fine-tuning is conducted with a batch size of 32 and a