جامعة حلوان
Helwan University

كلية الحاسبات والذكاء الإصطناعي
Faculty of Computers & Artificial Intelligence

Multi-Modal Document VQA with RAG

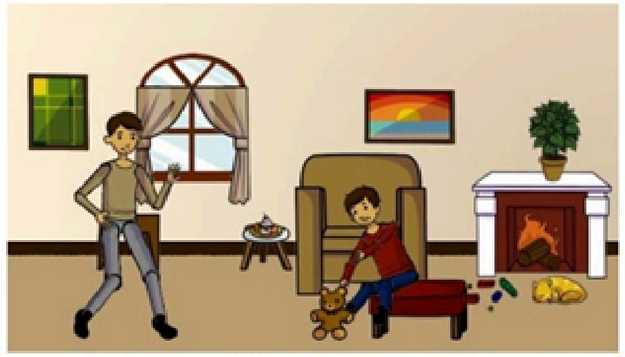
FCAI Helwan 2025

Supervisor:
DR. Amr S. Ghoneim

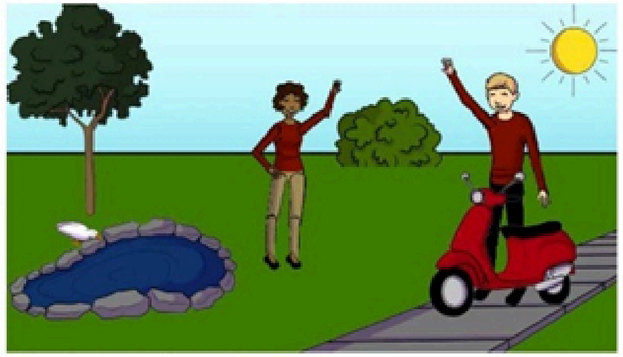
Outline

INTRODUCTION

Visual Question Answering (VQA)



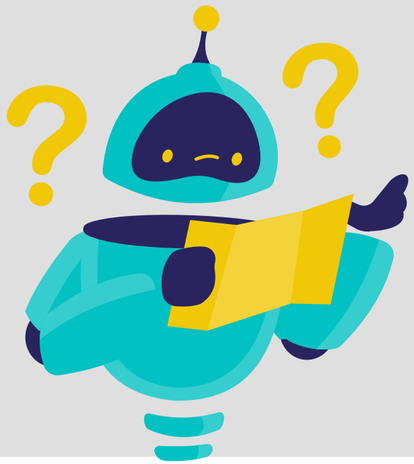
(a) **Q**: What is the boy playing with?
A: teddy bear



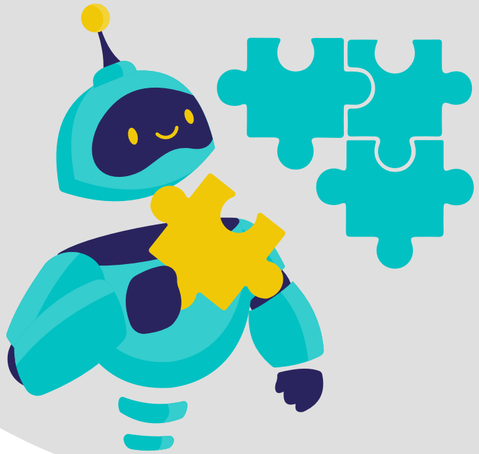
(b) **Q**: Are there any animals swimming in the pond? **A**: No

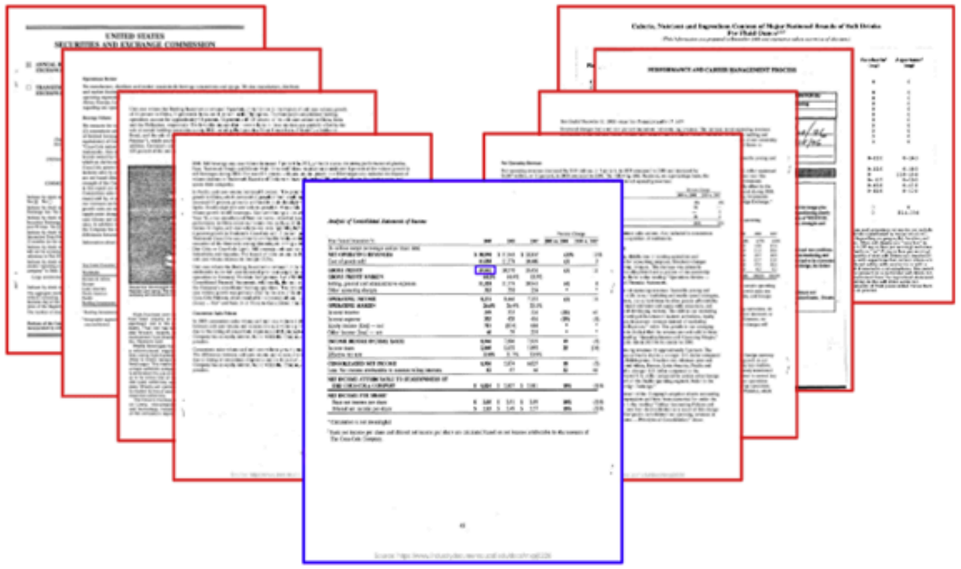
(c) **Q**: How many trees? **A**: 1

What is Document Visual Question Answering?



Document Visual Question Answering (DocVQA)



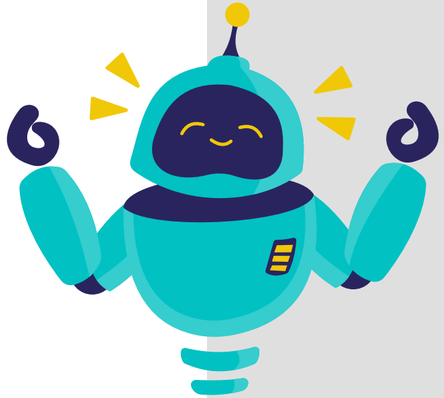


Q: What was the gross profit in the year 2009?
A: $19,902

VQA vs Document VQA

- Traditional Visual Question Answering (VQA) performs well on natural images
- Focuses on answering questions about single images using visual and textual cues, often lacking context from multi-page or structured documents.

- Document VQA (DocVQA) is more complex due to dense content, structured layouts, and text-visual interaction
- Targets multi-page documents, integrating visual (e.g., tables, charts) and textual information across pages to provide context-aware answers.



Problem Statement

Two Key Challenges in Document VQA

Context Scale

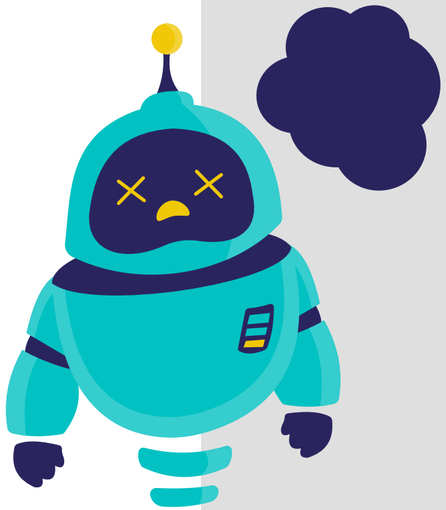
- Real-world documents are long and multi-page

- Even advanced LLMs face context limitations, leading to hallucinations and failure to synthesize cross-page information

Visual Complexity

- OCR + LLMs pipelines ignore visual structure (tables, charts, layout)

- OCR removes formatting, making the model blind to layout cues



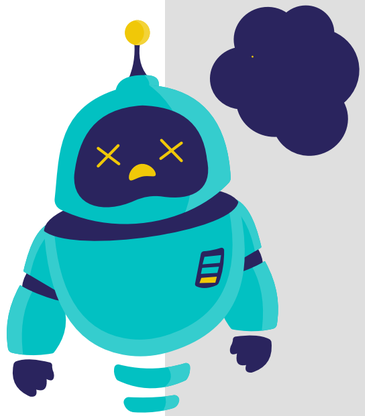
DocVQA Challenges

📌 LLMs in DocVQA

- Excellent at textual understanding & generation
- Blind to layout & visuals
- Cannot see spatial text layout (headers, tables)

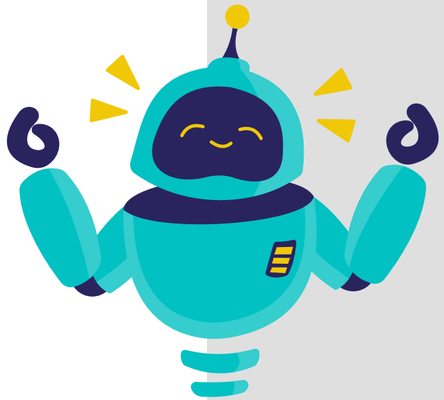
📌 VLMs in DocVQA

- Perform layout-aware reasoning
- Process images & text together
- No cross-page reasoning
- No dynamic retrieval



DocVQA Challenges

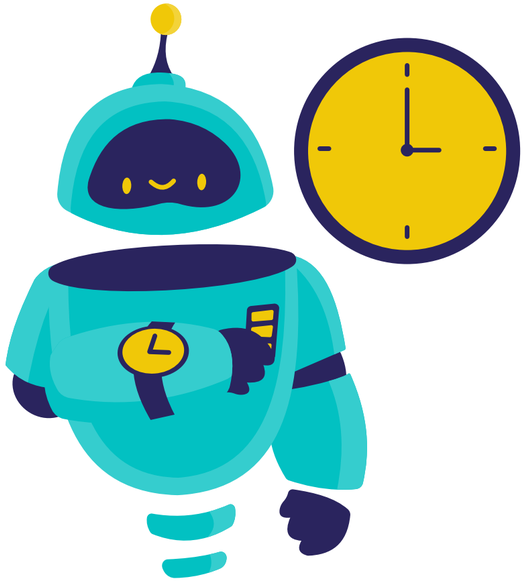
- LLMs scale well with text and work with retrieval, but are visually blind.
- VLMs see the layout and page design, but choke on scale and lack dynamic memory



Fine-Tuning of VLMs

Directly fine-tune a pre-trained VLM (e.g., LLaVA, Qwen-VL) on DocVQA data

Idea: Adapt model to document formats & question styles



fine-tuning has major drawbacks

💸 **High cost: Needs huge compute & memory**

🧠 **Catastrophic forgetting: Loses general knowledge**

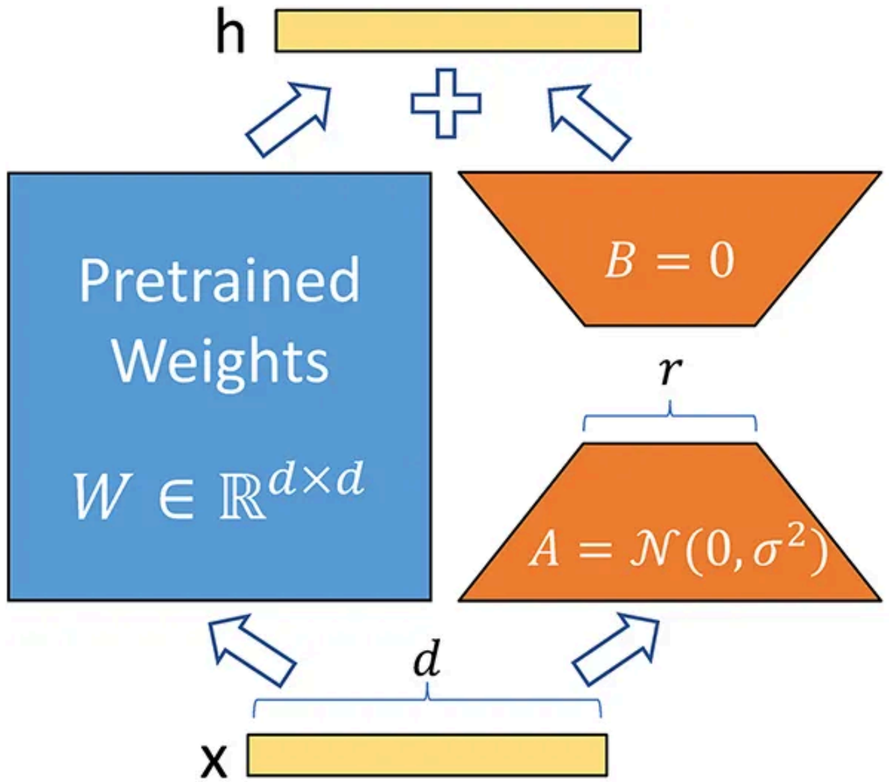
📄 **Static knowledge: Can’t handle new docs without retraining**

❓ **Low explainability: Answers lack clear evidence**

📌 LoRA: Parameter-Efficient Fine-Tuning

📌**Pros:**

- **Lower compute & memory needs**
- **Keeps general pretraining knowledge**



❌ **Cons remain:**

- **Still static — needs retraining for new docs**
- **Hallucinations persist**
- **Answers remain hard to trace back to source**

Limitations of Traditional RAG in DocVQA

📌 Strengths of Traditional RAG

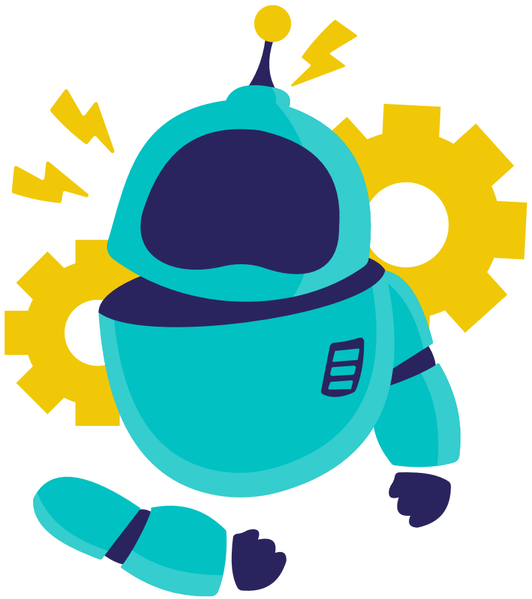
- Adds external knowledge to LLMs → improves factual accuracy
- Scalable: retrieves relevant info from large corpora
- Interpretable: answers trace back to retrieved contexts

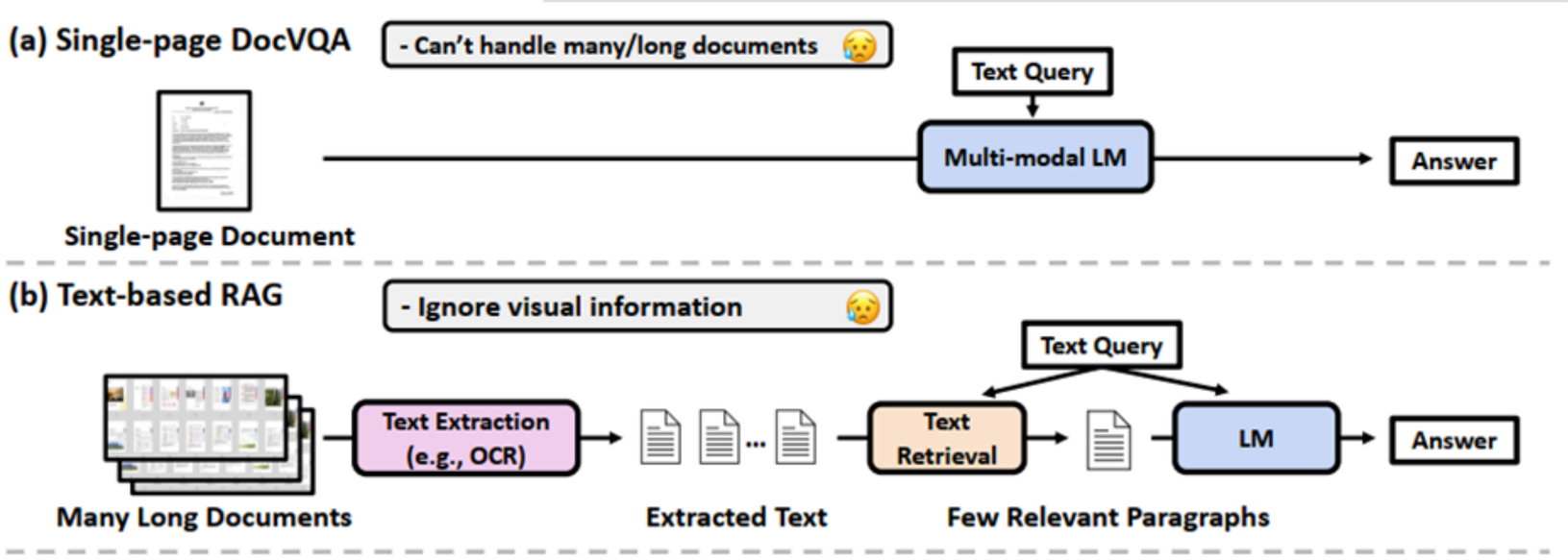
📌But: RAG Is Text-Only

- Retriever and generator work on textual data only
- Effective for plain-text QA (Wikipedia, papers)
- Fails for visually complex docs:
  - Can’t see images, tables, or layout
  - OCR flattens structure → loses cell alignment, captions, hierarchy

Existing Solutions

- **Vision-Language Models (VLMs): Visually aware, but inefficient on long documents**
- **RAG + OCR: Good at handling long context, but visually unaware**





Limitations of Traditional RAG in DocVQA

Scale vs. Visual Layout

Existing solutions force a trade-off

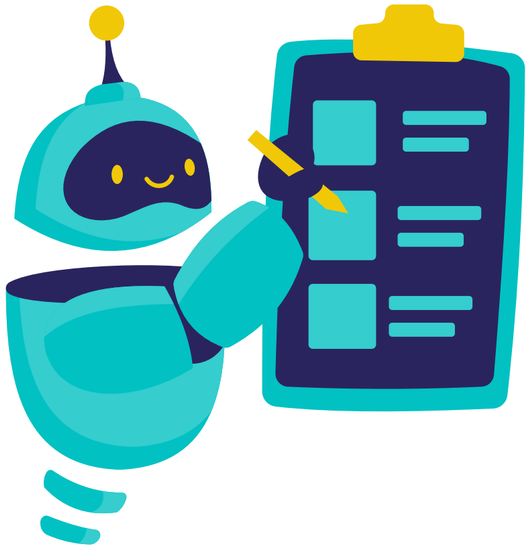
Dynamic retrieval vs. Layout understanding

The big question

How can we combine scalable retrieval with vision-language reasoning for robust, layoutaware DocVQA?

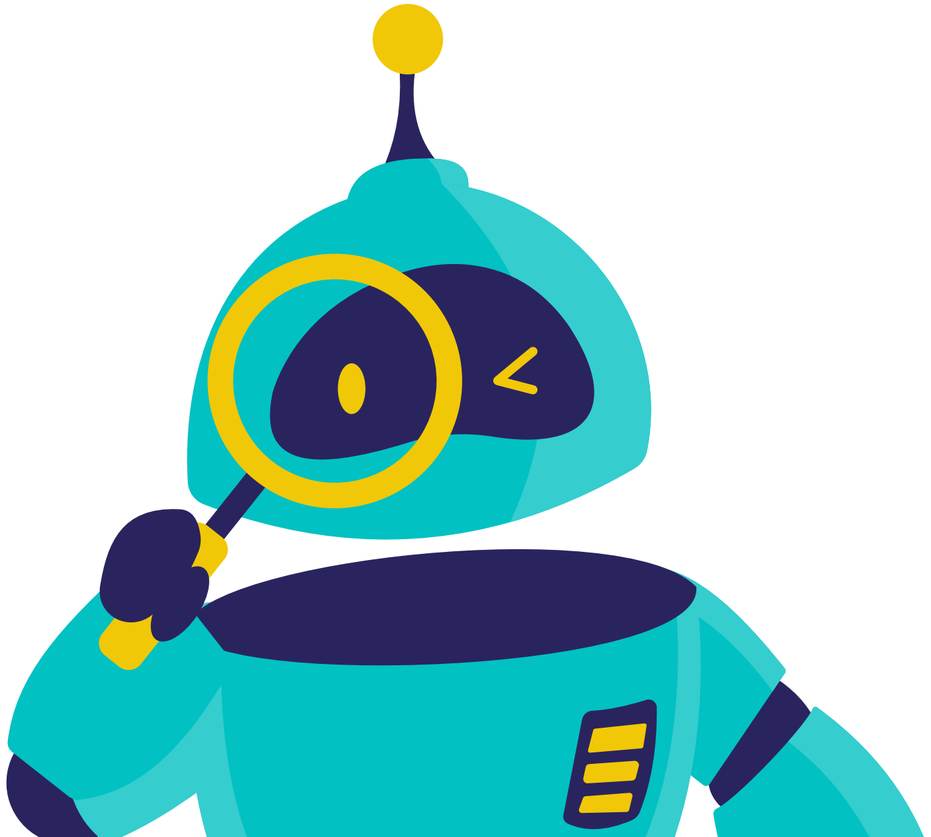
Research Goal 🧠

- **Address the trade-off between scale and visual understanding**
- **Step towards building intelligent document-understanding AI assistants**

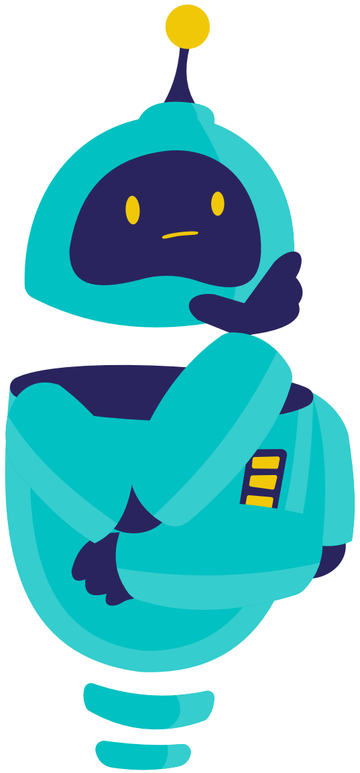


- **Develop a new framework that enables VLMs to efficiently handle long, visually complex documents**

Literature Review



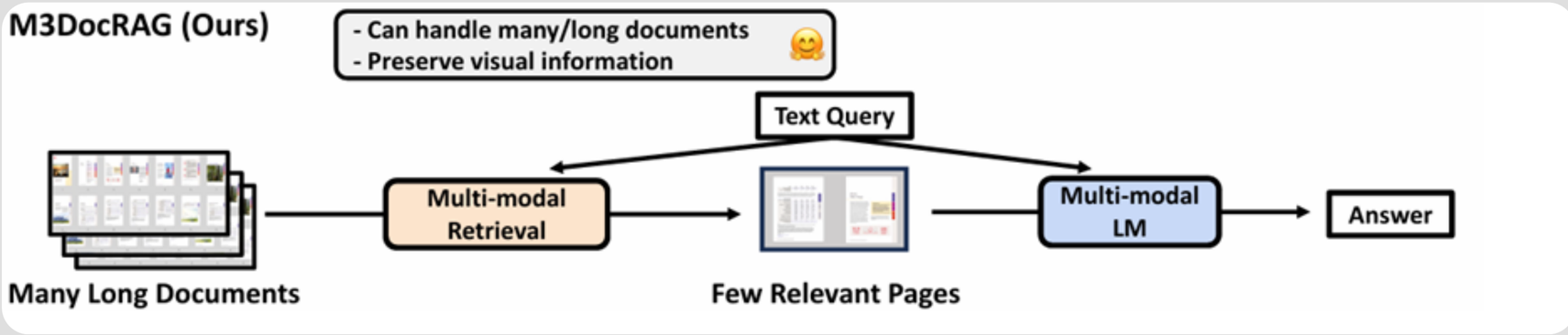
Paper	*Methodology*	*Input Modality*	*Key Contributions*	*Limitations*
Lewis et al. (2020) – RAG	Classical RAG	Text only	Introduced RAG architecture combining retrieval and generation in open-domain QA	Not applicable to multimodal inputs
Muludi et al. (2024)	Classical RAG + OCR	OCR + text	Applied GPT-3.5 and FAISS retrieval to DocVQA using textual chunks	OCR-dependent; lacks visual awareness
Adjali et al. (2024)	Multi-level RAG	Text (structured entities)	Combines entity-level and passage-level retrieval in a joint retriever-generator setup	No support for visual layout or images
VisRAG (2023)	Patch-based RAG	Image patches + OCR	Uses VLT5 and patches to retrieve relevant document regions	Fine-grained but lacks global context
M3DocRAG (2024)	Multimodal RAG	Document image + text	Introduced ColPaLi for multimodal embedding and MaxSim for page retrieval	Retrieval is static and not question-aware.
VisDoM (2024)	Visual document retriever	Document image	Performs document-wide image embedding with joint retrieval	Resource intensive; less scalable
MagicLens (2024)	Instruction-tuned retriever	Image + text + instruction	Introduces self-supervised visual retriever guided by open-ended instructions	Dependent on instruction formulation quality
PDF-WuKong (2024)	Sparse multimodal retriever	Long PDF pages	Improves document-level retrieval via sparse sampling of visual features	Requires complex training and annotation
Long et al. (2025) – ReAuSE	Unified retriever + generator	Image + text	Integrates retrieval and generation using a multimodal autoregressive model	Still limited in handling long documents



Proposed Solution

M3DOCRAG Framework

Solution Methodology Overview



01

Designed a modular, scalable system to overcome DocVQA limitations

02

Combines Retrieval-Augmented Generation (RAG) with Vision-Language Models (VLMs)

03

Focuses on long-context understanding and layout awareness

Our solution has 3 main stages

1) Dataset Preparation

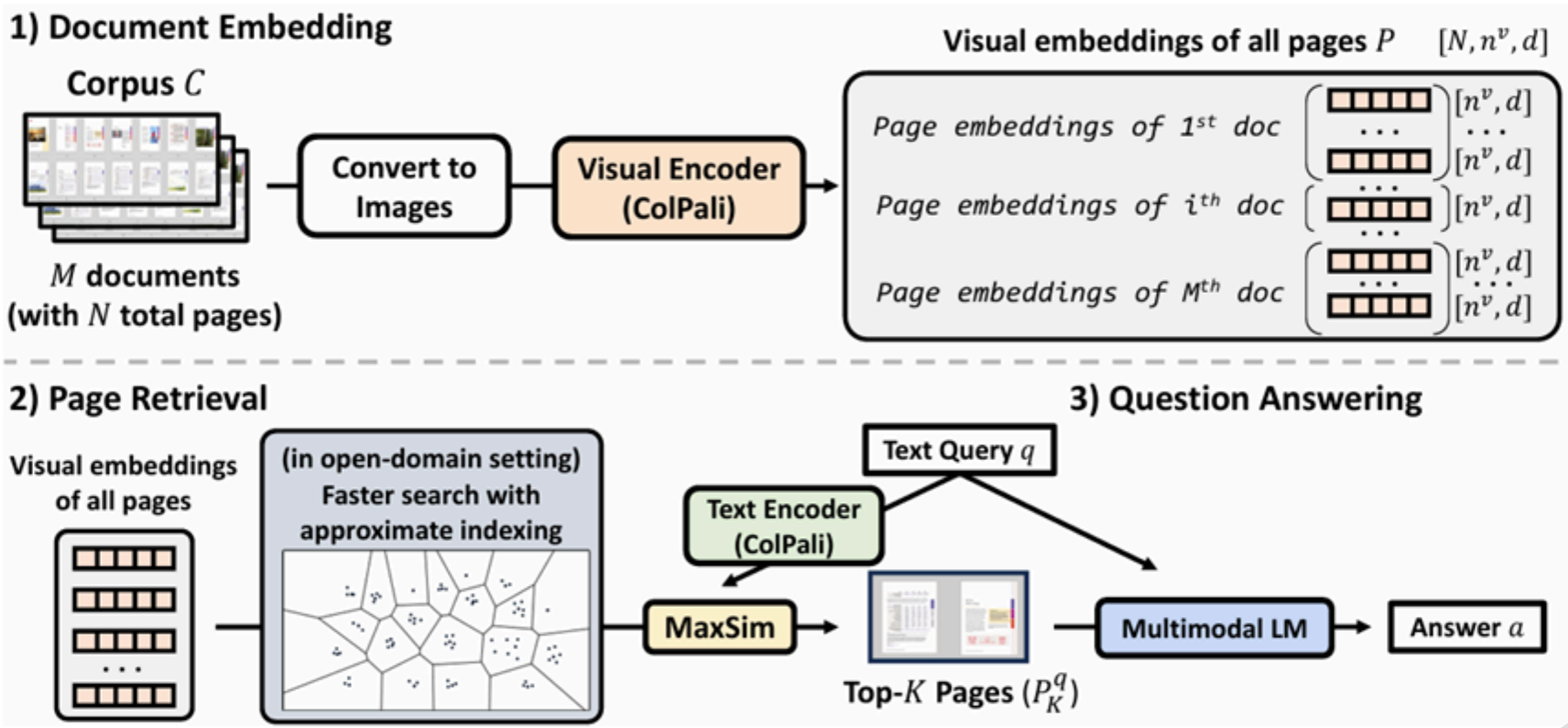
- Standardized datasets into a consistent format to simplify and streamline fine-tuning across benchmarks.

2) Page Retrieval

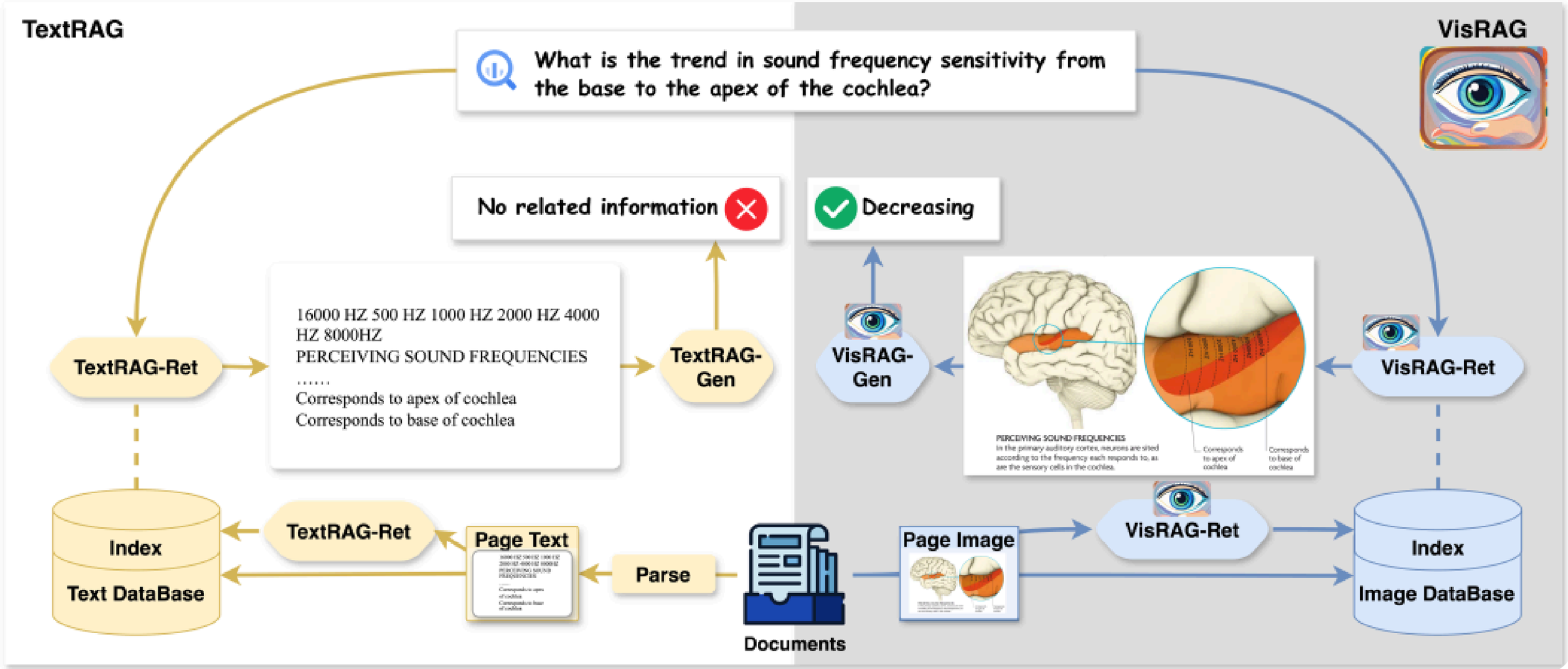
- Apply layout-sensitive retrieval to select relevant pages.retrieve top-K relevant pages based on text queries

3) Question Answering

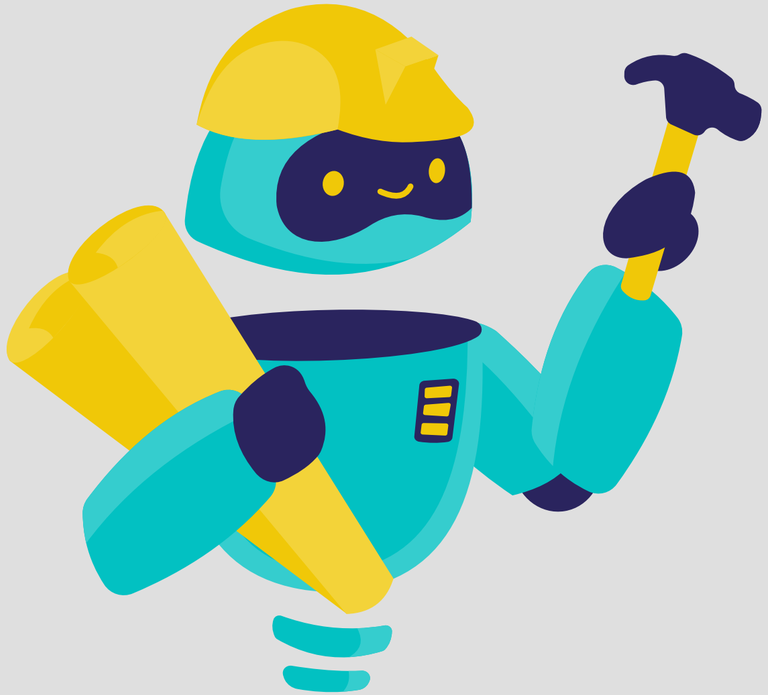
- Perform DocVQA using a Multimodal Language Model (MLM) on the selected context



TextRAG vs Multi-modal RAG



From Retrieval to Generation: How Our System Works



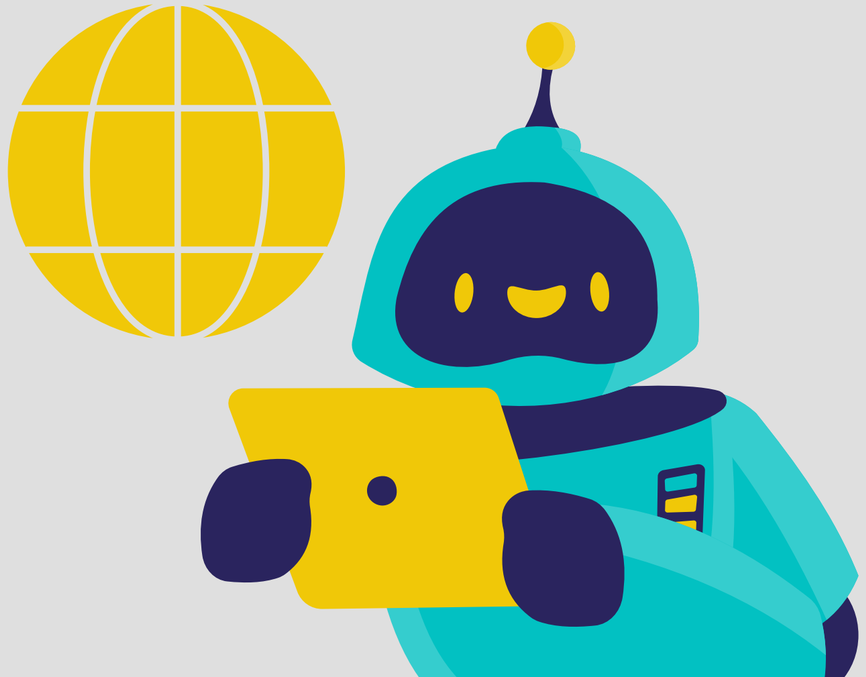
Model & Tool Selection

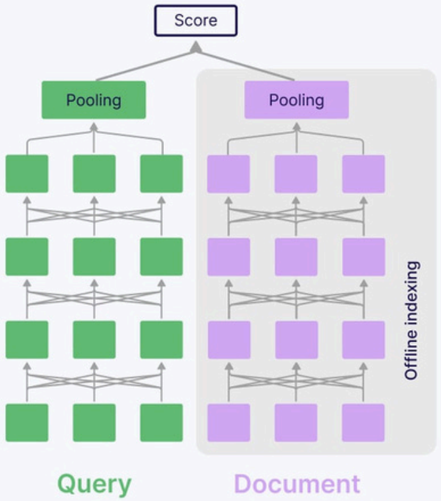
Carefully selected models/tools to ensure

High performance inmultimodal DocVQA

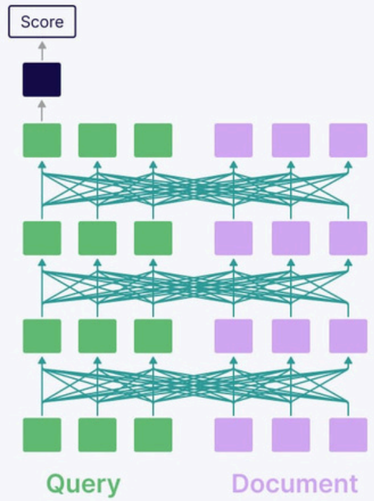
Fast processing and scalability

Flexibility across open/closed-domain settings

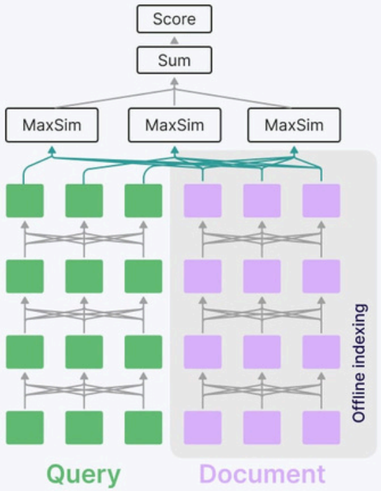




No interaction



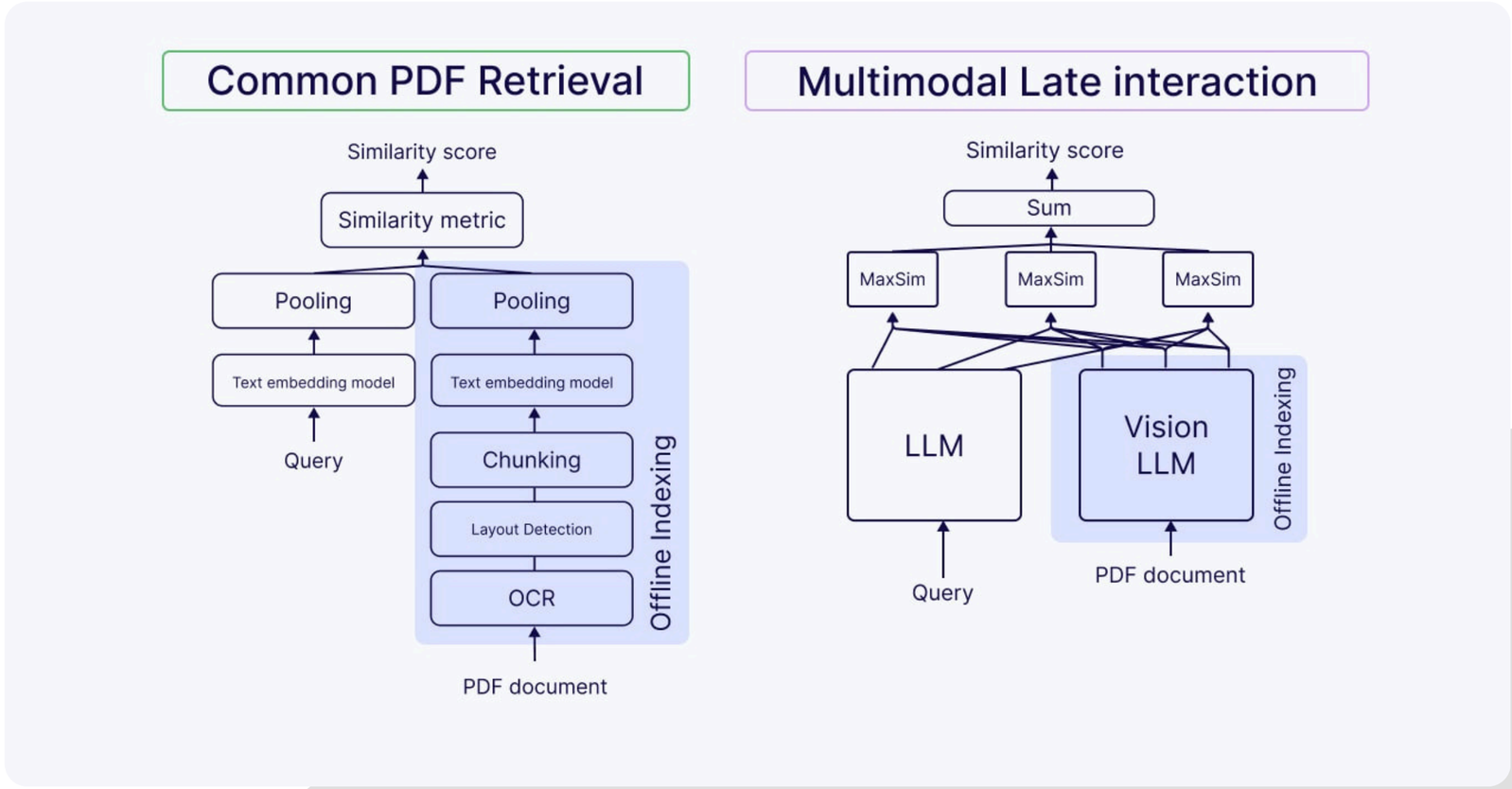
Full interaction



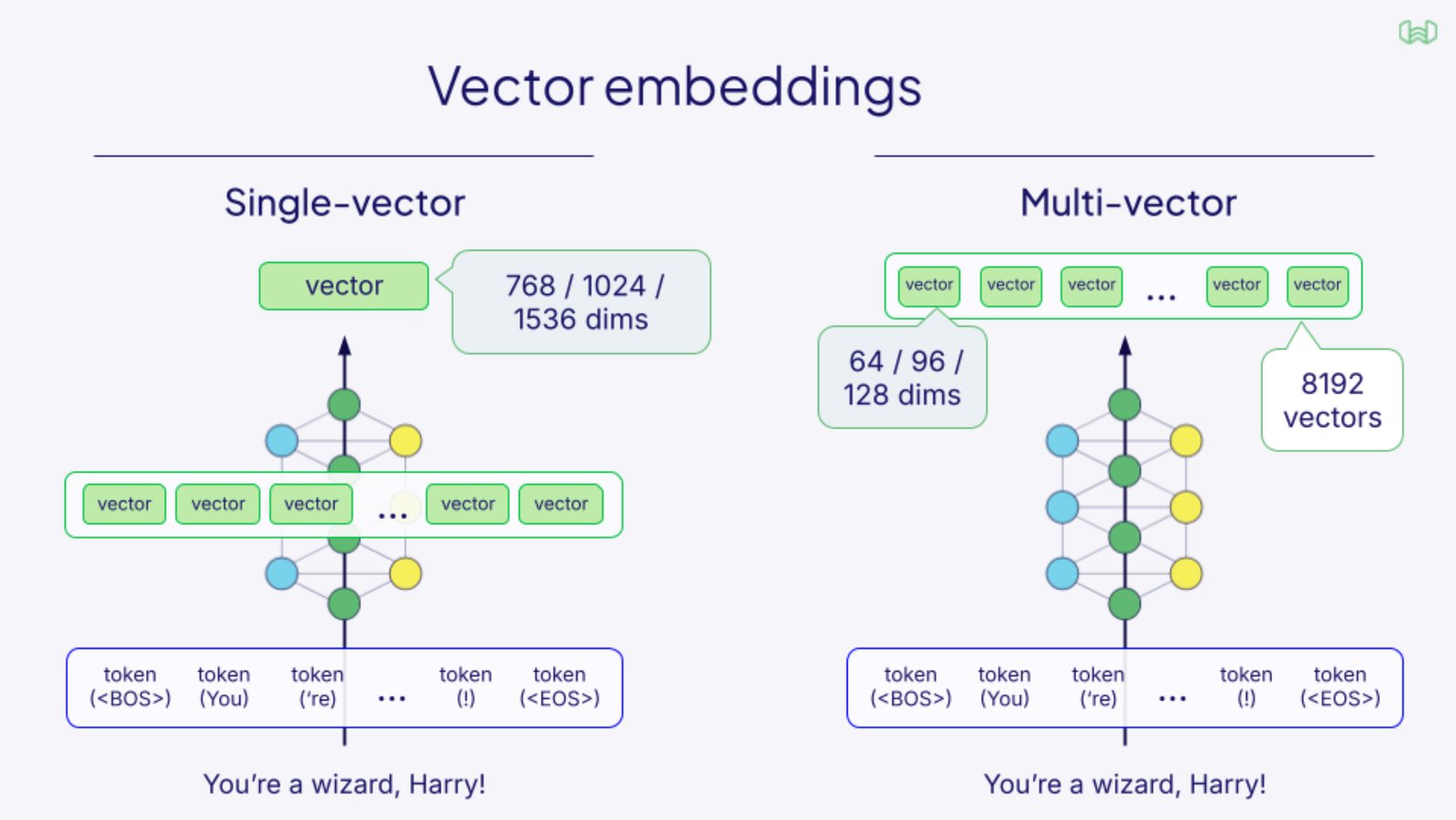
Late interaction

Retrieval & Interaction in Multimodal Document QA

How We Retrieve: From Text Chunks to Cross-Modal Signals



Embedding Documents into Vector Space:



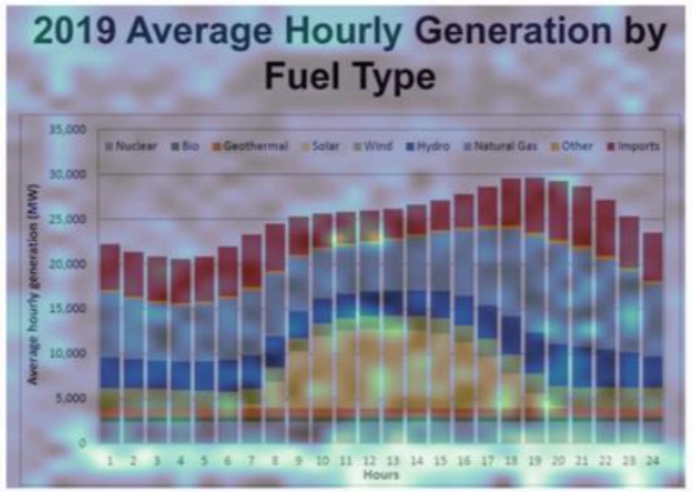
COLQWEN

Architecture

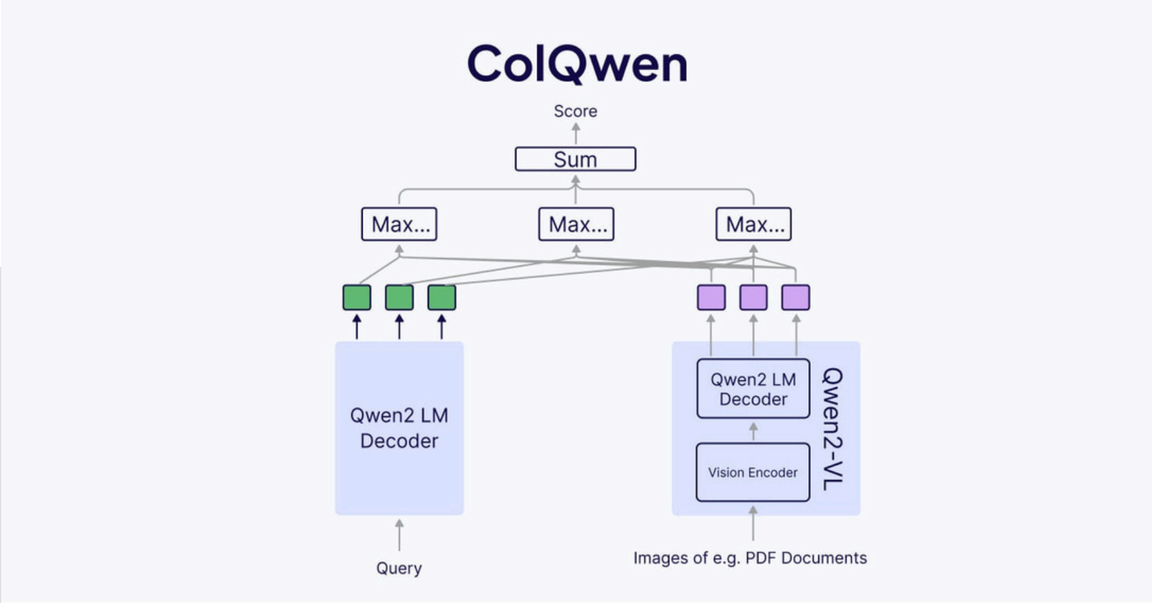
- Based on Qwen2-VL-2B, optimized for document-specific tasks.
- Produces fewer multi-vector embeddings (~700-768 per page) for efficiency.
- Employs a late-interaction mechanism similar to ColPali for precise retrieval.

Implementation

- Processes page images at native resolutions for efficient storage management.
- Leverages Vision-Language Model (VLM) with optimized vector search (e.g., FAISS).



Query: "Which hour of the day had the highest overall eletricity generation in 2019?"



Why ColQWEN?

- **Late interaction ➡ preserves layout & fine-grained features**
- **Efficient ➡ sub-2s retrieval across 40K+ pages**
- **Multimodal ➡ fuses text + visual info**
- **Empirical proof: Top performance on ViDoRe benchmark**

ViDoRe Benchmark Results (nDCG@5)

	ArxivQ	DocQ	InfoQ	TabF	TATQ	Shift	AI	Energy	Gov.	Health.	Avg.
Unstructured text-only											
- BM25	-	34.1	-	-	44.0	59.6	90.4	78.3	78.8	82.6	-
- BGE-M3	-	28.4↓5.7	-	-	36.1↓7.9	68.5↑8.9	88.4↓2.0	76.8↓1.5	77.7↓1.1	84.6↑2.0	-
Unstructured + OCR											
- BM25	31.6	36.8	62.9	46.5	62.7	64.3	92.8	85.9	83.9	87.2	65.5
- BGE-M3	31.4↓0.2	25.7↓11.1	60.1↓2.8	70.8↑24.3	50.5↓12.2	**73.2**↑8.9	90.2↓2.6	83.6↓2.3	84.9↑1.0	91.1↑3.9	66.1↑0.6
Unstructured + Captioning											
- BM25	40.1	38.4	70.0	35.4	61.5	60.9	88.0	84.7	82.7	89.2	65.1
- BGE-M3	35.7↓4.4	32.9↓5.4	71.9↑1.9	69.1↑33.7	43.8↓17.7	73.1↑12.2	88.8↑0.8	83.3↓1.4	80.4↓2.3	91.3↑2.1	67.0↑1.9
Contrastive VLMs											
Jina-CLIP	25.4	11.9	35.5	20.2	3.3	3.8	15.2	19.7	21.4	20.8	17.7
Nomic-vision	17.1	10.7	30.1	16.3	2.7	1.1	12.9	10.9	11.4	15.7	12.9
SigLIP (Vanilla)	43.2	30.3	64.1	58.1	26.2	18.7	62.5	65.7	66.1	79.1	51.4
Ours											
SigLIP (Vanilla)	43.2	30.3	64.1	58.1	26.2	18.7	62.5	65.7	66.1	79.1	51.4
BiSigLIP (+fine-tuning)	58.5↑15.3	32.9↑2.6	70.5↑6.4	62.7↑4.6	30.5↑4.3	26.5↑7.8	74.3↑11.8	73.7↑8.0	74.2↑8.1	82.3↑3.2	58.6↑7.2
BiPali (+LLM)	56.5↓-2.0	30.0↓-2.9	67.4↓-3.1	76.9↑14.2	33.4↑2.9	43.7↑17.2	71.2↓-3.1	61.9↓-11.7	73.8↓-0.4	73.6↓-8.8	58.8↑0.2
ColPali (+Late Inter.)	**79.1**↑22.6	**54.4**↑24.5	**81.8**↑14.4	**83.9**↑7.0	**65.8**↑32.4	**73.2**↑29.5	**96.2**↑25.0	**91.0**↑29.1	**92.7**↑18.9	**94.4**↑20.8	**81.3**↑22.5

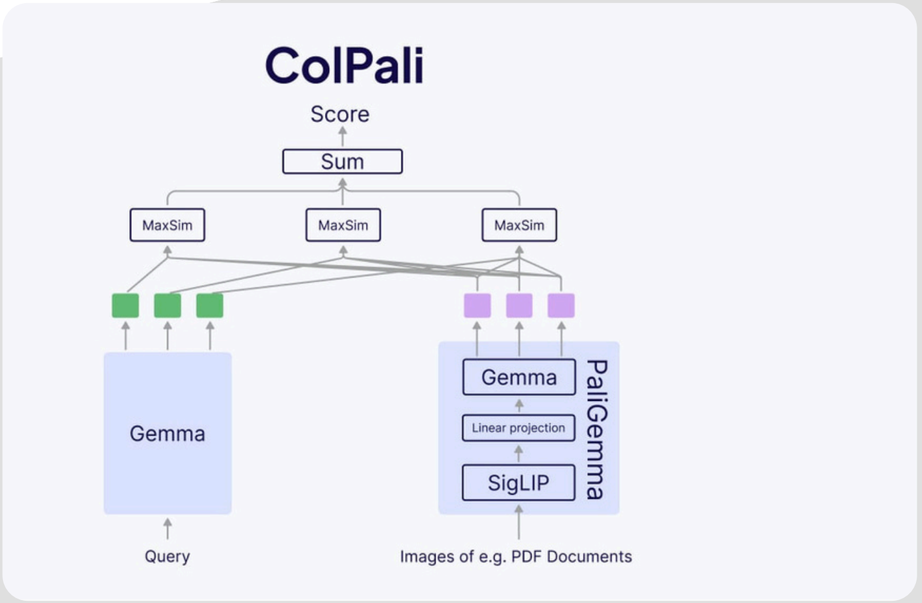
COLPALI

ColPali architecture:

- **built on PaliGemma3B a VLM __combining SigLIP and gemma-2B**
- **generates multi-vector embeddings for page images, capturing text and layout**
- **uses a late-interaction mechanism to amtch query embeddings to document embeddings.**

Model implementation:

- **Indexes pages at ~0.39s/page and queries in ~30ms**
- **Directly processes page images with each patch embedded via the VLM.**



Why ColPaLi?

- **Late interaction ➡ preserves layout & fine-grained features**
- **Efficient ➡ sub-2s retrieval across 40K+ pages**
- **Multimodal ➡ fuses text + visual info**
- **Empirical proof: Top performance on ViDoRe benchmark**

ViDoRe Benchmark Results (nDCG@5)

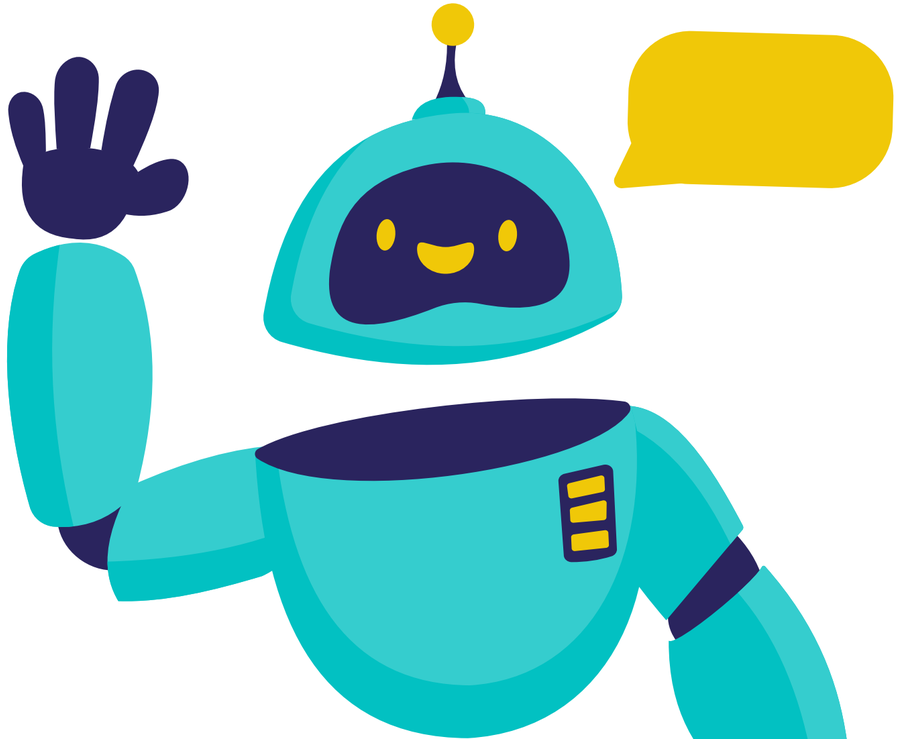
	ArxivQ	DocQ	InfoQ	TabF	TATQ	Shift	AI	Energy	Gov.	Health.	Avg.
Unstructured text-only											
- BM25	-	34.1	-	-	44.0	59.6	90.4	78.3	78.8	82.6	-
- BGE-M3	-	28.4↓5.7	-	-	36.1↓7.9	68.5↑8.9	88.4↓2.0	76.8↓1.5	77.7↓1.1	84.6↑2.0	-
Unstructured + OCR											
- BM25	31.6	36.8	62.9	46.5	62.7	64.3	92.8	85.9	83.9	87.2	65.5
- BGE-M3	31.4↓0.2	25.7↓11.1	60.1↓2.8	70.8↑24.3	50.5↓12.2	**73.2**↑8.9	90.2↓2.6	83.6↓2.3	84.9↑1.0	91.1↑3.9	66.1↑0.6
Unstructured + Captioning											
- BM25	40.1	38.4	70.0	35.4	61.5	60.9	88.0	84.7	82.7	89.2	65.1
- BGE-M3	35.7↓4.4	32.9↓5.4	71.9↑1.9	69.1↑33.7	43.8↓17.7	73.1↑12.2	88.8↑0.8	83.3↓1.4	80.4↓2.3	91.3↑2.1	67.0↑1.9
Contrastive VLMs											
Jina-CLIP	25.4	11.9	35.5	20.2	3.3	3.8	15.2	19.7	21.4	20.8	17.7
Nomic-vision	17.1	10.7	30.1	16.3	2.7	1.1	12.9	10.9	11.4	15.7	12.9
SigLIP (Vanilla)	43.2	30.3	64.1	58.1	26.2	18.7	62.5	65.7	66.1	79.1	51.4
Ours											
SigLIP (Vanilla)	43.2	30.3	64.1	58.1	26.2	18.7	62.5	65.7	66.1	79.1	51.4
BiSigLIP (+fine-tuning)	58.5↑15.3	32.9↑2.6	70.5↑6.4	62.7↑4.6	30.5↑4.3	26.5↑7.8	74.3↑11.8	73.7↑8.0	74.2↑8.1	82.3↑3.2	58.6↑7.2
BiPali (+LLM)	56.5↓-2.0	30.0↓-2.9	67.4↓-3.1	76.9↑14.2	33.4↑2.9	43.7↑17.2	71.2↓-3.1	61.9↓-11.7	73.8↓-0.4	73.6↓-8.8	58.8↑0.2
ColPali (+Late Inter.)	**79.1**↑22.6	**54.4**↑24.5	**81.8**↑14.4	**83.9**↑7.0	**65.8**↑32.4	**73.2**↑29.5	**96.2**↑25.0	**91.0**↑29.1	**92.7**↑18.9	**94.4**↑20.8	**81.3**↑22.5

ColPaLi vs ColQwen

Criteria	ColPali	ColQwen
Accuracy	Very high precision with 1,030 vectors/page.	Good accuracy with 700–768 vectors/page.
Efficiency	Slower and heavier due to more vectors.	Faster and lighter—better for large-scale use.
Licensing & Deployment	Restrictive license; limited production use.	easy to deploy and integrate.

“”””””we can make more designed criteria. “ “”””””

GENERATOR MODEL SELECTION



QWEN2.5-VL

QWEN2.5-VL

Closed-domain DocVQA evaluation results on MMLongBench-Doc

Method	# Pages	Evidence Modalities			Question Hops		Overall	
		Image	Table	Text	Single-hop	Multi-hop	EM	F1
Text RAG (w/ ColBERT v2)								
Llama 3.1 8B	1	8.3	15.7	29.6	25.3	12.3	15.4	20.0
Llama 3.1 8B	2	7.7	16.8	31.7	27.4	12.1	15.8	21.2
Llama 3.1 8B	4	7.8	21.0	34.1	29.4	15.2	17.8	23.7
M3DocRAG *(w/ ColPali)*								
Qwen2-VL 7B (Ours)	1	25.1	27.8	39.6	37.2	25.0	27.9	32.3
Qwen2-VL 7B (Ours)	2	**26.8**	**30.4**	**42.1**	41.0	25.2	29.9	34.6
Qwen2-VL 7B (Ours)	4	24.7	**30.4**	41.2	**43.2**	**26.6**	**31.4**	**36.5**

QWEN2.5-VL

Closed-domain DocVQA evaluation results on MPDocVQA

Method	# Pages	Evidence Modalities					Evidence Locations			Overall	
		TXT	LAY	CHA	TAB	IMG	SIN	MUL	UNA	ACC	F1
Text Pipeline											
LMs											
ChatGLM-128k [5]	up to 120	23.4	12.7	9.7	10.2	12.2	18.8	11.5	18.1	16.3	14.9
Mistral-Instruct-v0.2 [25]	up to 120	19.9	13.4	10.2	10.1	11.0	16.9	11.3	24.1	16.4	13.8
Text RAG											
ColBERT v2 + Llama 3.1	1	20.1	14.8	12.7	17.4	7.4	21.8	7.8	**41.3**	21.0	16.1
ColBERT v2 + Llama 3.1	4	23.7	17.7	14.9	**24.0**	11.9	25.7	12.2	38.1	**23.5**	19.7
Multi-modal Pipeline											
Multi-modal LMs											
DeepSeek-VL-Chat [38]	up to 120	7.2	6.5	1.6	5.2	7.6	5.2	7.0	**12.8**	7.4	5.4
Idefics2 [33]	up to 120	9.0	10.6	4.8	4.1	8.7	7.7	7.2	5.0	7.0	6.8
MiniCPM-Llama3-V2.5 [61, 64]	up to 120	11.9	10.8	5.1	5.9	12.2	9.5	9.5	4.5	8.5	8.6
InternLM-XC2-4KHD [15]	up to 120	9.9	14.3	7.7	6.3	13.0	12.6	7.6	9.6	10.3	9.8
mPLUG-DocOwl 1.5 [22]	up to 120	8.2	8.4	2.0	3.4	9.9	7.4	6.4	6.2	6.9	6.3
Qwen-VL-Chat [4]	up to 120	5.5	9.0	5.4	2.2	6.9	5.2	7.1	6.2	6.1	5.4
Monkey-Chat [36]	up to 120	6.8	7.2	3.6	6.7	9.4	6.6	6.2	6.2	6.2	5.6
M3DocRAG											
ColPali + Idefics2 (Ours)	1	10.9	11.1	6.0	7.7	15.7	15.4	7.2	8.1	11.2	11.0
ColPali + Qwen2-VL 7B (Ours)	1	25.7	21.0	18.5	16.4	19.7	30.4	10.6	5.8	18.8	20.1
ColPali + Qwen2-VL 7B (Ours)	4	**30.0**	**23.5**	**18.9**	20.1	**20.8**	**32.4**	**14.8**	5.8	21.0	**22.6**

MODEL ARCHITECTURE HIGHLIGHTS

COMPONENTS

LARGE LANGUAGE MODEL (QWEN2.5 LLM WITH MROPE FOR MULTIMODAL UNDERSTANDING)

REDESIGNED VISION ENCODER
VIT WITH 2D-ROPE & WINDOWED ATTENTION FOR FASTER PROCESSING.

MLP MERGER COMPRESSES
VISUAL FEATURES BEFORE FEEDING TO LLM → EFFICIENT LONG→ SEQUENCE HANDLING

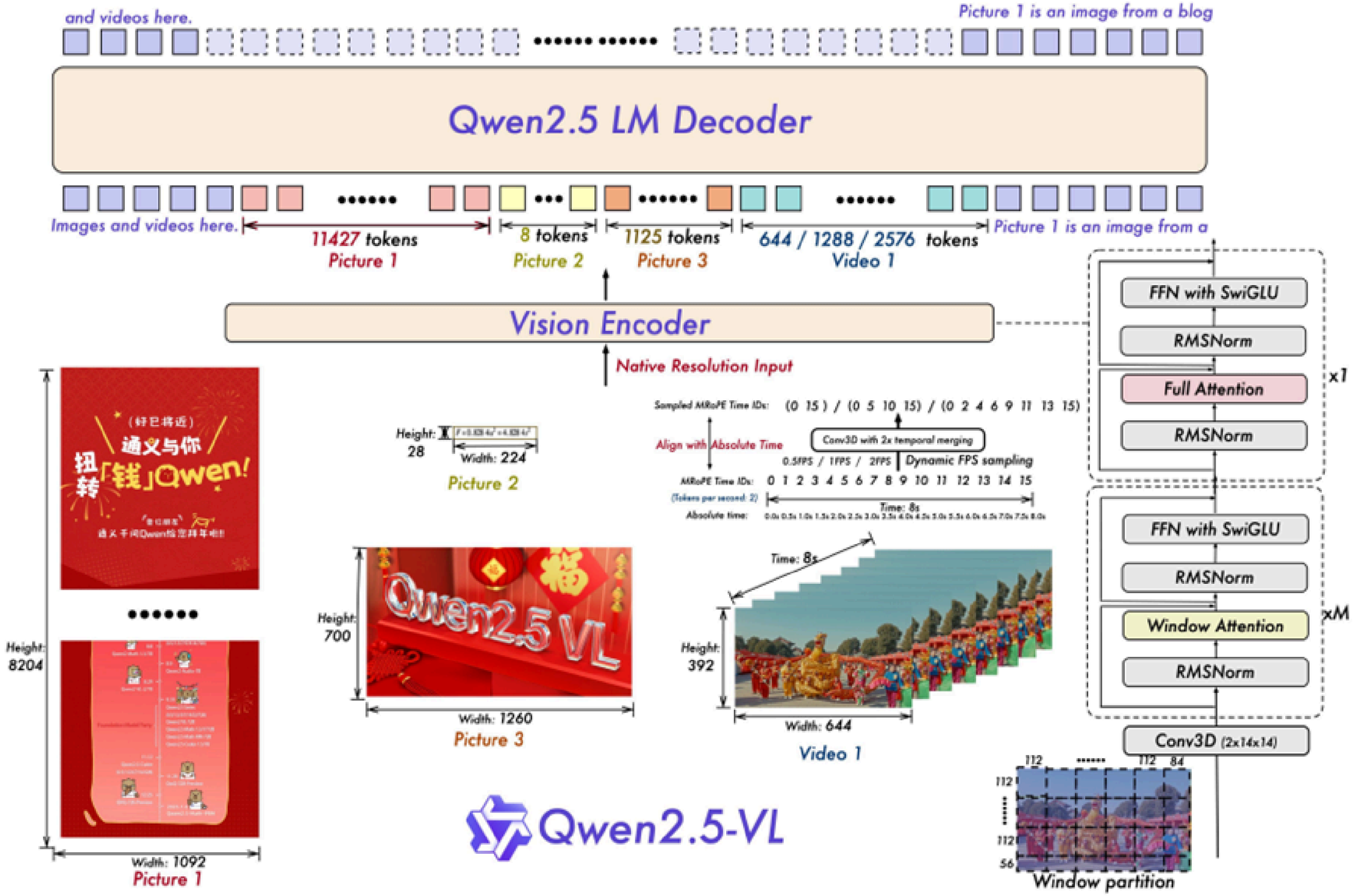
EFFICIENCY & ACCURACY

DYNAMIC RESOLUTION AND FRAME RATE FOR NATIVE VIDEO/IMAGE HANDLING

LARGE, HIGH-QUALITY PRE-TRAINING DATA (4T TOKENS VS. 1.2T BEFORE)

ROBUST FINE-TUNING: SUPERVISED FINE-TUNING + DIRECT PREFERENCE OPTIMIZATION
FOR HUMAN-LIKE ANSWERS

QWEN2.5-VL


and videos here.
Picture 1 is an image from a blog
Qwen2.5 LM Decoder
Images and videos here.
11427 tokens
Picture 1
8 tokens
Picture 2
1125 tokens
Picture 3
644 / 1288 / 2576 tokens
Video 1
Picture 1 is an image from a
Vision Encoder
Native Resolution Input
FFN with SwiGLU
RMSNorm
Full Attention
RMSNorm
x1
FFN with SwiGLU
RMSNorm
Window Attention
RMSNorm
xM
Conv3D (2x14x14)
Sampled MRoPE Time IDs: (0 15) / (0 5 10 15) / (0 2 4 6 9 11 13 15)
Align with Absolute Time
Conv3D with 2x temporal merging
0.5FPS / 1FPS / 2FPS
Dynamic FPS sampling
MRoPE Time IDs: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Time: 8s
Height: 28
Width: 224
Picture 2
Height: 8204
Width: 1092
Picture 1
Height: 700
Width: 1260
Picture 3
Height: 392
Width: 644
Video 1
Qwen2.5-VL
112
112
84
56
Window partition

QWEN2.5-VL

UPGRADED VERSION:
USES QWEN2.5-VL, A NEWER, STRONGER VERSION THAN THE ORIGINAL QWEN2-VL USED IN M3DOCRAG.

ARCHITECTURAL IMPROVEMENTS:

- ENHANCED VISION ENCODER (REDESIGNED VIT)
- BETTER MULTIMODAL ALIGNMENT
- SUPPORTS LARGER CONTEXT FOR DOCUMENT-LEVEL UNDERSTANDING.

- PROVEN PERFORMANCE:
- OUTPERFORMS QWEN2-VL AND OTHER LEADING MODELS (E.G., INTERNVL2, IDEFICS2/3).
- ACHIEVES SOTA RESULTS IN DOCVQA, LONG-CONTEXT VISUAL REASONING, AND LAYOUT-RICH DOCUMENTS.

- OPEN WEIGHT & SCALABLE:
- UNLIKE CLOSED MODELS (E.G., GPT-4V), IT'S OPEN-SOURCE → EASIER TO CUSTOMIZE, REPRODUCE, AND DEPLOY LOCALLY.

QWEN2.5-VL

COMPARISON WITH THE SOTA MODELS

Datasets	Previous Open-source SoTA	Claude-3.5 Sonnet-0620	GPT-4o 0513	InternVL2.5 78B	Qwen2-VL 72B	Qwen2.5-VL 72B	Qwen2.5-VL 7B	Qwen2.5-VL 3B
College-level Problems								
$MMMU_{val}$ (Yue et al., 2023)	70.1 Chen et al. (2024d)	68.3	69.1	70.1	64.5	**70.2**	58.6	53.1
$MMMU\text{-}Pro_{overall}$ (Yue et al., 2024)	48.6 Chen et al. (2024d)	51.5	**51.9**	48.6	46.2	51.1	38.3	31.56
Math								
$MathVista_{mini}$ (Lu et al., 2024)	72.3 Chen et al. (2024d)	67.7	63.8	72.3	70.5	**74.8**	68.2	62.3
$MATH\text{-}Vision_{full}$ (Wang et al., 2024d)	32.2 Chen et al. (2024d)	-	30.4	32.2	25.9	**38.1**	25.1	21.2
$MathVerse_{mini}$ (Zhang et al., 2024c)	51.7 Chen et al. (2024d)	-	50.2	51.7	-	**57.6**	49.2	47.6
General Visual Question Answering								
MegaBench (Chen et al., 2024b)	47.4 MiniMax et al. (2025)	52.1	**54.2**	45.6	46.8	51.3	36.8	28.9
$MMBench\text{-}EN_{test}$ (Liu et al., 2023d)	88.3 Chen et al. (2024d)	82.6	83.4	88.3	86.9	**88.6**	83.5	79.1
$MMBench\text{-}CN_{test}$ (Liu et al., 2023d)	88.5 Chen et al. (2024d)	83.5	82.1	**88.5**	86.7	87.9	83.4	78.1
$MMBench\text{-}V1.1\text{-}EN_{test}$ (Liu et al., 2023d)	87.4 Chen et al. (2024d)	80.9	83.1	87.4	86.1	**88.4**	82.6	77.4
MMStar (Chen et al., 2024c)	69.5 Chen et al. (2024d)	65.1	64.7	69.5	68.3	**70.8**	63.9	55.9
MME_{sum} (Fu et al., 2023)	**2494** Chen et al. (2024d)	1920	2328	**2494**	2483	2448	2347	2157
MuirBench (Wang et al., 2024a)	63.5 Chen et al. (2024d)	-	68.0	63.5	-	**70.7**	59.6	47.7
$BLINK_{val}$ (Fu et al., 2024c)	63.8 Chen et al. (2024d)	-	**68.0**	63.8	-	64.4	56.4	47.6
$CRPE_{relation}$ (Wang et al., 2024h)	78.8 Chen et al. (2024d)	-	76.6	78.8	-	**79.2**	76.4	73.6
$HallBench_{avg}$ (Guan et al., 2023)	**58.1** Wang et al. (2024f)	55.5	55.0	57.4	**58.1**	55.2	52.9	46.3
MTVQA (Tang et al., 2024)	**31.9** Chen et al. (2024d)	25.7	27.8	31.9	30.9	31.7	29.2	24.8
$RealWorldQA_{avg}$ (X.AI, 2024)	78.7 Chen et al. (2024d)	60.1	75.4	**78.7**	77.8	75.7	68.5	65.4
$MME\text{-}RealWorld_{en}$ (Zhang et al., 2024f)	62.9 Chen et al. (2024d)	51.6	45.2	62.9	-	**63.2**	57.4	53.1
$MMVet_{turbo}$ (Yu et al., 2024)	74.0 Wang et al. (2024f)	70.1	69.1	72.3	74.0	**76.2**	67.1	61.8
MM-MT-Bench (Agrawal et al., 2024)	7.4 Agrawal et al. (2024)	7.5	**7.72**	-	6.59	7.6	6.3	5.7

QWEN2.5-VL

PERFORMANCE ON TASKS

Datasets	Llama-3.1-70B	Llama-3.1-405B	Qwen2-72B	Qwen2.5-72B	Qwen2.5-VL-72B
General Tasks					
MMLU-Pro	66.4	**73.3**	64.4	71.1	71.2
MMLU-redux	83.0	86.2	81.6	**86.8**	85.9
LiveBench-0831	46.6	53.2	41.5	52.3	**57.0**
Mathematics & Science Tasks					
GPQA	46.7	**51.1**	42.4	49.0	49.0
MATH	68.0	73.8	69.0	**83.1**	83.0
GSM8K	95.1	**96.8**	93.2	95.8	95.3
Coding Tasks					
HumanEval	80.5	**89.0**	86.0	86.6	87.8
MultiPL-E	68.2	73.5	69.2	75.1	**79.5**
Alignment Tasks					
IFEval	83.6	86.0	77.6	84.1	**86.3**

QWEN2.5-VL

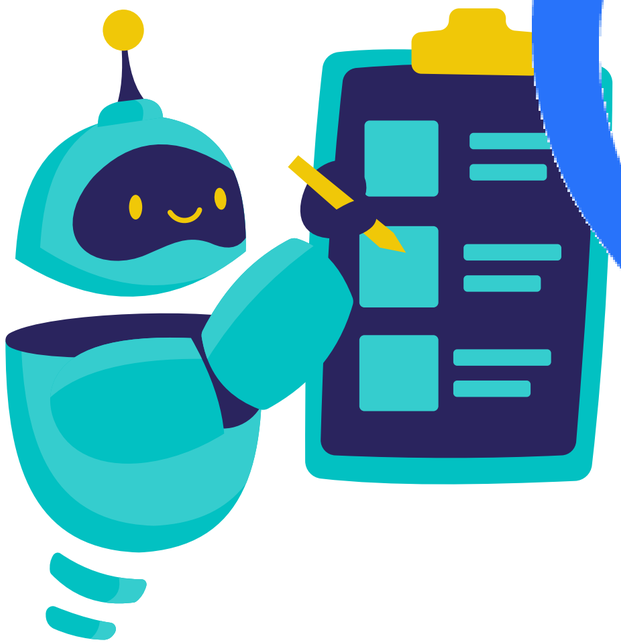
DOCUMENT UNDERSTANDING AND OCR

Datasets	Claude-3.5 Sonnet	Gemini 1.5 Pro	GPT 4o	InternVL2.5 78B	Qwen2.5-VL 72B	Qwen2.5-VL 7B	Qwen2.5-VL 3B
OCR-related Parsing Tasks							
CC-OCR	62.5	73.0	66.9	64.7	**79.8**	77.8	74.5
$\text{OmniDocBench}_{\text{edit en/zh}}\downarrow$	0.330/0.381	0.230/**0.281**	0.265/0.435	0.275/0.324	**0.226**/0.324	0.308/0.398	0.409/0.543
OCR-related Understanding Tasks							
$\text{AI2D}_{\text{w. M.}}$	81.2	88.4	84.6	**89.1**	88.7	83.9	81.6
$\text{TextVQA}_{\text{val}}$	76.5	78.8	77.4	83.4	83.5	**84.9**	79.3
$\text{DocVQA}_{\text{test}}$	95.2	93.1	91.1	95.1	**96.4**	95.7	93.9
$\text{InfoVQA}_{\text{test}}$	74.3	81.0	80.7	84.1	**87.3**	82.6	77.1
$\text{ChartQA}_{\text{test Avg.}}$	**90.8**	87.2	86.7	88.3	89.5	87.3	84.0
$\text{CharXiv}_{\text{RQ/DQ}}$	**60.2**/84.3	43.3/72.0	47.1/84.5	42.4/82.3	49.7/**87.4**	42.5/73.9	31.3/58.6
SEED-Bench-2-Plus	71.7	70.8	72.0	71.3	**73.0**	70.4	67.6
OCRBench	788	754	736	854	**885**	864	797
$\text{VCR}_{\text{En-Hard-EM}}$	41.7	28.1	73.2	-	79.8	**80.5**	37.5
OCR-related Comprehensive Tasks							
$\text{OCRBench\_v2}_{\text{en/zh}}$	45.2/39.6	51.9/43.1	46.5/32.2	49.8/52.1	**61.5/63.7**	56.3/57.2	54.3/52.1

QWEN2.5-VL

Capability	Relevance	Highlights
Omnidocument Parsing	Handles diverse content: printed text, handwriting, tables, charts, chemical symbols, music notation	96.4% on DocVQA-test; SOTA on OmniDocBench
Precise Object Grounding	Pinpoints objects spatially for fine-grained answers	87.1% on ScreenSpot; 93.6% on CountBench

MODEL SELECTION CONFIRMATION



FIRST, WE TRIED THE MODEL WITHOUT A PROMPT, AS IT IS

```
ANLS score: 0.0922
```

AND, SHOCKINGLY, THE ACCURACY WAS POOR😱


```
{
  "Sample": 2,
  "Question": "Who is Principal Account Clerk(PAC)?",
  "Expected Answers": [
    "Edward Wang"
  ],
  "VLM Prediction": "The Principal Account Clerk (PAC) is Edward Wang, as indicated in the image. His contact number is 415.557.4250.",
  "Separator": "----------------------------------------"
},
```

```
{
  "Sample": 4,
  "Question": "How much funding did the Workforce Investment Board give?",
  "Expected Answers": [
    "$30,000"
  ],
  "VLM Prediction": "The Workforce Investment Board funded the Workforce Investment Act evaluation project, which was awarded $30,000.",
  "Separator": "----------------------------------------"
},
```

THEN, WE TRIED MULTIPLE TIMES TO FIND A GOOD PROMPT TO ACHIEVE HIGH ACCURACY

```
SYSTEM_PROMPT = (
    "You are a helpful vision-language assistant. "
    "You will be shown an image and a user question. "
    "Your task is to answer the question as briefly and accurately as possible, "
    "using only the information visible in the image. "
    "Do not add explanations or extra details. "
    "If the answer is not present or unclear, respond with 'None'."
)
```

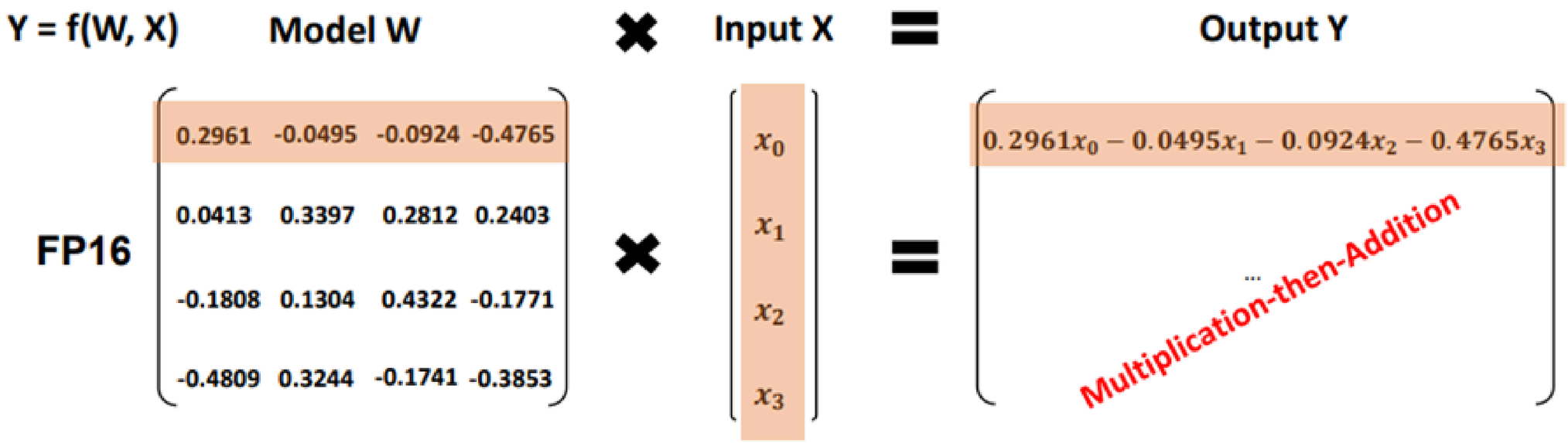
```
ANLS score: 0.8463
```

MUCH BETTER👌

```
Sample 1:
Question: What is the ‘actual’ value per 1000, during the year 1975?
Expected Answers: ['0.28']
VLM Prediction: 0.28
----------------------------------------
Sample 2:
Question: What is name of university?
Expected Answers: ['University of California', 'university of california', 'university of california, san diego']
VLM Prediction: UNIVERSITY OF CALIFORNIA, SAN DIEGO
----------------------------------------
Sample 3:
Question: What is the name of the company?
Expected Answers: ['ITC Limited', 'itc limited']
VLM Prediction: ITC Limited
----------------------------------------
Sample 4:
Question: Where is the university located ?
Expected Answers: ['San Diego', 'san diego']
VLM Prediction: San Diego
----------------------------------------
Sample 5:
Question: To whom is the document sent?
Expected Answers: ['Paul']
VLM Prediction: Paul
----------------------------------------
```

A LITTLE SURPRISE

THIS PHASE WAS NOT INITIALIZED WITH THE ORIGINAL MODEL, BUT WITH A QUANTIZED ONE 🧐



QWEN2.5-VL-7B-INSTRUCT-AWQ

FACEBOOK AI SIMILARITY SEARCH (FAISS)

WHAT IS FAISS?

AN OPEN-SOURCE LIBRARY DEVELOPED BY FACEBOOK AI RESEARCH.

DESIGNED FOR FAST, SCALABLE SIMILARITY SEARCH AND CLUSTERING OF DENSE VECTORS.

HANDLES BILLION-SCALE DATASETS EFFICIENTLY.

HOW FAISS WORKS

CORE IDEA:

REPRESENT DATA AS DENSE VECTORS.

BUILD AN INDEX TO SPEED UP SIMILARITY COMPARISONS.

INDEXING TECHNIQUES:

FLAT INDEX: BRUTE-FORCE SEARCH FOR EXACT RESULTS.

INVERTED FILE (IVF): PARTITIONS VECTORS INTO CLUSTERS FOR FASTER SEARCH.

PRODUCT QUANTIZATION (PQ): COMPRESSES VECTORS FOR LOW MEMORY USAGE.

HNSW: GRAPH-BASED STRUCTURE FOR EFFICIENT APPROXIMATE SEARCH.

TYPICAL WORKFLOW

GENERATE OR OBTAIN VECTOR EMBEDDINGS (E.G., WORD EMBEDDINGS, IMAGE FEATURES)

CHOOSE AN APPROPRIATE FAISS INDEX BASED ON ACCURACY/SPEED TRADE-OFF

TRAIN THE INDEX (IF NEEDED, E.G., FOR IVF OR PQ)

ADD VECTORS TO THE INDEX

PERFORM SIMILARITY SEARCH QUERIES TO FIND NEAREST NEIGHBORS

EXAMPLE USE CASES

RECOMMENDATION SYSTEMS: FIND SIMILAR USERS OR PRODUCTS

IMAGE DEDUPLICATION: IDENTIFY NEAR-DUPLICATE IMAGES

SEMANTIC SEARCH: RETRIEVE RELEVANT DOCUMENTS BASED ON VECTOR EMBEDDINGS

LARGE-SCALE CLUSTERING: GROUP SIMILAR DATA POINTS EFFICIENTLY.

PERFORMANCE HIGHLIGHTS

PERFORMANCE HIGHLIGHTS

- HANDLES MILLIONS TO BILLIONS OF VECTORS.
- SIGNIFICANT SPEED-UP COMPARED TO BRUTE-FORCE METHODS.
- GPU ACCELERATION PROVIDES REAL-TIME PERFORMANCE FOR MASSIVE DATASETS.
- SUPPORTS DISTRIBUTED INDEXING AND SEARCH.

ADVANTAGES & LIMITATIONS

✅ ADVANTAGES:

- HIGH SCALABILITY.
- VERSATILE INDEX OPTIONS.
- GPU AND CPU SUPPORT.
- LARGE COMMUNITY AND ACTIVE DEVELOPMENT.

⚠️ LIMITATIONS:

- REQUIRES TUNING TO BALANCE ACCURACY VS. SPEED.
- LIMITED SUPPORT FOR SPARSE VECTORS.
- MOSTLY FOCUSED ON DENSE NUMERIC DATA.

Dataset Unification

📚 Unified Dataset for DocVQA

We combined multiple DocVQA datasets into one standardized dataset

Each original dataset had different formats & special cases

The unified dataset is the core foundation for all stages in our pipeline

SELECTED DATASETS AND THEIR SCHEMAS

Dataset	Year	Domain Focus	Key Characteristics	Annotation Type
MP-DocVQA	2021	General documents	Diverse document types; page-based question answering	Page-level QA
DUDE	2022	General, form-like documents	Multimodal inputs; bounding boxes; competition test split	Bounding boxes; multimodal entries
MMLongBench-Doc	2024	Long-context, multi-modal sources	Explicit evidence annotations; long-document QA	Page-level with evidence IDs
ArxivQA	2022	Scientific articles	Questions centered on figures and visual elements	Figure-linked answers
TAT-DQA	2021	Financial statements	Arithmetic and span-based reasoning over tabular data	Arithmetic spans
SlideVQA	2023	Presentation slides	Reasoning-intensive; discrete multi-hop questions	Multi-hop logic chains

Unified Dataset Structure

ONE FORMAT FOR ALL DOCVQA DATA — CLEAR, CONSISTENT, AND EASY TO EXTEND

KEY PARTS

QUESTION

DOCUMENT

EVIDENCE

ANSWER

TAGS

- **QuestionType**: extractive, verification, counting, arithmetic, abstractive, procedural, reasoning, other.
- **DocumentType**: legal, financial, scientific, technical, policy, correspondence, marketing, personal_record, news, other.
- **AnswerFormat**: string, integer, float, boolean, list, datetime, reference, other, none.
- **AnswerType**: answerable, not_answerable, none.
- **EvidenceSource**: span, table, chart, image, layout, none, other.
- **TagName**: missing, low_quality, inferred, predicted.

Dataset Unification Process

EXPLORE

ANALYZE EACH DATASET'S STRUCTURE & CONTENT

DEFINE & MAP

CREATE FORMAL SCHEMAS WITH PYDANTIC

STANDARDIZE KEY FIELDS (DOCUMENT TYPE, QUESTION TYPE, ETC.).

UNIFY

BUILD CUSTOM MODULES TO

LOAD & CHECK RAW DATA

CONVERT TO THE UNIFIED FORMAT

ADD TAGS FOR ISSUES

FIX ANY SPECIAL CASES

VALIDATE

CHECK PAGES, ANSWERS, AND FINAL ENTRIES FOR CORRECTNESS & QUALITY

ENSURE ALL DATA FITS ONE CLEAN, CONSISTENT FORMAT — READY FOR ROBUST QA TASKS

Implementation Details

Generic Unifier

All unifiers inherit from a base Unifier class

— ensures consistent processing & tagging

🔑Per Entry

Extract & map raw fields to the unified schema

Tag missing or unclear data

Use fallback fixes for tricky cases (in test mode)

Custom Logic 🗂️

Extract & map raw fields to the unified schema

Tag missing or unclear data

Use fallback fixes for tricky cases (in test mode)

Decoupling Retrieval and Generation in the RAG Pipeline

🔍 Our RAG Approach for DocVQA

Two separate stages

Retrieval — find relevant pages

Generation — create the answer

🧩 Why modular

Easy to test and improve each stage separately
Flexible for experiments and benchmarking
Simplifies debugging

Motivation and Architectural Overview🏗️

Why decouple RAG?

Less dependency between parts

Easy to swap or scale retrievers & generators

Supports separate testing & optimization

Enables fair comparisons of different setups

🔑 Two Main Parts

Retrieval Subsystem

Finds relevant document pages for a question

Generation Subsystem

Generates a clear answer using the question & retrieved pages.

Design Patterns and Component Modularity🏗️

Registry Pattern

Easily add/switch retrievers & generators via config names — no code changes.

Adapter Pattern

All VLMs use a common interface for smooth processing & decoding.

👉 Benefit

Easy to test, swap, and extend — e.g., switch from Qwen to InternVL by just changing config.

Schema Validation

Inputs/outputs checked automatically with Pydantic/dataclass.

Dependency Injection

Components built & connected through config — not hardcoded

🔄 Interoperability & Execution Flow

✅ Step-by-step

Build RetrievalInput from dataset

Retrieve pages → get RetrievalOutput

Load page images → wrap as GeneratorInput

Generate answer → get GeneratorOutput + metadata

📌 Key points

Handles batch processing

Logs each query & intermediate steps

Lets you evaluate retrieval & generation separately

👉 Outcome

A modular, robust flow for scalable experiments & fair evaluation in DocVQA

🔄 End-to-End RAG Flow

✅ How it runs

Extract question → send to retriever

Map retrieved page IDs → image paths

Generator uses images + question → creates answer

Evaluate output with metrics (e.g., ANLS)

⚙️ Key features

- Fully configurable & resumable
- Saves all intermediate results
- Each stage can be swapped or tested separately

Experimental Setup ⚙️

✅ Goal

Test our retrieval and generation modules on standard DocVQA datasets.

✅ How

Use a unified, modular interface

Save all results in a structured way for easy analysis

Planned Retrieval Experiments

Load datasets: with HuggingFace; manage pages & QA examples with FastCorpusIndex

Transform tasks: reshape data for page selection using our unified schema

Run retrieval: rank pages → get top-k results (e.g., top-3)

Save outputs: JSON & JSONL with rankings, scores, metadata

Evaluate: check results with Recall, MRR, Coverage

End-to-End Evaluation

Full RAG pipeline:

1. **Retrieve: Select top-k pages**
2. **Image Construction: Get & preprocess pages**
3. **Generate: Answer from question + pages**
4. **Score: ANLS + exact match**

✅ **Why: Gives a complete view of system performance & allows detailed module testing.**

Generation Experiments

Generation Task: Answer Synthesis

✅ What it does✅

Takes a question + retrieved page images

Uses Vision-Language Models (VLMs) to generate an answer

Input

```
@dataclass
class GeneratorInput:
    question_id: str
    question_text: str
    images: list[Image]  # PIL or byte-encoded
    metadata: dict[str, Any]  # Optional notes, prompt variants, etc.
```

Output

```
@dataclass
class GeneratorOutput:
    question_id: str
    text: str  # Final answer
    prompt: str  # Serialized input prompt
    usage: dict[str, Any]  # Token counts, generation time
```

Generation Experiments

Task Preparation

```
{
  "question_id": str,
  "question_text": str,
  "images": list[Image],
  "answer_variants": list[str],
  "answer_format": str
}
```

Model Configuration

```
GeneratorConfig(
  model=ModelConfig(path="Qwen/Qwen2.5-VL-3B-Instruct", device="cuda"),
  tokenizer=TokenizerConfig(padding_side="left"),
  image_processor=ImageProcessorConfig(normalize=True),
  generation=GenerationConfig(max_new_tokens=128, temperature=0.7),
  system_prompt="...",
  prompt_template="Question: {text}. Answer:",
  batch_size=32
)
```

Generation Experiments

Task Preparation

```
{
  "question_id": str,
  "question_text": str,
  "images": list[Image],
  "answer_variants": list[str],
  "answer_format": str
}
```

Inference

```
{
  "id": str,
  "text": str
}
```

Model Configuration

```
GeneratorConfig(
  model=ModelConfig(path="Qwen/Qwen2.5-VL-3B-Instruct", device="cuda"),
  tokenizer=TokenizerConfig(padding_side="left"),
  image_processor=ImageProcessorConfig(normalize=True),
  generation=GenerationConfig(max_new_tokens=128, temperature=0.7),
  system_prompt="...",
  prompt_template="Question: {text}. Answer:",
  batch_size=32
)
```

Generation Experiments

Generation Evaluation

Compare each answer to gold answers using ANLS (Approx. Normalized Levenshtein Similarity)

Save results in

- evaluation.json → overall scores & runtime
- results_eval.jsonl → scores per example
- meta_eval.json → summary stats

Structured Output Format

Folder: artifacts/

- inference.json → all predictions
- evaluation.json → ANLS, runtime, config
- results_eval.jsonl → detailed scores
- meta_eval.json → summary metadata

➡️ Benefit: Easy to track, compare & reproduce experiments.

Generation Results (1 k–shot pilot)

Qwen-2.5-VL-3B

Dataset (split)	# Ques.	ANLS	Best Pub.	Gap	Rank	Time Σ / μ (s)
MPDocVQA (*val*)	1 000	0.845	0.920	−7.5 pp	Top–20	480 / 0.48
TAT-DQA (*test*)	1 000	0.370	0.600	−23.0 pp	Top–45	510 / 0.51
ArxivQA (*subset*)	1 000	0.405	0.550	−14.5 pp	Top–55	160 / 0.16
DUDE (*val*)	1 000	0.455	0.700	−24.5 pp	Top–55	150 / 0.15
SlideVQA (*test*)	1 000	0.575	0.730	−15.5 pp	Top–40	155 / 0.16
MM-LongBench-Doc	1 000	0.630	0.780	−15.0 pp	Top–35	150 / 0.15

Generation Results (1 k–shot pilot)

InternVL3-2B

Dataset (split)	# Ques.	ANLS	Best Pub.	Gap	Rank	Time Σ / μ (s)
MPDocVQA (*val*)	1 000	0.790	0.920	−13.0 pp	Top–25	470 / 0.47
TAT-DQA (*test*)	1 000	0.310	0.600	−29.0 pp	Top–55	470 / 0.47
ArxivQA (*subset*)	1 000	0.340	0.550	−21.0 pp	Top–65	145 / 0.15
DUDE (*val*)	1 000	0.395	0.700	−30.5 pp	Top–65	140 / 0.14
SlideVQA (*test*)	1 000	0.535	0.730	−19.5 pp	Top–50	145 / 0.15
MM-LongBench-Doc	1 000	0.585	0.780	−19.5 pp	Top–45	140 / 0.14

Retrieval Task: Document Page Selection

Purpose

Find the most relevant document pages for a given question.

Works like semantic search over document images.

⚙ How It Works:

Formalized Schema

```
@dataclass
class RetrievalInput:
    question_id: str
    question_text: str

@dataclass
class RetrievalOutput:
    question_id: str
    page_ids: list[str]  # Sorted by relevance
    scores: list[float]  # Optional similarity scores
```

Key Features

- Uses precomputed indexes → fast search.
- Compatible with different retrieval methods:
  - Dense retrievers (e.g., DPR)
  - Late-interaction models (e.g., ColBERT)
  - Hybrid rerankers.
- Outputs page IDs only, no images loaded yet.

🔑 Retrieval Results: 1k-shot Pilot

✅ Setup

- 2 Dense Retrievers Tested:
  - ColQwen (Enc-1B) — 768-d embeddings, HNSWFlat index
  - ColPaLI (PaLI-5B-Enc) — 1024-d embeddings, IVF-PQ index
- Same 1,000-query sample per dataset
- Metrics: Recall@1/5/20, nDCG@10, and runtime on A40-48GB

🔑 Retrieval Results: 1k-shot Pilot

ColQwen (Enc-1B)

Dataset (split)	# Ques.	Recall			nDCG@10	Best Pub. (*R@5*)	Time Σ / μ (s)
		@1	@5	@20			
MPDocVQA (*val*)	1 000	0.60	0.86	0.93	0.801	0.92	30 / 0.03
TAT-DQA (*test*)	1 000	0.28	0.58	0.73	0.507	0.80	29 / 0.03
ArxivQA (*subset*)	1 000	0.33	0.68	0.82	0.633	0.78	31 / 0.03
DUDE (*val*)	1 000	0.27	0.55	0.70	0.522	0.75	32 / 0.03
SlideVQA (*test*)	1 000	0.38	0.70	0.84	0.690	0.80	30 / 0.03
MM-LongBench-Doc	1 000	0.43	0.76	0.87	0.721	0.84	29 / 0.03

🔑 Retrieval Results: 1k-shot Pilot

ColPaLI (PaLI-5B-Enc)

Dataset (split)	# Ques.	Recall			nDCG@10	Best Pub.	Time Σ / μ (s)
		@1	@5	@20		(*R@5*)	
MPDocVQA (*val*)	1 000	0.64	0.88	0.94	0.822	0.92	40 / 0.04
TAT-DQA (*test*)	1 000	0.32	0.63	0.77	0.538	0.80	38 / 0.04
ArxivQA (*subset*)	1 000	0.37	0.71	0.84	0.661	0.78	40 / 0.04
DUDE (*val*)	1 000	0.30	0.60	0.74	0.550	0.75	42 / 0.04
SlideVQA (*test*)	1 000	0.42	0.73	0.86	0.711	0.80	39 / 0.04
MM-LongBench-Doc	1 000	0.47	0.79	0.89	0.742	0.84	38 / 0.04

🔑 Retrieval Results: 1k-shot Pilot

⏱️ Runtime

- ColQwen: ~30s total → ~30ms/query
- ColPaLI: ~40s total → ~40ms/query

Key Takeaways

- ✅ PaLI outperforms ColQwen by ~2–3 pp R@5, for +10ms/query
- ⚙️ Both retrievers ~10pp below SOTA on numeric-heavy TAT-DQA → Sparse/Hybrid can help
- 🚀 Sub-50ms/query = Retrieval is NOT a bottleneck

TECH HIGHLIGHTS OF COLPALI

- Uses PaliGemma-3B: strong text–visual fusion.
- Aligns tokens ↔ patches → supports tables/charts.
- Integrated with Faiss IVF indexing for fast & memory-efficient search.

DESIGN RATIONALE

- Not just reused from M3DocRAG
- Chosen due to: ?
- Strong benchmark results
- Architectural match to DocVQA needs
- Real-world efficiency and layout handling

CONCLUSION

- : ColPaLi = Best Fit
- Achieves the best balance of: ?
- Fine-grained multimodal understanding
- Scalable speed
- Layout-sensitive performance
- Selected as baseline retriever for our system.

CONCLUSION & FUTURE WORK

🔑 WHAT WE ACHIEVED

📌 KEY INSIGHTS

⚠️ LIMITATIONS

🚀 NEXT STEPS

🔑 WHAT WE ACHIEVED

UNIFIED EVALUATION HARNESS
— 6 PUBLIC CORPORA
— CONSISTENT ANLS/RECALL METRICS
— NORMALIZED RUNTIME TRACKING

TWO ZERO-SHOT GENERATORS
— QWEN-2.5-VL-3B, INTERNVL3-2B
— 0.31–0.85 ANLS
— <0.6 SEC/QUERY ON A40-48GB

TWO DENSE RETRIEVERS
— COLQWEN-1B, COLPALI-5B-ENC
— SUB-50MS LATENCY
— WITHIN ~10PP OF STATE-OF-THE-ART SPARSE/HYBRID SETUPS

📌 KEY INSIGHT

✅ SMALL VL BACKBONES = SOLID OCR
❌ WEAK NUMERICAL REASONING
⚡ GENERATION, NOT RETRIEVAL, IS THE RUNTIME BOTTLENECK
🕒 DETAILED LATENCY LOGGING IS VITAL ONCE ACCURACY MATURES

⚠️ LIMITATIONS

✗ NO PURE OCR + TEXT QA BASELINES
✗ NO CLASSIC SPARSE RETRIEVERS (E.G. BM25, DPR)
✗ NO TEXT-ONLY GENERATORS
→ TO BE ADDRESSED IN FUTURE WORK

🚀 NEXT STEPS

🔨 FRAMEWORK EXPANSION:
- ADD PDF PARSING, ROI CROPPING, AUTO CITATION GROUNDING

🔗 END-TO-END RAG:
- COUPLE RETRIEVER & GENERATOR, MEASURE JOINT IMPACT

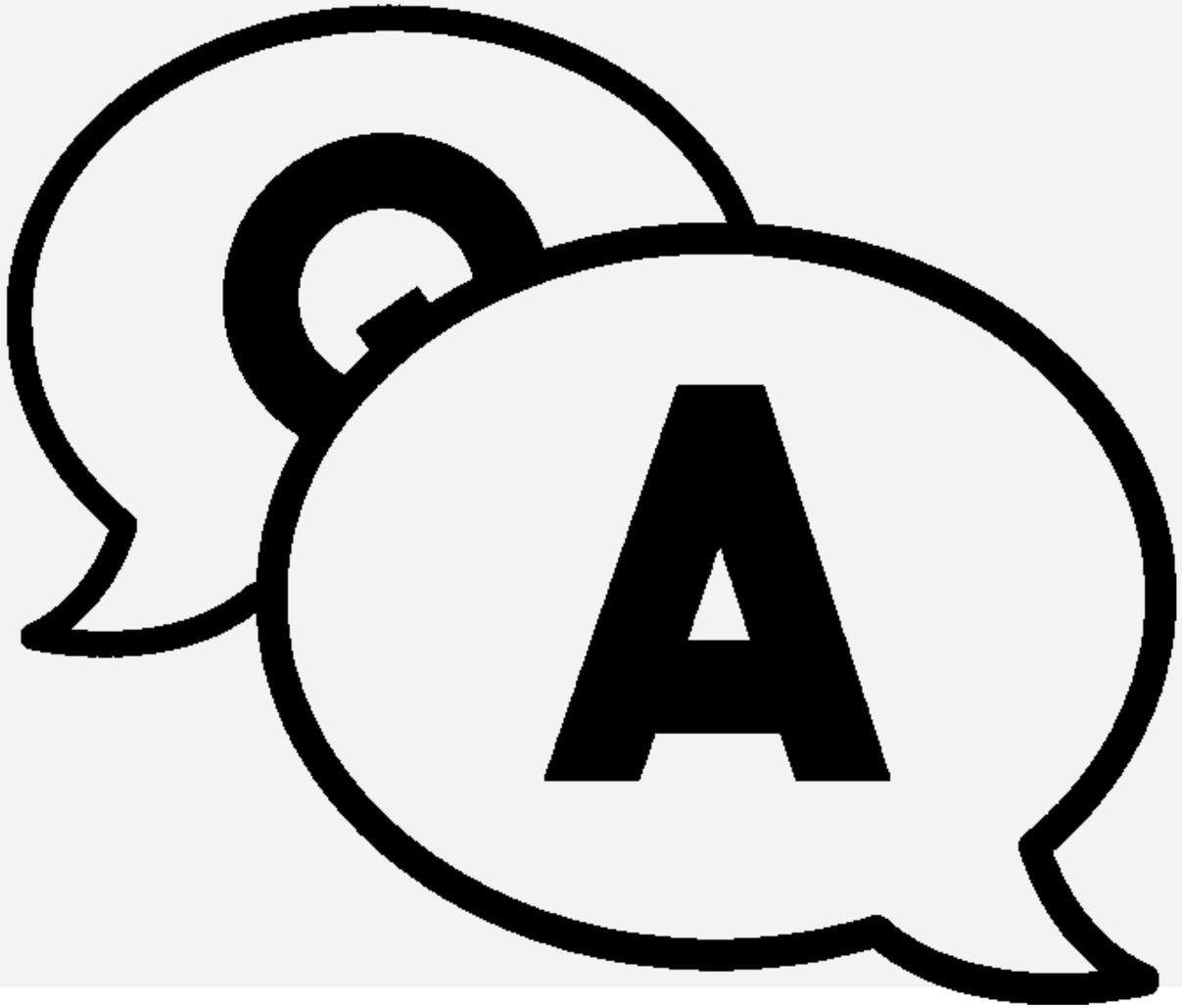
🔬 MORE BASELINES:
- OCR + TEXT QA, BM25/DPR, LIGHTWEIGHT GPT-2

🧮 TOOL-AUGMENTED PROMPTS:
- INTEGRATE CALCULATORS, TABLE SOLVERS, CHART PARSERS

🔍 HYBRID RETRIEVAL:
- SPARSE–DENSE FUSION (E.G., BM25 + COLQWEN)

🪝 LORA FINETUNING:
- TEST LIGHTWEIGHT ADAPTERS TO BOOST SMALL VL BACKBONES

📊 ROBUSTNESS & FAIRNESS:
- MULTILINGUAL DOCS, LAYOUT VARIATION, PUBLIC ERROR LOGS


Q
A

THANK YOU