

ColPali: Efficient Document Retrieval with Vision Language Models

Manuel Faysse^{* 1,3} Hugues Sibille^{*1,4} Tony Wu^{*1} Bilel Omrani¹
 Gautier Viaud¹ Céline Hudelot³ Pierre Colombo^{2,3}

¹Illuin Technology ²Equall.ai

³CentraleSupélec, Paris-Saclay ⁴ETH Zürich

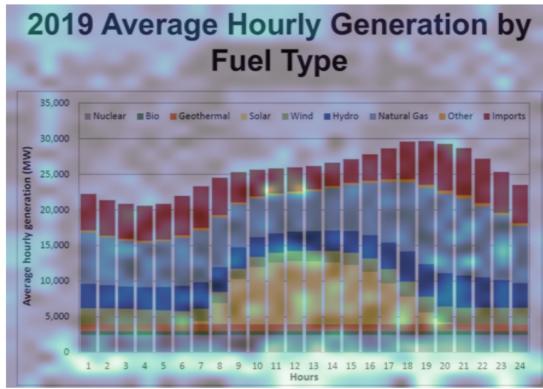
manuel.faysse@centralesupelec.fr

Abstract

Documents are visually rich structures that convey information through text, as well as tables, figures, page layouts, or fonts. While modern document retrieval systems exhibit strong performance on query-to-text matching, they struggle to exploit visual cues efficiently, hindering their performance on practical document retrieval applications such as Retrieval Augmented Generation. To benchmark current systems on visually rich document retrieval, we introduce the Visual Document Retrieval Benchmark *ViDoRe*, composed of various page-level retrieving tasks spanning multiple domains, languages, and settings. The inherent shortcomings of modern systems motivate the introduction of a new retrieval model architecture, *ColPali*, which leverages the document understanding capabilities of recent Vision Language Models to produce high-quality contextualized embeddings solely from images of document pages. Combined with a late interaction matching mechanism, *ColPali* largely outperforms modern document retrieval pipelines while being drastically faster and end-to-end trainable. We release all project artifacts at <https://huggingface.co/vidore>.

1 Introduction

Document Retrieval consists in matching a user query to relevant documents in a given corpus. It is central to many industrial applications, either as a standalone ranking system (search engines) or as part of more complex information extraction or Retrieval Augmented Generation (RAG) pipelines. Over recent years, pretrained language models have enabled large improvements in text embedding models. In practical industrial settings, however, the main performance bottleneck for efficient document retrieval is not in embedding model performance but in the prior data ingestion pipeline. To



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

Figure 1: For each term in a user query, *ColPali* identifies the most relevant document image patches (highlighted zones) and computes a query-to-page matching score. We can then swiftly retrieve the most relevant documents from a large pre-indexed corpus.

index a standard PDF document, many steps are required. First, PDF parsers or Optical Character Recognition (OCR) systems are used to extract words from the pages. Document layout detection models can then be run to segment paragraphs, titles, and other page objects such as tables, figures, and headers. A chunking strategy is then defined to group text passages with some semantical coherence, and modern retrieval setups may even integrate a captioning step to describe visually rich elements in a natural language form, more suitable for embedding models. In our experiments (Table 2), we typically find that optimizing the ingestion pipeline yields much greater performance on visually rich document retrieval than optimizing the text embedding model.

Contribution 1: ViDoRe. In this work, we argue that document retrieval systems should not be evaluated solely on the capabilities of text embedding models (Bajaj et al., 2016; Thakur et al., 2021; Muennighoff et al., 2022), but should also

^{*}Equal Contribution 瓶颈在数据提取上

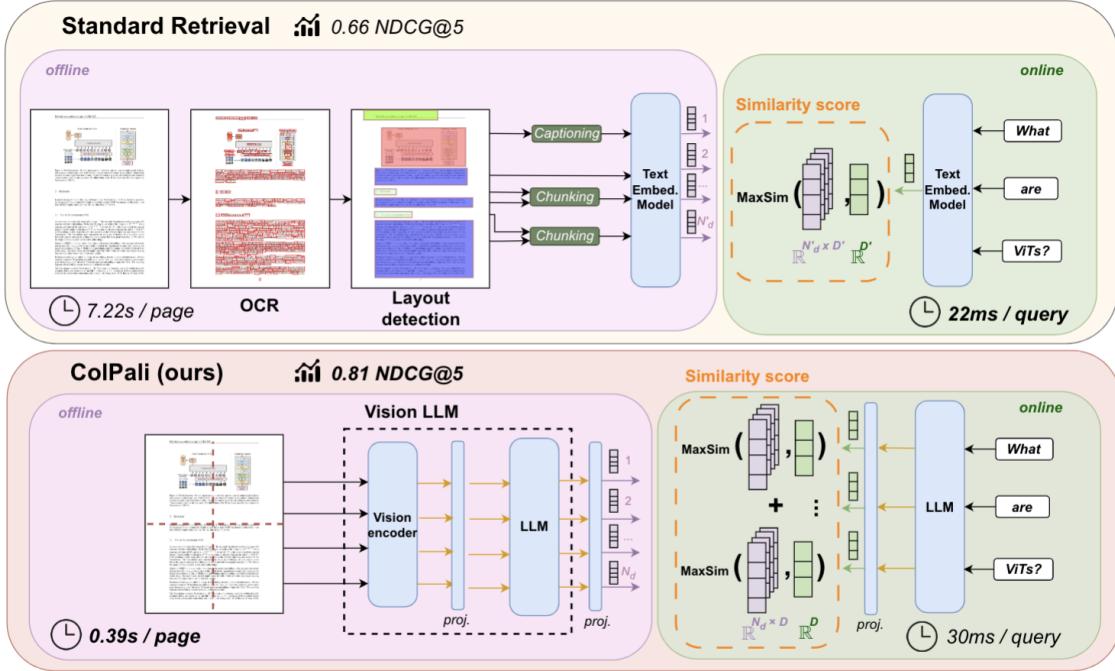


Figure 2: *ColPali* simplifies document retrieval w.r.t. standard retrieval methods while achieving stronger performances with better latencies. Latencies and results are detailed in section 5 and subsection B.5.

consider the context and visual elements of the documents to be retrieved. To this end, we create and openly release *ViDoRe*, a comprehensive benchmark to evaluate systems on page-level document retrieval with a wide coverage of domains, visual elements, and languages. *ViDoRe* targets practical document retrieval settings, in which user queries may require both textual and visual understanding to be correctly matched to relevant documents. We highlight the shortcomings of current text-centric systems in these settings.¹

Contribution 2: *ColPali*. We propose a novel model architecture and training strategy based on Vision Language Models (VLMs) to efficiently index documents purely from their visual features, allowing for subsequent fast query matching with late interaction mechanisms (Khattab and Zaharia, 2020). Our method, *ColPali*, outperforms all other retrieval systems on *ViDoRe* while being fast and end-to-end trainable. We release models and code at <https://huggingface.co/vidore>.

2 Problem Formulation & Related Work

Problem Setting. In our setting, a retrieval system scores how relevant a document d from corpus \mathcal{D} is

¹The benchmark leaderboard is hosted publicly at <https://huggingface.co/spaces/vidore/vidore-leaderboard> to encourage further developments.

with respect to a query q . Computing the similarity score $s(q, d) \in \mathbb{R}$ for each of the $|\mathcal{D}|$ documents in the corpus creates a ranking we can use to extract the most relevant documents. In this work, we focus on page-level retrieval: given a query, is the correct document page retrieved by the system? For coherence with existing literature, we further use the term *document* to refer to individual pages, i.e. the atomic retrieved elements in our setting. As we focus on practical industrial retrieval applications (RAG, search engines) with potentially large corpora sizes, latency constraints are imposed on scoring systems. Most current retrieval systems can be decomposed into (1) an *offline* indexation phase in which a document index is built and (2) an *online* querying phase in which a query is matched to documents from the index and where low latency is vital to the user experience.

Efficient document retrieval systems exhibit joint properties of high retrieval performance (R1), low latency during querying (R2), and high throughput during indexation (R3).

2.1 Textual Retrieval Methods

Document Retrieval in Text Space. Statistical methods based on word frequency like TF-IDF (Sparck Jones, 1972) and BM25 (Robertson et al., 1994) are still widely used due to their simplicity

and efficiency. More recently, neural embedding models based on fine-tuned large language models display state-of-the-art performance on a variety of text embedding tasks and top the retrieval leaderboards (Muennighoff et al., 2022).

Neural Retrievers. In bi-encoder models (Reimers and Gurevych, 2019; Karpukhin et al., 2020; Wang et al., 2022), documents are independently mapped *offline* to a dense vector space. Queries are embedded *online* and matched to documents through a fast cosine distance computation. A slower, but slightly more performant alternative, cross-encoder systems (Wang et al., 2020; Cohere, 2024) concatenate query and document as a single input sequence and iteratively attribute matching scores to each possible combination. This enables full attention computation between query and document terms but comes at the cost of computational efficiency, as $|\mathcal{D}|$ encoding passes must be done online.

Multi-Vector retrieval via late interaction. In the late interaction paradigm (Khattab and Zaharia, 2020), an embedding is pre-computed and indexed per document token. At runtime, similarity can be computed with individual query token embeddings. The idea is to benefit from the rich interaction between individual query and document terms while taking advantage of the offline computation and fast query matching enabled by bi-encoders.

Retrieval Evaluation. Although benchmarks and leaderboards have been developed to evaluate text embedding models (Thakur et al., 2021; Muennighoff et al., 2022), as previously stated, much of the performance improvements in industrial use cases of embedding models stem from the prior data ingestion pipeline. While documents often rely on visual elements to more efficiently convey information to human readers, text-only systems barely tap into these visual cues.

To our knowledge, no benchmark evaluates document retrieval methods by considering both textual and visual document features like a human would.

2.2 Integrating Visual features

Contrastive Vision Language Models. Mapping latent representations of textual content to corresponding representations of visual content has been done by aligning disjoint visual and text encoders through contrastive losses (Radford et al., 2021; Zhai et al., 2023). While some OCR capabilities exist in these models, the visual component is often not optimized for text understanding. The Fine-grained Interactive Language-Image Pre-training

(Yao et al., 2021) framework extends the late interaction mechanism to cross-modal vision-language models, relying on max similarity operations between text tokens and image patches.

Visually Rich Document Understanding. To go beyond text, some document-focused models jointly encode text tokens alongside visual or document layout features (Appalaraju et al., 2021; Kim et al., 2021; Huang et al., 2022; Tang et al., 2022). Large Language transformer Models (LLMs) with strong reasoning capabilities have recently been combined with Vision Transformers (ViTs) (Dosovitskiy et al., 2020) to create VLMs (Alayrac et al., 2022; Liu et al., 2023b; Bai et al., 2023; Laurençon et al., 2024) where image patch vectors from contrastively trained ViT models (Zhai et al., 2023) are fed as input embeddings to the language model and concatenated with the text-token embeddings.

PaliGemma. The PaliGemma-3B model (Beyer et al., 2024) extends concepts from Pali3 (Chen et al., 2023), and projects SigLIP-So400m/14 (Al-abdulmohsin et al., 2023) patch embeddings into Gemma-2B’s text vector space (Gemma Team et al., 2024). Along with its reasonable size w.r.t. other performant VLMs, an interesting property of PaliGemma’s text model is that it is fine-tuned with full-block attention on the prefix (instruction text and image tokens).

VLMs display enhanced capabilities in Visual Question Answering, captioning, and document understanding (Yue et al., 2023), but are not optimized for retrieval tasks.

3 The *ViDoRe* Benchmark

Existing benchmarks for contrastive vision-language models primarily evaluate retrieval for natural images (Lin et al., 2014; Borchmann et al., 2021; Thapliyal et al., 2022). On the other hand, textual retrieval benchmarks (Muennighoff et al., 2022) are evaluated at a textual passage level and are not tailored for document retrieval tasks. We fill the gap with *ViDoRe*, a comprehensive benchmark for document retrieval using visual features.

3.1 Benchmark Design

ViDoRe is designed to comprehensively evaluate retrieval systems on their capacity to match queries to relevant documents at the page level. This benchmark encompasses multiple orthogonal subtasks, with focuses on various modalities - text, figures, infographics, tables; thematic domains - medical,

正交的

business, scientific, administrative; or languages - English (eng), French (fra).

Dataset	# Queries	Domain
Academic Tasks		
DocVQA (eng)	500 (500)	Industrial
InfoVQA (eng)	500 (500)	Infographics
TAT-DQA (eng)	1600 (1600)	Varied Modalities
arXiVQA (eng)	500 (500)	Scientific Figures
TabFQuAD (fra)	210 (210)	Tables
Practical Tasks		
Energy (eng)	100 (1000)	Scientific
Government (eng)	100 (1000)	Administrative
Healthcare (eng)	100 (1000)	Medical
AI (eng)	100 (1000)	Scientific
Shift Project (fra)	100 (1000)	Environment

Table 1: *ViDoRe* comprehensively evaluates multimodal retrieval methods. The size of the document corpus is indicated in parentheses.

Academic Tasks. We repurpose widely used visual question-answering benchmarks for retrieval tasks: for each page-question-answer triplet, we use the question as the query, and the associated page as the gold document (Table 1). These academic datasets either focus on single specific modalities (Mathew et al., 2020, 2021; Li et al., 2024) or target more varied visually rich documents (Zhu et al., 2022). Moreover, we consider TabFQuAD, a human-labeled dataset on tables extracted from French industrial PDF documents released with this work. Details can be found in subsection A.1.

Practical tasks. We construct topic-specific retrieval benchmarks spanning multiple domains to go beyond repurposed QA datasets and evaluate retrieval in more realistic industrial situations (e.g. RAG). To achieve this, we collect publicly accessible PDF documents and generate queries pertaining to document pages using Claude-3 Sonnet, a high-quality proprietary vision-language model (Anthropic, 2024). In total, we collect 1,000 document pages per topic, which we associate with 100 queries extensively filtered for quality and relevance by human annotators. The corpus topics are intentionally specific to maximize syntactic proximity between documents, creating challenging retrieval tasks and covering an array of orthogonal domains (Table 1). Query-page pair examples are shown in Appendix E.²

Evaluation Metrics. We evaluate performance on our benchmark (Requirement R1) using standard

metrics from the retrieval literature (NDCG, Recall@K, MRR). We report NDCG@5 values as the main performance metric in this work and release the complete sets of results along with the models³. To validate compliance with practical industrial constraints, we also consider query latencies (R2) and indexing throughputs (R3).

3.2 Assessing Current Systems

Unstructured. We evaluate retrieval systems representative of those found in standard industrial RAG pipelines. As is common practice, we rely on the Unstructured⁴ off-the-shelf tool in the highest resolution settings to construct high-quality text chunks from PDF documents. Unstructured orches-trates the document parsing pipeline, relying on deep learning vision models to detect titles and document layouts (Ge et al., 2021), OCR engines (Smith, 2007) to extract text in non-native PDFs, specialized methods or models to detect and reconstruct tables, and implements a chunking strategy (by-title) that leverages the detected document structure to preserve section boundaries when concatenating texts. As is common practice, in our simplest Unstructured configuration (*text-only*), only textual elements are kept and figures, images, and tables are considered noisy information and are filtered out.

Unstructured + X. While Unstructured is a strong baseline by itself, we further augment Unstructured’s output by integrating the visual elements. In (+ OCR), tables, charts, and images are run through an OCR engine, processed by Unstructured, and chunked independently. In (+ Captioning), we set up a fully-fledged captioning strategy (Zhao et al., 2023), in which we feed visual elements to a strong proprietary Vision Language Model (Claude-3 Sonnet (Anthropic, 2024)) to obtain highly detailed textual descriptions of the elements. Both strategies aim to integrate visual elements in the retrieval pipeline but incur significant latency and resource costs (subsection 5.2).

Embedding Model. To embed textual chunks, we evaluate Okapi BM25, the *de facto* standard sparse statistical retrieval method, and the dense encoder of BGE-M3 (Chen et al., 2024), a multilingual neural method with SOTA performance in its size category. Chunks are embedded and scored independently, and page-level scores are obtained by

²Answers are generated alongside queries to (1) ground queries and improve their quality and (2) provide resources to foster future work.

³<https://huggingface.co/vidore>

⁴www.unstructured.io

策划，编排

max-pooling over the page’s chunk scores.⁵

Contrastive VLMs. We also evaluate the strongest available vision-language embedding models; Jina CLIP (Koukounas et al., 2024), Nomic Embed Vision (Nomic, 2024), and SigLIP-So400m/14 (Al-abdulmohsin et al., 2023).

Results. From a performance perspective, best results are obtained by combining the Unstructured parser with visual information, either from captioning strategies or by running OCR on the visual elements (Table 2). Little difference is seen between BM25 and BGE-M3 embeddings highlighting the visual information bottleneck. Contrastive VLMs lag behind. Beyond retrieval performance (R1), the indexing latencies (R2) reported in Figure 3 illustrate that PDF parsing pipelines can be very lengthy, especially when incorporating OCR or captioning strategies. Querying latencies at runtime (R3) are very good for all evaluated systems (≤ 22 ms on NVIDIA L4) due to fast query encoding and cosine similarity matching.

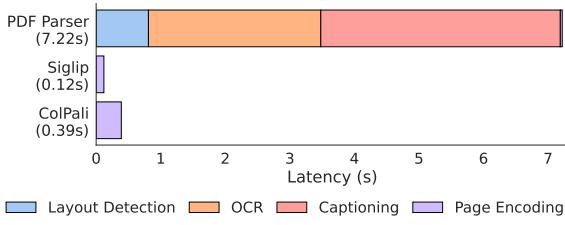


Figure 3: Offline indexing with *ColPali* is much simpler and faster compared to standard retrieval methods. Indexing speeds reported are computed on Nvidia L4 GPUs and detailed in subsection B.5.

4 Late interaction based Vision Retrieval

4.1 Architecture

Vision-Language Models. Encouraged by their strong document understanding capabilities, we propose adapting recent VLMs for retrieval. The key concept is to leverage the alignment between output embeddings of text and image tokens acquired during multi-modal finetuning. To this extent, we introduce **ColPali**, a Paligemma-3B extension that is capable of generating ColBERT-style multi-vector representations of text and images (Figure 2). PaliGemma-3B is a strong candidate due to its small size, the many released checkpoints fine-tuned for different image resolutions and tasks,

⁵We empirically validated the max-pooling strategy over sub-page chunks to be more effective than concatenating all page chunks before embedding pagewise.

and the promising performances on various document understanding benchmarks. We add a projection layer to map the output language modeling embeddings to a vector space of reduced dimension $D = 128$ as used in the ColBERT paper (Khattab and Zaharia, 2020) to keep lightweight bag-of-embedding representations.

Late Interaction. Given query q and document d , we denote as $\mathbf{E}_q \in \mathbb{R}^{N_q \times D}$ and $\mathbf{E}_d \in \mathbb{R}^{N_d \times D}$ their respective multi-vector representation in the common embedding space \mathbb{R}^D . The late interaction operator, $LI(q, d)$, is the sum over all query vectors $\mathbf{E}_q^{(j)}$, of its maximum dot product $\langle \cdot | \cdot \rangle$ with each of the N_d document embedding vectors $\mathbf{E}_d^{(1:N_d)}$.

$$LI(q, d) = \sum_{i \in [1, N_q]} \max_{j \in [1, N_d]} \langle \mathbf{E}_q^{(i)} | \mathbf{E}_d^{(j)} \rangle \quad (1)$$

Contrastive Loss. The Late Interaction operation is fully differentiable, enabling backpropagation. Let a batch $\{q_k, d_k\}_{k \in [1, b]}$ composed of b query-page pairs, where for all $k \in [1, b]$, the document page d_k is the document corresponding to query q_k . Following Khattab and Zaharia (2020), we define our in-batch contrastive loss \mathcal{L} as the softmaxed cross-entropy of the positive scores $s_k^+ = LI(q_k, d_k)$ w.r.t. to the maximal negative scores $s_k^- = \max_{l, l \neq k} LI(q_k, d_l)$.

4.2 Model training

Dataset. Our training dataset of 127,460 query-page pairs is comprised of train sets of openly available academic datasets (63%) and a synthetic dataset made up of pages from web-crawled PDF documents and augmented with VLM-generated (Claude-3 Sonnet) pseudo-questions (37%). Our training set is fully English by design, enabling us to study zero-shot generalization to non-English languages⁶. We explicitly verify no multi-page PDF document is used both *ViDoRe* and in the train set to prevent evaluation contamination. A validation set is created with 2% of the samples to tune hyperparameters.

Parameters. All models are trained for 1 epoch on the train set. Unless specified otherwise, we train models in bfloat16 format, use low-rank adapters (LoRA, Hu et al. (2021)) with $\alpha = 32$ and $r = 32$ on the transformer layers from the language model,

⁶Multilingual data is present in the pretraining corpus of the language model (Gemma-2B) and potentially occurs during PaliGemma-3B’s multimodal training.

as well as the final randomly initialized projection layer, and use a paged_adamw_8bit optimizer. We train on an 8 GPU setup with data parallelism, a learning rate of $5e - 5$ with linear decay with 2.5% warmup steps, and a batch size of 32.

Query Augmentation. As in [Khattab and Zäharia \(2020\)](#), we append 5 <unused0> tokens to the query tokens to serve as a soft, differentiable query expansion or re-weighting mechanism.

5 Results

5.1 Performance (R1)

We iteratively construct *ColPali*, starting from an off-the-shelf SigLIP model ([Table 2](#)).

BiSigLIP: Improving a strong model. SigLIP⁷ is a strong vision-language bi-encoder model, pre-trained on the English split of WebLI ([Chen et al., 2023](#)), a corpus of billions of image-text pairs. We find that [SigLIP largely outperforms both Jina CLIP and Nomic-vision on document retrieval tasks](#). Further fine-tuning the textual component of this model on our document-oriented dataset (BiSigLIP) yields clear improvements across the board, particularly on figure retrieval (ArxivQA) and table retrieval tasks (TabFQuAD).

BiPali: Pairing with a language model. In the PaliGemma model architecture, SigLIP-generated patch embeddings are fed to a text language model to obtain LLM contextualized output patch embeddings.⁸ We average pool these representations to obtain a single dense vector, effectively creating a PaliGemma bi-encoder model (BiPali). After fine-tuning on the training dataset, we obtain a model that performs slightly worse in English than the tuned BiSigLIP variant. This can be explained by the fact that contrary to SigLIP, the original PaliGemma is not trained on contrastive matching tasks, but rather on next token prediction. Our contrastive fine-tuning phase on 100K images to transform PaliGemma into a bi-encoder is 5 orders of magnitude smaller than SigLIP’s original contrastive training. However, we see notable improvements in French tasks, indicating that BiPali’s LLM (Gemma 2B) helps multilingual text understanding. This is particularly notable as our training dataset does not contain non-English samples.

⁷<https://huggingface.co/google/siglip-so400m-patch14-384>

⁸Note that the SigLIP model used in PaliGemma slightly differs in terms of number patches - 1024 patches for PaliGemma’s vision encoder, and 729 for the standalone SigLIP model.

ColPali: Adding Late Interaction. One benefit of inputting image patch embeddings through a language model is that they are natively mapped to a latent space similar to textual input (query). This enables leveraging the ColBERT strategy to compute interactions between text tokens and image patches, which enables a step-change improvement in performance compared to BiPali. Results in [Table 2](#) show that our *ColPali* model also largely outperforms the strong baselines based on Unstructured and captioning, as well as all evaluated text-image embedding models. The difference is particularly stark on the more visually complex benchmark tasks, such as InfographicVQA, ArxivQA, and TabFQuAD representing respectively infographics, figures, and tables. However, text-centric documents are also better retrieved by the *ColPali* models across all evaluated domains and languages, making our approach the overall best-performing document-retrieval model.

Negative Results. For extensiveness, we also train ColSigLIP, a late interaction variant of the BiSigLIP model but obtain abysmal performances. We attribute this to the large gaps w.r.t. SigLIP’s pre-training, in which only a pooled latent representation is used in the contrastive loss, which does not optimize the representations of individual patch and token embeddings. Similarly, we train a BiSigLIP_{PaliGemma} variant, in which we retrieve the image representations from the SigLIP model that has been further updated by PaliGemma fine-tuning, and use the text representations from PaliGemma’s text model. After fine-tuning on our dataset, performance is severely inferior to SigLIP_{Vanilla} which simply encodes with SigLIP’s original text and vision components. This indicates a logical misalignment between SigLIP embeddings, and Gemma embeddings after PaliGemma training. We detail these results in [Table 5](#).

5.2 Latencies & Memory Footprint

Online Querying. (R2) Logically, querying latencies differ between *ColPali* and a BGE-M3 embedding model. For BGE, encoding takes about 22 ms for 15 tokens, while encoding a query with *ColPali*’s language model takes about 30 ms⁹. For smaller corpus sizes, computing the late interaction operation induces marginally small overheads (≈ 1 ms per 1000 pages in the corpus), and the cosine similarity computation between bi-encoder vectors

⁹Computed for a batch size of 1 (online), and averaged over 1000 queries. See [subsection B.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}

Table 2: **Comprehensive evaluation of baseline models and our proposed method on ViDoRe.** Results are presented using **NDCG@5** metrics, and illustrate the impact of different components. Text-only metrics are not computed for benchmarks with only visual elements.

is even faster. Optimized late interaction engines ([Santhanam et al., 2022](#); [Lee et al., 2023](#)) enable to easily scale corpus sizes to millions of documents with reduced latency degradations.

Offline Indexing. (R3) Standard retrieval methods using bi-encoders represent each chunk as a single vector embedding, which is easy to store and fast to compute. However, processing a PDF to get the different chunks is the most time-consuming part (layout detection, OCR, chunking), and using captioning to handle multimodal data will only exacerbate this already lengthy process. On the other hand, *ColPali* directly encodes pages from their image representation. Although the encoder model is larger than standard retrieval encoders, skipping the preprocessing allows large speedups at indexing¹⁰ ([Figure 3](#)).

Memory Footprint. Our method requires storing a vector per image patch. We project each PaliGemma vector to a lower dimensional space ($D=128$) to maximize efficiency, leading to a memory footprint of 256 KB per page ([subsection B.4](#)). Importantly, the memory footprint of the naive ColBERT indexing strategy can be drastically improved through compression and clustering mecha-

¹⁰Measures a NVIDIA L4 GPU, averaged on 100 pages, with a batch size of 4 pages for *ColPali* and 8 text chunks for Bi-Encoders. On average, a page is divided into 2.1 chunks. See [subsection B.5](#).

nisms as proposed in the Performance-optimized Late Interaction Driver ([Santhanam et al., 2022](#)).

5.3 Interpretability

By superimposing the late interaction heatmap on top of the original image, we can visualize the most salient image patches with respect to each term of the query, yielding interpretable insights into model focus zones. As epitomized in [Figure 1](#), we observe *ColPali* exhibits strong OCR capabilities as both the words “hourly” and “hours” present a high similarity score with the query token $\langle \text{hour} \rangle$. We also note particular focus on other non-trivial image features such as the x-axis representing hours being salient. Other visualization examples with similar trends of the model transcending pure OCR are shown in [Appendix C](#).

6 Ablation study

Should we scale models or patch numbers ?

We train a variant of PaliGemma with half the number of image patches (512). While there is a clear performance degradation w.r.t. to the 1024-patch *ColPali* model ([Figure 4](#)), memory usage is much lower.¹¹ As an alternative to PaliGemma, we train

¹¹While another PaliGemma variant exists with 2048 patches, the different training datamix and the large memory requirements make this model impractical for both training



Figure 4: Relative NDCG@5 performance gain w.r.t. the default *ColPali* (1024 patches). *TabFQuAD finetuning* measures the performance difference on the TabFQuAD task after the introduction of targeted data in the training set. All other results refer to performance deltas averaged on all *ViDoRe* tasks.

Idefics2-8B (Laurençon et al., 2024), a VLM with a similar architecture and based on a Mistral-7B (Jiang et al., 2023) language backbone and a SigLIP vision encoder paired with a perceiver resampler. The most notable differences with PaliGemma lie in the size of the language model (2B and 7B resp.) and the number of image patches (between 512 and 2048 for PaliGemma, and 64 post-resampling for Idefics2¹²). Our results (Figure 4) suggest language model size has a strong impact on performance, and along with the trained resampler enables more efficient representations for smaller numbers of image embeddings - ColIdefics2 with 64 patches edges out *ColPali* with 512 patches. Scaling the number of patches of the smaller *ColPali* model from 512 to 1024, enables largely surpassing the 60-patch ColIdefics2 while being about twice as fast in terms of training and inference latency. These results suggest there are *tradeoffs* between performance (R1), latencies during online querying (R2) and offline indexation phases (R3), and index memory size.

Should we fine-tune the vision component?

We run our contrastive finetuning on a *ColPali* model in which we also train the vision encoder and the projection layer. Results in Figure 4 show this leads to no significant improvements.

Do "query augmentation" tokens help?

In ColBERT, special tokens are concatenated to the input query to serve as soft query augmentation buffers. Training without these tokens, we observe no significant performance difference (Figure 4) in the English benchmarks. However, performance on the French tasks seems to improve (Table 5).

and inference time.

¹²With the option of adding 4 sub-image crops of 64 tokens each to the sequence, for a total of 320 tokens

Is the Pairwise CE loss best?

Training with an in-batch negative contrastive loss, instead of the pairwise CE loss that only considers the hardest negative sample, leads to a slight performance degradation (-2.4%) on the aggregated benchmark.

Can the model adapt to new tasks?

Contrary to more complex multi-step retrieval pipelines, *ColPali* can be trained end-to-end, directly optimizing the downstream retrieval task which greatly facilitates fine-tuning to boost performance on specialized domains, multilingual retrieval, or specific visual elements the model struggles with. To demonstrate, we add 1552 samples representing French tables and associated queries to the training set. This represents the only French data in the training set, with all other examples being kept unchanged. We see significant NDCG@5 improvements (Figure 4) and even starker Recall@1 gains (+6.63%) on the TabFQuAD benchmark, with no performance degradation on the rest of the benchmark tasks (+0.34%).

7 Conclusions

Through the conception of a new benchmark *ViDoRe*, we established the limits of both modern industrial document retrieval pipelines and off-the-shelf image-text contrastive models for visually rich document retrieval. We introduced *ColPali*, a novel retrieval model that leverages the latest generative Vision Language models to create highly performing multi-vector embeddings purely from visual document features. *ColPali* largely outperforms the best existing document retrieval methods while enabling faster corpus indexing time and maintaining low querying latencies, suggesting a very high potential for industrial document retrieval applications. We hope to encourage future work by publicly releasing the *ViDoRe* benchmark and all models and baselines from our study.

Future Work. Further performance gains could be obtained by exploring sub-image decomposition (Liu et al., 2023a), optimal image patch resampling strategies (Laurençon et al., 2024), or hard-negative mining. Subsequently, our vision is to combine visual retrieval and visually grounded query answering to create RAG systems that purely function from visual features. An interesting line of research could be attempting to generate answers leveraging information stored in the indexed multi-vector patch embeddings.

Limitations

Focus. In this work, we evaluate models on document retrieval tasks, covering several modalities (figures, text, tables, infographics). We however primarily focus on PDF-type documents, and evaluating systems on image retrieval with documents stemming from web page screenshots or handwritten documents might be an interesting generalization. We also focus on high-resource languages (English and French) and although we have shown the capacity of the *ColPali* model to generalize to languages outside of its fine-tuning set, it is unclear how the model would perform on languages that are not as represented in the model’s language backbone. Finally, our setup assumes relevant documents exist, but abstention methods for Information Retrieval systems might be interesting to explore in more practical settings in which confidence estimation might be important (Gisserot-Boukhlef et al., 2024).

Support. This work relies on multi-vector retrieving derived from the ColBERT late interaction mechanism. Although some vector databases support late interaction engines¹³, many widely used vector retrieval frameworks do not propose native multi-vector support, and some engineering infrastructure efforts may be required to adapt them to work with *ColPali* (or ColBERT) models.

Data. In the creation of *ViDoRe*, we partially rely on synthetic query generation based on a commercial large language model, which may induce some amount of bias in the generated queries. To compensate for this, we have iterated on the prompting strategy and given real query examples to the models to help ground generation in realistic settings. We have further manually verified all synthetic queries through a lengthy process to validate their relevance and their quality. Our benchmark also includes many benchmark tasks with no synthetic data, and result trends observed between all tasks are correlated, further confirming the coherence of our benchmark design.

Ethical Considerations

Carbon Footprint. Our work fully leverages prior pretrained models and training is not particularly compute-intensive. Furthermore, we rely on low-rank adapters to further reduce the computational resources needed, both during training and for

storage. Overall, a training run represents about 40 hours of Mi250x AMD GPUs. Our experiments, in total, represent 1405 Mi250x GPU hours from highly efficient compute clusters running on low-carbon nuclear energy, representing a total of around 15kg CO₂ eq.

Impact. We believe our work could have a strong impact on improving industrial document retrieval systems. Our method is efficient, performs well, and the additional support towards visually rich information from documents could go a long way in unlocking knowledge sources previously difficult to index or query.

Resource Release. For transparency, and to foster future work, we release our comprehensive benchmark under open license and host a public leaderboard¹⁴. Our models are released under the same usage license as the base model (Gemma Research license for ColPali, Apache2.0 for ColIdefics2) and should be used as intended by the VLM license.

Acknowledgements

This work is partially supported by Illuin Technology, and by a grant from ANRT France. This work was performed using HPC resources from the CINES ADASTRA through Grant 2024-AD011015443. We extend our warm thanks to Jonathan Dong, Caio Corro, Victor Pellegrain and Ender Konukoglu for their valuable feedback on the paper.

References

- Ibrahim Alabdulmohsin, Xiaohua Zhai, Alexander Kolesnikov, and Lucas Beyer. 2023. [Getting ViT in Shape: Scaling Laws for Compute-Optimal Model Design](#). Publisher: arXiv Version Number: 5.
- Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. 2022. [Flamingo: a Visual Language Model for Few-Shot Learning](#). Publisher: arXiv Version Number: 2.
- Anthropic. 2024. [The Claude 3 Model Family: Opus, Sonnet, Haiku](#).

¹⁴<https://huggingface.co/spaces/vidore/vidore-leaderboard>

¹³Vespa Engine, RAGatouille, QDrant, colbert.ai

- Srikanth Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R. Manmatha. 2021. **DocFormer: End-to-End Transformer for Document Understanding**. *arXiv preprint*. Version Number: 2.
- Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. **Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond**. Publisher: arXiv Version Number: 3.
- Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2016. **MS MARCO: A Human Generated Machine Reading Comprehension Dataset**. *arXiv preprint*. Version Number: 3.
- Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazovic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. 2024. **Paligemma: A versatile 3b vlm for transfer**. *Preprint*, arXiv:2407.07726.
- Burton H. Bloom. 1970. **Space/time trade-offs in hash coding with allowable errors**. *Commun. ACM*, 13(7):422–426. Place: New York, NY, USA Publisher: Association for Computing Machinery.
- Łukasz Borchmann, Michał Pietruszka, Tomasz Stanisławek, Dawid Jurkiewicz, Michał Turski, Karolina Szyndler, and Filip Graliński. 2021. **DUE: End-to-End Document Understanding Benchmark**. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.
- Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. **BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation**. *arXiv preprint*. Version Number: 3.
- Xi Chen, Xiao Wang, Lucas Beyer, Alexander Kolesnikov, Jialin Wu, Paul Voigtlaender, Basil Mustafa, Sebastian Goodman, Ibrahim Alabdulmohsin, Piotr Padlewski, Daniel Salz, Xi Xiong, Daniel Vlasic, Filip Pavetic, Keran Rong, Tianli Yu, Daniel Keysers, Xiaohua Zhai, and Radu Soricut. 2023. **PaLI-3 Vision Language Models: Smaller, Faster, Stronger**. *arXiv preprint*. Version Number: 2.
- Cohere. 2024. **Introducing Rerank 3: A New Foundation Model for Efficient Enterprise Search & Retrieval**.
- Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. 2023. **Vision Transformers Need Registers**. Publisher: [object Object] Version Number: 2.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2020. **An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale**. Publisher: arXiv Version Number: 2.
- Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. 2021. **YOLOX: Exceeding YOLO Series in 2021**. *arXiv preprint*. Version Number: 2.
- Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Huszenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Sloane, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Milligan, Lars Lowe Sjøesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yотов, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu-hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. **Gemma: Open Models Based on Gemini Research and Technology**. *arXiv preprint*. Version Number: 4.
- Hippolyte Gisserot-Boukhlef, Manuel Faysse, Emmanuel Malherbe, Céline Hudelot, and Pierre Colombo. 2024. **Towards trustworthy reranking: A simple yet effective abstention mechanism**. *Preprint*, arXiv:2402.12997.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. **LoRA: Low-Rank Adaptation**

- of Large Language Models. Publisher: arXiv Version Number: 2.
- Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. **LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking**. Publisher: arXiv Version Number: 3.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lampe, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. **Mistral 7B**. Publisher: arXiv Version Number: 1.
- Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. **Dense Passage Retrieval for Open-Domain Question Answering**. *arXiv preprint*. Version Number: 3.
- Omar Khattab and Matei Zaharia. 2020. **ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT**.
- Geewook Kim, Teakgyu Hong, Moonbin Yim, Jeongyeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2021. **OCR-free Document Understanding Transformer**. *arXiv preprint*. Version Number: 5.
- Andreas Koukounas, Georgios Mastrapas, Michael Günther, Bo Wang, Scott Martens, Isabelle Mohr, Saba Sturua, Mohammad Kalim Akram, Joan Fontanals Martínez, Saahil Ognawala, Susana Guzman, Maximilian Werk, Nan Wang, and Han Xiao. 2024. **Jina CLIP: Your CLIP Model Is Also Your Text Retriever**. *arXiv preprint*. Version Number: 1.
- Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. 2024. **What matters when building vision-language models?** *arXiv preprint*. ArXiv:2405.02246 [cs].
- Jinhyuk Lee, Zhuyun Dai, Sai Meher Karthik Duddu, Tao Lei, Iftekhar Naim, Ming-Wei Chang, and Vincent Y. Zhao. 2023. **Rethinking the Role of Token Retrieval in Multi-Vector Retrieval**. *arXiv preprint*. Version Number: 3.
- Lei Li, Yuqi Wang, Runxin Xu, Peiyi Wang, Xiachong Feng, Lingpeng Kong, and Qi Liu. 2024. **Multimodal arxiv: A dataset for improving scientific comprehension of large vision-language models**. *Preprint*, arXiv:2403.00231.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2014. **Microsoft COCO: Common Objects in Context**. *arXiv preprint*. Version Number: 3.
- Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023a. **Improved Baselines with Visual Instruction Tuning**. *arXiv preprint*. Version Number: 2.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. **Visual Instruction Tuning**. Publisher: arXiv Version Number: 1.
- Minesh Mathew, Viraj Bagal, Rubén Pérez Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V Jawahar. 2021. **InfographicVQA**. *arXiv preprint*. Version Number: 2.
- Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. 2020. **DocVQA: A Dataset for VQA on Document Images**.
- Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. **MTEB: Massive Text Embedding Benchmark**. *arXiv preprint*. Version Number: 3.
- Nomic. 2024. **Nomic Embed Vision: Expanding The Nomic Latent Space**.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. **Learning Transferable Visual Models From Natural Language Supervision**. Publisher: arXiv Version Number: 1.
- Nils Reimers and Iryna Gurevych. 2019. **Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks**. *arXiv preprint*. Version Number: 1.
- Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. **Okapi at TREC-3**. In *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994*, volume 500-225 of *NIST Special Publication*, pages 109–126. National Institute of Standards and Technology (NIST).
- Keshav Santhanam, Omar Khattab, Christopher Potts, and Matei Zaharia. 2022. **PLAID: An Efficient Engine for Late Interaction Retrieval**. *arXiv preprint*. Version Number: 1.
- R. Smith. 2007. **An Overview of the Tesseract OCR Engine**. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 2*, pages 629–633, Curitiba, Parana, Brazil. IEEE. ISSN: 1520-5363.
- Karen Sparck Jones. 1972. **A STATISTICAL INTERPRETATION OF TERM SPECIFICITY AND ITS APPLICATION IN RETRIEVAL**. *Journal of Documentation*, 28(1):11–21.
- Zineng Tang, Ziyi Yang, Guoxin Wang, Yuwei Fang, Yang Liu, Chenguang Zhu, Michael Zeng, Cha Zhang, and Mohit Bansal. 2022. **Unifying Vision, Text, and Layout for Universal Document Processing**. *arXiv preprint*. Version Number: 3.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. **BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models.** *arXiv preprint*. Version Number: 4.

Ashish V. Thapliyal, Jordi Pont-Tuset, Xi Chen, and Radu Soricut. 2022. **Crossmodal-3600: A Massively Multilingual Multimodal Evaluation Dataset.** *arXiv preprint*. Version Number: 2.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. **Text Embeddings by Weakly-Supervised Contrastive Pre-training.** *arXiv preprint*. Version Number: 2.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. **MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers.** *arXiv preprint*. ArXiv:2002.10957 [cs].

Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. 2021. **FILIP: Fine-grained Interactive Language-Image Pre-Training.** *arXiv preprint*. Version Number: 1.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhua Chen. 2023. **MMMU: A Massive Multi-discipline Multimodal Understanding and Reasoning Benchmark for Expert AGI.** *arXiv preprint*. Version Number: 3.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. **Sigmoid Loss for Language Image Pre-Training.** Publisher: [object Object] Version Number: 4.

Ruochen Zhao, Hailin Chen, Weishi Wang, Fangkai Jiao, Xuan Long Do, Chengwei Qin, Bosheng Ding, Xiaobao Guo, Minzhi Li, Xingxuan Li, and Shafiq Joty. 2023. **Retrieving Multimodal Information for Augmented Generation: A Survey.** *arXiv preprint*. Version Number: 3.

Fengbin Zhu, Wenqiang Lei, Fuli Feng, Chao Wang, Haozhou Zhang, and Tat-Seng Chua. 2022. **Towards Complex Document Understanding By Discrete Reasoning.** Publisher: arXiv Version Number: 3.

A Benchmark Datasets

A.1 Academic Datasets

DocVQA (Mathew et al., 2020) includes collected images from the UCSF Industry Documents Library. Questions and answers were manually annotated.

InfoVQA (Mathew et al., 2021) includes infographics collected from the Internet using the search query “infographics”. Questions and answers were manually annotated.

TAT-DQA (Zhu et al., 2022) is a large-scale Document VQA dataset that was constructed from publicly available real-world financial reports. It focuses on rich tabular and textual content requiring numerical reasoning. Questions and answers were manually annotated by human experts in finance. **arXivQA** (Li et al., 2024) is a VQA dataset based on figures extracted from arXiv publications. The questions were generated synthetically using GPT-4 Vision.

TabFQuAD (Table French Question Answering Dataset) is designed to evaluate TableQA models in realistic industry settings. We create additional queries to augment the existing human-annotated ones using the same method described in subsection A.2.

A.2 Practical Datasets

Methodology. Creating a relevant retrieval dataset close to real use cases is a major challenge as the dataset needs to be both sufficiently large for effective fine-tuning and sufficiently diverse to cover a broad range of modalities (full text, tables, charts, ...), domains (industry, healthcare, ...), and query-document interactions (extractive questions, open-ended questions, ...). Our approach to building this dataset involves several steps: (1) we use a web crawler to collect publicly available documents on various themes and sources, (2) we convert these PDFs into a series of images, one per page, and (3) we generate queries related to each image using a VLM.

Web-Crawler. We implemented a web crawler to efficiently collect large volumes of documents related to a given topic. The crawler is seeded with a user-defined query (e.g. "artificial intelligence") and then uses GPT-3.5 Turbo to brainstorm related topics and subtopics. This query augmentation strategy aims at both broadening and deepening the search. GPT-3.5 Turbo is further used to generate diverse search queries from each subtopic. This query set is then consumed by a pool of parallel workers whose job is to fetch the associated most relevant documents. We use SerpAPI¹⁵ along with a filetype filter (PDF documents only) to programmatically scrape Google Search rankings. Each

¹⁵<https://serpapi.com/>

file is hashed and stored in a Bloom filter ([Bloom, 1970](#)) shared among workers to avoid duplicate documents in the final corpus. Unique scraped files are downloaded, and inserted into a SQLite database along with additional metadata.

Datamix. Using the web crawler, we collected approximately 1,000 documents for each of the following four seeds: "*energy*", "*government reports*", "*healthcare industry*", and "*artificial intelligence*". These seeds were meticulously hand-picked to align with real-use cases for retrieval models and visually rich pages. We also removed all documents containing any private information. At this stage, we randomly selected 900 files for the training set and 100 files for the test set, ensuring that data leakage into the test set was avoided during subsequent processing steps.

Query Generation. To increase the efficiency of our query generation scheme and to limit API calls, we generate at most 3 questions per image. From all the documents collected, we randomly sample 10,000 images per theme and call Claude-3 Sonnet with the following prompt:

You are an assistant specialized in
 ↳ Multimodal RAG tasks.\n
 The task is the following: given an image
 ↳ from a pdf page, you will have to
 generate questions that can be asked by a
 ↳ user to retrieve information from
 a large documentary corpus.
 The question should be relevant to the
 ↳ page, and should not be too specific
 or too general. The question should be
 ↳ about the subject of the page, and
 the answer needs to be found in the page.
 ↳ \n

Remember that the question is asked by a
 ↳ user to get some information from a
 large documentary corpus that contains
 ↳ multimodal data. Generate a question
 that could be asked by a user without
 ↳ knowing the existence and the content
 of the corpus. \n
 Generate as well the answer to the
 ↳ question, which should be found in the
 page. And the format of the answer should
 ↳ be a list of words answering the
 question. \n

Generate at most THREE pairs of questions
 ↳ and answers per page in a
 dictionary with the following format,
 ↳ answer ONLY this dictionary
 NOTHING ELSE: \n

```
{
  "questions": [
    {
      "question": "XXXXXX",
      "answer": ["YYYYYY"]
    },
    {
      "question": "XXXXXX",
      "answer": ["YYYYYY"]
    },
    {
      "question": "XXXXXX",
      "answer": ["YYYYYY"]
    }
  ]
}
```

where XXXXX is the question and
 ↳ ['YYYYYY'] is the corresponding list
 ↳ of answers
 that could be as long as needed. \n

Note: If there are no questions to ask
 ↳ about the page, return an empty list.
 Focus on making relevant questions
 ↳ concerning the page. \n
 Here is the page: \n

Human Validation. We manually validate every single synthetically created query in *ViDoRe* to ensure quality, query relevance, and consistency with the benchmark objective of evaluating retrieval in practical industrial settings. During this step, we randomly assign document-pair queries to 4 vol-

unteer annotators and instruct them to filter out queries that do not fit the above-listed criteria. We also instruct annotators to flag any documents they deem to contain PII information or content not suited for an academic benchmark. No flag was raised during the entirety of the process, validating our prior PDF collection strategy. 100 queries per topic are collected in this manner. Annotators are colleagues and collaborators of the authors who volunteered to help. Each annotator spent approximately 3 hours filtering the larger query set down to 100 high-quality queries per topic.

B Implementation details

B.1 Codebase

The codebase is written in PyTorch¹⁶ and leverages HuggingFace tooling for model implementations and trainers¹⁷.

B.2 Pairwise CE loss

Our in-batch contrastive loss \mathcal{L} is defined as the softmaxed cross-entropy of the positive scores $s_k^+ = \text{LI}(d_k, q_k)$ w.r.t. to the maximal negative scores $s_k^- = \max_{l,l \neq k} \text{LI}(q_k, p_l)$.

For numerical stability, we reformulate the loss with the softplus function, leading to:

$$\mathcal{L} = \frac{1}{b} \sum_{k=1}^b \text{softplus}(s_k^- - s_k^+) \quad (2)$$

B.3 Hyperparameters

Hyperparameters are tuned on a validation split composed of 2% of the training dataset. We find bi-encoder methods to be more sensible to learning rate variations than late interaction-based models and achieve the best performance for all models with a learning rate of $5e-5$. We experiment with LoRA rank and α values and do not notice particular improvements past $r = \alpha = 32$. Per-device batch sizes are kept small due to long sequence lengths that complicate scaling past $b = 4$. Simulating larger batch sizes for in-batch negative sampling should enable even better results. We find the best results with global batch size $b = 32$ for 1 epoch on our training set.

B.4 Embedding size

Minimizing storage footprint can be essential to industrial retrieval systems if databases contain mil-

lions of documents. With this criterion in view, we have compared the embedding sizes of the models in our study. As shown in Table 3, *ColPali*'s embedding size is an order of magnitude larger than BM25 and two orders of magnitude larger than BGE-M3. However, this study is limited to the naive method of storing *ColPali*'s multi-vector embeddings. In practical scenarios, using cluster centroids can reduce the size of *ColPali* multi-vector embeddings by up to an order of magnitude (Santhanam et al., 2022) and make it a competitive retrieval system.

Model	Embedding size (KB)
BGE-M3	8.60
BM25 (dense emb.)	3.00
BM25 (sparse emb.)	1.56 ± 0.51
<i>ColPali</i> (float16)	256

Table 3: Comparison of the embedding sizes for the DocVQA test set from *ViDoRe* w.r.t. different retrieval models. The lower the size the smaller the storage footprint of the model. The $mean \pm std$ size is given for the sparse embeddings.

B.5 Latency computations

All latency computations are done on a NVIDIA L4 GPU. Queries are encoded independently (batch size of 1) to simulate online querying, and pages are encoded with a batch size of 4 for PaliGemma derived models, and 8 for BGE-M3. Reported times include image and text processing time before the model forward pass, as well as query-to-index matching times. We note an interesting feature of *ColPali* is that all documents have the same sequence length, leading to prior knowledge of runtime and memory consumptions. Query latency experiments are averaged over 1000 queries, and indexing times are measured for a 100 page document. Per page time is obtained by diving total time by 100, corresponding to inverse page throughput.

B.6 Captioning

Examples of captions generated for visually rich document chunks with Claude-3 Sonnet are shown in Figure 6 and Figure 5. The prompt used for generating the description is the following:

¹⁶<https://pytorch.org/>

¹⁷<https://huggingface.co>

You are an assistant specialized in document analysis. Given a table or a figure, you have to provide a detailed summary of the content in maximum 3000 characters. Your summary should be qualitative and not quantitative. Here is the table/figure to analyze: {image}. Answer ONLY with the caption of the table/figure.

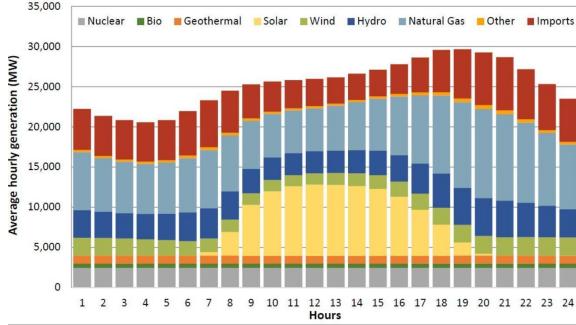


Figure 5: Example from the "Energy" test set.

Caption: The image depicts the hourly energy generation profile, illustrating the contributions of various energy sources over 24 hours. The data is presented as a stacked bar chart, with the x-axis representing the hours of the day from 1 to 24, and the y-axis showing the average hourly generation in MW. The bars are segmented into different colors, each representing a distinct energy source: nuclear, bio, geothermal, solar, wind, hydro, natural gas, and other imports. The chart provides insights into the temporal variations in energy generation across different sources, highlighting the interplay between baseload and intermittent sources throughout the day.

Document	System of Record ¹				
	County Servers	Project View	OnBase	CGI Advantage Financial System	Purchasing Department
Project plan/scope	W	T			
Project budgets	W	T	W	T	
Schedules/timelines (design)	W	T	W	T	
Schedules/timelines (construction)	W	T	W		
Executed contracts	T			W	T
Purchase orders		T		W	
Supplier advertisements		T		W	T
Proposal bid submissions		T		W	
Selection committee evaluation documents		T		W	T
Independent cost estimates		T	W	T	W
Negotiation details		T	W		W
Notice of award		T	W		W
Notice to proceed		T	W		W
Invoices/inifications for payment ("AFP")		T	W	T	W
Change order requests		T	W		
Approved change orders		T	W		
Requests for Information (RFI)	W	T			
Bid submissions			W		
Shop drawings	W	T			
Photographs	W	T			
As-builds	W	T			
Operating manuals	W	T			
Maintenance schedules	W	T			
Warrenings	W	T			
Daily diaries	N/A	T			
Weekly meeting minutes	W	T			
Monthly progress reports	W	T			
Allowance usage logs	W	T			
Punch lists	W	T			

Figure 6: Example from the "Government Reports" test set.

Caption: The image shows a table titled "System of Record" which outlines the different types of documents or records maintained across various systems or departments within an organization related to project management and construction. The rows list documents like project plans, budgets, schedules, contracts, purchase orders, invoices, change requests, bid submissions, drawings, manuals, meeting minutes, and reports. The columns indicate the system or department responsible for maintaining each record, such as County Servers, Project View, OnBase, CGI Advantage Financial System, and Purchasing Department. The table uses "W" and "T" markers to denote which system or department serves as the primary source (writer) or storage location (trailer) for each type of document.

C More similarity maps

In Figure 7, ColPali assigns a high similarity to all patches with the word "Kazakhstan" when given the token <_Kazakhstan>. Moreover, our model seems to exhibit world knowledge capabilities as the patch around the word "Kashagan" - an off-shore oil field in Kazakhstan - also shows a high similarity score. On the other hand, in Figure 8, we observe that ColPali is also capable of complex image understanding. Not only are the patches containing the word "formulations" highly similar to the query token _formula, but so is the upper-left molecule structure.

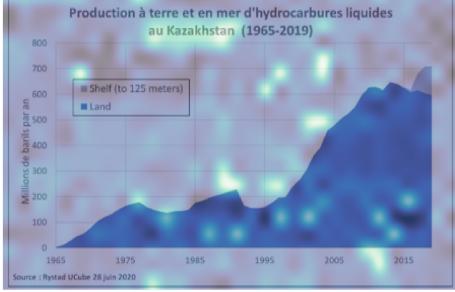
It is also interesting to highlight that both similarity maps showcase a few white patches with high similarity scores. This behavior might first seem surprising as the white patches should not carry a meaningful signal from the original images. We believe the vectors associated with these patches share a similar role with the ViT registers (Dariset al., 2023), i.e. these patches were repurposed for internal computations and stored the global information from the whole image.

La taille moyenne des champs pétroliers découverts au Kazakhstan est en déclin depuis les années 1970.

La taille moyenne a augmenté légèrement au cours des années 2000 de concert avec le nouveau cycle de découvertes lié aux champs offshore. Cependant, ces chiffres ont repris rapidement leur déclin rejoignant la tendance générale. Depuis 2007, seules deux années présentent une valeur supérieure à 25 millions de barils.

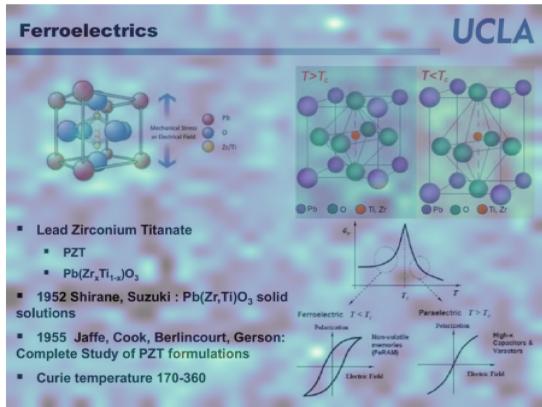
II. Historique de production

La production d'hydrocarbures liquides du Kazakhstan est en forte hausse depuis le début des années 1990. En 2019, elle représente 710 millions de barils (1,1 Mbd). Cette production se trouvait depuis 1977 sur un plateau d'environ 150 millions de barils par an. La production d'hydrocarbures du Kazakhstan reste principalement située à terre (84 % en 2019). Une partie de cette production est issue depuis 2016 de champs se situant en eaux peu profondes (*shelf*), grâce au lancement de la production du champ géant de Kashagan.



Query: "Quelle partie de la production pétrolière du Kazakhstan provient de champs en mer ?"

Figure 7: Similarity of the image patches w.r.t. the underlined token in the user query. This example is from the *Shift* test set.



Query: What is the chemical formula for the ferroelectric material Lead Zirconium Titanate (PZT)?

Figure 8: Similarity of the image patches w.r.t. the underlined token in the user query. This example is from the *Healthcare Industry* test set.

D Additional results

D.1 Other Metrics

	ArxivQ	DocQ	InfoQ	TabF	TATQ	Shift	AI	Energy	Gov.	Health.	Avg.
Unstructured Text only											
BM25	-	26.6	-	-	34.6	45.0	86.0	70.0	68.0	74.0	-
BGE-M3	-	22.8 _{↓3.8}	-	-	26.1 _{↓8.5}	51.0 _{↑6.0}	81.0 _{↓5.0}	72.0 _{↑2.0}	67.0 _{↓1.0}	77.0 _{↑3.0}	-
Unstructured + OCR											
BM25	26.7	28.9	54.0	30.4	50.0	52.0	86.0	77.0	74.0	80.0	55.9
BGE-M3	28.1 _{↑1.4}	22.9 _{↓6.0}	53.8 _{↓0.2}	55.7 _{↑25.3}	38.6 _{↓11.4}	56.0 _{↑4.0}	82.0 _{↓4.0}	79.0 _{↑2.0}	76.0 _{↑2.0}	83.0 _{↑3.0}	57.5 _{↑1.6}
Unstructured + Captioning											
BM25	35.5	30.2	61.5	24.3	49.0	47.0	79.0	76.0	75.0	81.0	55.9
BGE-M3	29.3 _{↓6.2}	26.0 _{↓4.2}	62.1 _{↑0.6}	58.6 _{↑34.3}	30.6 _{↓18.4}	55.0 _{↑8.0}	80.0 _{↑1.0}	78.0 _{↑2.0}	69.0 _{↓6.0}	83.0 _{↑2.0}	57.2 _{↑1.3}
Contrastive VLMs											
Jina-CLIP	19.4	7.3	26.7	12.5	1.6	2.0	11.0	13.0	15.0	17.0	12.6
Nomic-vision	10.4	6.7	22.1	9.6	1.6	0.0	9.0	9.0	7.0	13.0	8.8
SigLIP (Vanilla)	34.2	21.3	51.8	46.1	17.9	13.0	50.0	51.0	47.0	65.0	39.7
Ours											
(Copied) SigLIP (Vanilla)	34.2	21.3	51.8	46.1	17.9	13.0	50.0	51.0	47.0	65.0	39.7
BiSigLIP (+fine-tuning)	49.2 _{↑15.0}	23.8 _{↑2.5}	59.0 _{↑7.2}	52.1 _{↑6.0}	20.7 _{↑2.8}	16.0 _{↑3.0}	62.0 _{↑12.0}	61.0 _{↑10.0}	55.0 _{↑8.0}	72.0 _{↑7.0}	47.1 _{↑7.4}
BiPali (+LLM)	46.4 _{↓2.8}	20.0 _{↓3.8}	54.6 _{↓4.4}	63.2 _{↑11.1}	20.4 _{↓0.4}	34.0 _{↑18.0}	59.0 _{↓3.0}	45.0 _{↓16.0}	57.0 _{↑2.0}	56.0 _{↓16.0}	45.6 _{↓1.5}
ColPali (+Late Inter.)	72.4 _{↑26.0}	45.6 _{↑25.6}	74.6 _{↑20.0}	75.4 _{↑12.1}	53.1 _{↑32.7}	55.0 _{↑21.0}	93.0 _{↑34.0}	85.0 _{↑40.0}	85.0 _{↑28.0}	88.0 _{↑32.0}	72.7 _{↑27.1}

Table 4: **Comprehensive evaluation of baseline models and our proposed method on ViDoRe.** Results are presented using Recall@1 metrics. Text-only metrics are not computed for benchmarks with only visual elements.

D.2 Model Variants

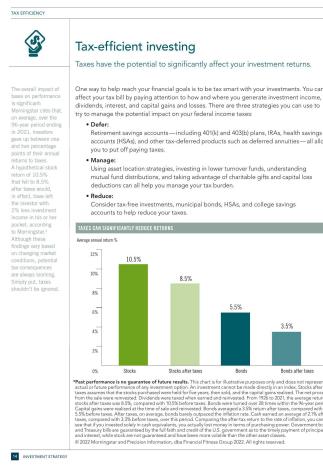
	ArxivQ	DocQ	InfoQ	TabF	TATQ	Shift	AI	Energy	Gov.	Health.	Avg.
ColSigLIP (PaliGemma)	3.1	3.0	5.1	6.2	2.5	1.0	3.4	3.4	2.3	2.2	3.2
BiSigLIP (PaliGemma)	18.5	14.6	33.4	39.5	16.1	5.2	27.6	32.6	36.6	35.7	26.0
ColSigLIP (Original)	2.6	2.2	2.3	5.7	1.8	1.0	2.6	4.1	1.4	1.5	2.5
ColPali (No Mem. Tokens)	80.4	53.2	82.4	77.4	65.7	63.4	97.0	89.9	93.6	92.4	79.6
ColPali (Best)	79.1	54.4	81.8	83.9	65.8	73.2	96.2	91.0	92.7	94.4	81.3

Table 5: **Evaluation of some "negative results" and ablations on ViDoRe; ColPali for reference.** Results are presented using NDCG@5 metrics. Text-only metrics are not computed for benchmarks with only visual elements.

E *ViDoRe* examples

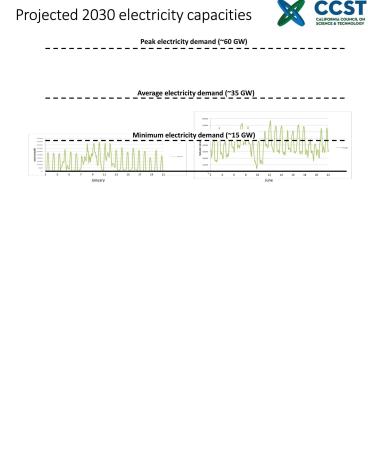
Energy

Query: What types of accounts or products allow investors to defer paying taxes?



- **Query:** What is the estimated total savings for a PV system in Durham under the net metering (flat rate) billing option over the system's useful life of 25 years?

- **Query:** What is the projected peak electricity demand in California for the year 2030?



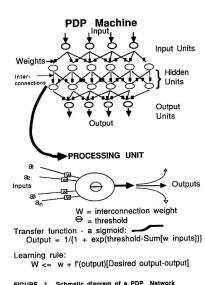
Artificial Intelligence

Query: What are some common outcome areas targeted by TAII for different age groups?



- **Query:** What did the robot monitor
- determine when to activate or deactivate the blower motor and blinker?

Query: What is the key approach used in the PDP architecture?



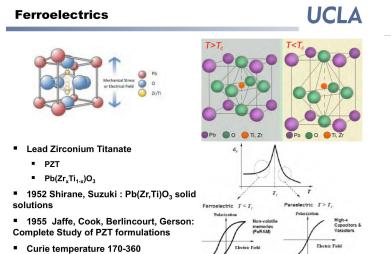
approach in the PDP architecture is the use of an adaptive filter. The adaptive filter takes a real-world analog input and compare it to a stored pattern. The stored pattern represents the desired input level which can then be modified as necessary so that signals can be produced the correct results. A more abstract argument driving the design of the PDP network is machine efficiency, that is, the optimal utilization of the computer circuitry. In the more familiar structure of a modern computer (the von Neumann architecture) most of the computer's circuitry in the memory is not in active use at this time. This is very wasteful from a pure resource management standpoint. The

PDP machine addresses this problem by using many processors and memories, but does so

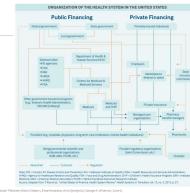
18

Healthcare Industry

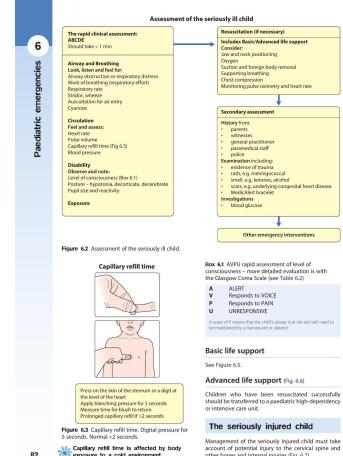
Query: What is the chemical formula for the ferroelectric material Lead Zirconium Titanate (PZT)?



Query: What government entities are involved in public financing for healthcare in the US?



Query: What does the AVPU scale stand for in assessing the level of consciousness of a seriously ill child?



82

Figure 6.2 Assessment of the seriously ill child.

Figure 6.3 Capillary refill time. Digital pressure for 5 seconds. Normal <2 seconds.

Figure 6.4 Basic life support.

Figure 6.5 Advanced life support (Fig. 6.6).

Figure 6.6 The seriously injured child.

Figure 6.7 Other emergency interventions.

Figure 6.8 Basic life support.

Figure 6.9 Advanced life support (Fig. 6.6).

Figure 6.10 The seriously injured child.

Figure 6.11 Other emergency interventions.

Figure 6.12 Basic life support.

Figure 6.13 Advanced life support (Fig. 6.6).

Figure 6.14 The seriously injured child.

Figure 6.15 Other emergency interventions.

Figure 6.16 Basic life support.

Figure 6.17 Advanced life support (Fig. 6.6).

Figure 6.18 The seriously injured child.

Figure 6.19 Other emergency interventions.

Figure 6.20 Basic life support.

Figure 6.21 Advanced life support (Fig. 6.6).

Figure 6.22 The seriously injured child.

Figure 6.23 Other emergency interventions.

Figure 6.24 Basic life support.

Figure 6.25 Advanced life support (Fig. 6.6).

Figure 6.26 The seriously injured child.

Figure 6.27 Other emergency interventions.

Figure 6.28 Basic life support.

Figure 6.29 Advanced life support (Fig. 6.6).

Figure 6.30 The seriously injured child.

Figure 6.31 Other emergency interventions.

Figure 6.32 Basic life support.

Figure 6.33 Advanced life support (Fig. 6.6).

Figure 6.34 The seriously injured child.

Figure 6.35 Other emergency interventions.

Figure 6.36 Basic life support.

Figure 6.37 Advanced life support (Fig. 6.6).

Figure 6.38 The seriously injured child.

Figure 6.39 Other emergency interventions.

Figure 6.40 Basic life support.

Figure 6.41 Advanced life support (Fig. 6.6).

Figure 6.42 The seriously injured child.

Figure 6.43 Other emergency interventions.

Figure 6.44 Basic life support.

Figure 6.45 Advanced life support (Fig. 6.6).

Figure 6.46 The seriously injured child.

Figure 6.47 Other emergency interventions.

Figure 6.48 Basic life support.

Figure 6.49 Advanced life support (Fig. 6.6).

Figure 6.50 The seriously injured child.

Figure 6.51 Other emergency interventions.

Figure 6.52 Basic life support.

Figure 6.53 Advanced life support (Fig. 6.6).

Figure 6.54 The seriously injured child.

Figure 6.55 Other emergency interventions.

Figure 6.56 Basic life support.

Figure 6.57 Advanced life support (Fig. 6.6).

Figure 6.58 The seriously injured child.

Figure 6.59 Other emergency interventions.

Figure 6.60 Basic life support.

Figure 6.61 Advanced life support (Fig. 6.6).

Figure 6.62 The seriously injured child.

Figure 6.63 Other emergency interventions.

Figure 6.64 Basic life support.

Figure 6.65 Advanced life support (Fig. 6.6).

Figure 6.66 The seriously injured child.

Figure 6.67 Other emergency interventions.

Figure 6.68 Basic life support.

Figure 6.69 Advanced life support (Fig. 6.6).

Figure 6.70 The seriously injured child.

Figure 6.71 Other emergency interventions.

Figure 6.72 Basic life support.

Figure 6.73 Advanced life support (Fig. 6.6).

Figure 6.74 The seriously injured child.

Figure 6.75 Other emergency interventions.

Figure 6.76 Basic life support.

Figure 6.77 Advanced life support (Fig. 6.6).

Figure 6.78 The seriously injured child.

Figure 6.79 Other emergency interventions.

Figure 6.80 Basic life support.

Figure 6.81 Advanced life support (Fig. 6.6).

Figure 6.82 The seriously injured child.

Figure 6.83 Other emergency interventions.

Figure 6.84 Basic life support.

Figure 6.85 Advanced life support (Fig. 6.6).

Figure 6.86 The seriously injured child.

Figure 6.87 Other emergency interventions.

Figure 6.88 Basic life support.

Figure 6.89 Advanced life support (Fig. 6.6).

Figure 6.90 The seriously injured child.

Figure 6.91 Other emergency interventions.

Figure 6.92 Basic life support.

Figure 6.93 Advanced life support (Fig. 6.6).

Figure 6.94 The seriously injured child.

Figure 6.95 Other emergency interventions.

Figure 6.96 Basic life support.

Figure 6.97 Advanced life support (Fig. 6.6).

Figure 6.98 The seriously injured child.

Figure 6.99 Other emergency interventions.

Figure 6.100 Basic life support.

Figure 6.101 Advanced life support (Fig. 6.6).

Figure 6.102 The seriously injured child.

Figure 6.103 Other emergency interventions.

Figure 6.104 Basic life support.

Figure 6.105 Advanced life support (Fig. 6.6).

Figure 6.106 The seriously injured child.

Figure 6.107 Other emergency interventions.

Figure 6.108 Basic life support.

Figure 6.109 Advanced life support (Fig. 6.6).

Figure 6.110 The seriously injured child.

Figure 6.111 Other emergency interventions.

Figure 6.112 Basic life support.

Figure 6.113 Advanced life support (Fig. 6.6).

Figure 6.114 The seriously injured child.

Figure 6.115 Other emergency interventions.

Figure 6.116 Basic life support.

Figure 6.117 Advanced life support (Fig. 6.6).

Figure 6.118 The seriously injured child.

Figure 6.119 Other emergency interventions.

Figure 6.120 Basic life support.

Figure 6.121 Advanced life support (Fig. 6.6).

Figure 6.122 The seriously injured child.

Figure 6.123 Other emergency interventions.

Figure 6.124 Basic life support.

Figure 6.125 Advanced life support (Fig. 6.6).

Figure 6.126 The seriously injured child.

Figure 6.127 Other emergency interventions.

Figure 6.128 Basic life support.

Figure 6.129 Advanced life support (Fig. 6.6).

Figure 6.130 The seriously injured child.

Figure 6.131 Other emergency interventions.

Figure 6.132 Basic life support.

Figure 6.133 Advanced life support (Fig. 6.6).

Figure 6.134 The seriously injured child.

Figure 6.135 Other emergency interventions.

Figure 6.136 Basic life support.

Figure 6.137 Advanced life support (Fig. 6.6).

Figure 6.138 The seriously injured child.

Figure 6.139 Other emergency interventions.

Figure 6.140 Basic life support.

Figure 6.141 Advanced life support (Fig. 6.6).

Figure 6.142 The seriously injured child.

Figure 6.143 Other emergency interventions.

Figure 6.144 Basic life support.

Figure 6.145 Advanced life support (Fig. 6.6).

Figure 6.146 The seriously injured child.

Figure 6.147 Other emergency interventions.

Figure 6.148 Basic life support.

Figure 6.149 Advanced life support (Fig. 6.6).

Figure 6.150 The seriously injured child.

Figure 6.151 Other emergency interventions.

Figure 6.152 Basic life support.

Figure 6.153 Advanced life support (Fig. 6.6).

Figure 6.154 The seriously injured child.

Figure 6.155 Other emergency interventions.

Figure 6.156 Basic life support.

Figure 6.157 Advanced life support (Fig. 6.6).

Figure 6.158 The seriously injured child.

Figure 6.159 Other emergency interventions.

Figure 6.160 Basic life support.

Figure 6.161 Advanced life support (Fig. 6.6).

Figure 6.162 The seriously injured child.

Figure 6.163 Other emergency interventions.

Figure 6.164 Basic life support.

Figure 6.165 Advanced life support (Fig. 6.6).

Figure 6.166 The seriously injured child.

Figure 6.167 Other emergency interventions.

Figure 6.168 Basic life support.

Figure 6.169 Advanced life support (Fig. 6.6).

Figure 6.170 The seriously injured child.

Figure 6.171 Other emergency interventions.

Figure 6.172 Basic life support.

Figure 6.173 Advanced life support (Fig. 6.6).

Figure 6.174 The seriously injured child.

Figure 6.175 Other emergency interventions.

Figure 6.176 Basic life support.

Figure 6.177 Advanced life support (Fig. 6.6).

Figure 6.178 The seriously injured child.

Figure 6.179 Other emergency interventions.

Figure 6.180 Basic life support.

Figure 6.181 Advanced life support (Fig. 6.6).

Figure 6.182 The seriously injured child.

Figure 6.183 Other emergency interventions.

Figure 6.184 Basic life support.

Figure 6.185 Advanced life support (Fig. 6.6).

Figure 6.186 The seriously injured child.

Figure 6.187 Other emergency interventions.

Figure 6.188 Basic life support.

Figure 6.189 Advanced life support (Fig. 6.6).

Figure 6.190 The seriously injured child.

Figure 6.191 Other emergency interventions.

Figure 6.192 Basic life support.

Figure 6.193 Advanced life support (Fig. 6.6).

Figure 6.194 The seriously injured child.

Figure 6.195 Other emergency interventions.

Figure 6.196 Basic life support.

Figure 6.197 Advanced life support (Fig. 6.6).

Figure 6.198 The seriously injured child.

Figure 6.199 Other emergency interventions.

Figure 6.200 Basic life support.

Figure 6.201 Advanced life support (Fig. 6.6).

Figure 6.202 The seriously injured child.

Figure 6.203 Other emergency interventions.

Figure 6.204 Basic life support.

Figure 6.205 Advanced life support (Fig. 6.6).

Figure 6.206 The seriously injured child.

Figure 6.207 Other emergency interventions.

Figure 6.208 Basic life support.

Figure 6.209 Advanced life support (Fig. 6.6).

Figure 6.210 The seriously injured child.

Figure 6.211 Other emergency interventions.

Figure 6.212 Basic life support.

Figure 6.213 Advanced life support (Fig. 6.6).

Figure 6.214 The seriously injured child.

Figure 6.215 Other emergency interventions.

Figure 6.216 Basic life support.

Figure 6.217 Advanced life support (Fig. 6.6).

Figure 6.218 The seriously injured child.

