Look in the Middle: Structural Anchor Pruning for Scalable Visual RAG Indexing

Anonymous ACL submission

Abstract

Recent Vision-Language Models (e.g., ColPali) enable fine-grained Visual Document Retrieval (VDR) but incur prohibitive index vector size overheads. Training-free pruning solutions (e.g., EOS-attention based methods) can reduce index vector size by approximately 60% without model adaptation, but often underperform random selection in high-compression scenarios ($\geq 80\%$). Prior research (e.g., Light-ColPali) attributes this to the conclusion that visual token importance is inherently query-dependent, thereby questioning the feasibility of training-free pruning. In this work, we propose Structural Anchor Pruning (SAP), a training-free pruning method that identifies key visual patches from middle layers to achieve high performance compression. We also introduce Oracle Score Retention (OSR) protocol to evaluate how layer-wise information affects compression efficiency. Evaluations on the ViDoRe benchmark demonstrate that SAP reduces index vectors by over 90% while maintaining robust retrieval fidelity, providing a highly scalable solution for Visual RAG. Furthermore, our OSR-based analysis reveals that semantic structural anchor patches persist in the middle layers, unlike traditional pruning solutions that focus on the final layer where structural signals dissipate.

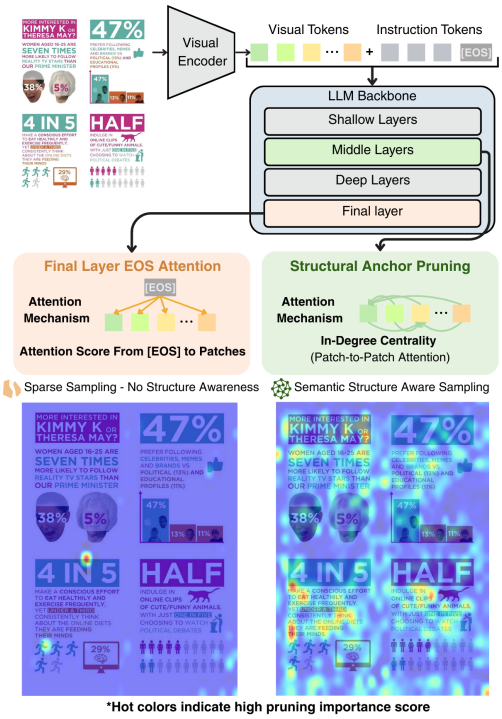

Figure 1: **Comparison of Pruning Mechanisms.** The left panel illustrates Final Layer EOS Attention, often fails to capture semantic structure. The right panel depicts our **Structural Anchor Pruning**, which utilizes **In-Degree Centrality** within the **Middle Layers** of the LLM backbone. This approach effectively identifies and preserves semantic structural anchor patches.

1 Introduction

Visual Document Retrieval (VDR) has shifted from traditional pipelines to end-to-end Vision-Language Models (VLMs) (Zhang et al., 2024). VLM-based retrievers like ColPali (Faysse et al., 2024) achieve superior precision by representing documents as bags of visual patch embeddings. While this paradigm captures rich document structures through late interaction (Khattab and Zaharia, 2020), it suffers from massive index size overhead. Addressing this scalability challenge through embedding compression is essential for deploying Visual RAG in realistic, large-scale scenarios.

To address this bottleneck, recent research has diverged into two primary streams. One involves training-based methods like Light-ColPali (Ma et al., 2025) which are effective but require fully re-training and additional model modifications. Alternatively, training-free methods like DocPruner (Yan et al., 2025) offer a modular solution, but these methods often suffer from performance degradation in high-compression regimes ($\geq 80\%$ reduction) (Ma et al., 2025; Yan et al., 2025). Observing these challenges, (Ma et al., 2025) conclude that visual token importance for

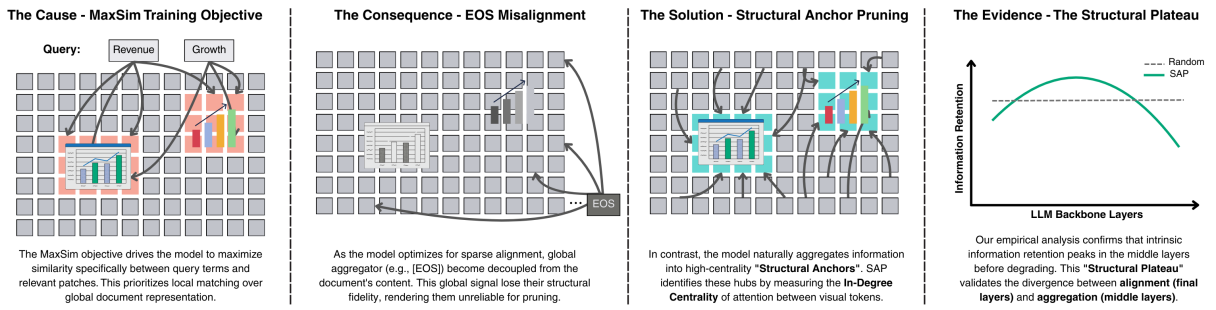

Figure 2: **The Mechanics of SAP.** We illustrate the *Alignment-Aggregation Divergence*. Unlike final layers where global signals decay due to MaxSim optimization, the middle layers naturally aggregate information into high-centrality semantic structural anchor patches. SAP exploits this by measuring the **In-Degree Centrality** between visual tokens to identify key patches without query supervision.

pruning is inherently query-dependent, thereby arguing that training-free pruning is insufficient for high compression ratio.

In this work, we challenge this conclusion. Firstly, we propose a training-free, query-agnostic method **Structural Anchor Pruning (SAP)**. Unlike prior methods, SAP identifies key tokens from middle layers in the VLM backbone, preserving the document's intrinsic semantic structural anchor patches (see Figure 1). Extensive evaluations on the ViDoRe dataset (Faysse et al., 2024; Macé et al., 2025) confirm that SAP consistently outperforms EOS-Adaptive (Yan et al., 2025), Random (Ma et al., 2025), and Semantic Clustering (Ma et al., 2025) baselines. Notably, our method reduces the number of stored vectors by over 90% compared to the full-page index while maintaining robust retrieval fidelity. This offers a scalable, zero-shot solution for efficient visual RAG.

We also introduce the **Oracle Score Retention (OSR)** protocol, a diagnostic metric designed to isolate intrinsic information retention from corpus information. Through this, we uncover the **Alignment-Aggregation Divergence**, illustrated in Figure 2. We observe that as the model approaches its final layers, it optimizes for sparse query alignment, causing global structural information to dissipate. Consequently, final-layer signals become poor proxies for document structure. In contrast, middle layers effectively provide informative structural signal for effective pruning.

2 Related Work and Preliminaries

2.1 VDR Multi-Vector Late Interaction

Visual Document Retrieval has recently shifted from traditional OCR-based pipelines to end-to-end Vision-Language Models (VLMs) (Zhang et al., 2024). Unlike dense retrievers that map a document to a single vector, models such as ColPali (Faysse et al., 2024) employ a **Multi-Vector Late Interaction** mechanism (Khattab and Zaharia, 2020).

Formally, a document image D is encoded into a bag of visual patch embeddings $E_D = \{v_1, \ldots, v_N\} \in \mathbb{R}^{N \times d}$, where N represents the sequence length (typically 1024 patches per image). Given a text query Q encoded as tokens $\{q_1, \ldots, q_M\}$, the relevance score is computed via the MaxSim operator:

$$S(Q, D) = \sum_{i=1}^{M} \max_{j=1}^{N} (q_i \cdot v_j) \tag{1}$$

This mechanism bridges visual perception and semantic retrieval by preserving fine-grained layout details. However, it necessitates indexing the full matrix E_D, leading to index vector size that scales linearly with N. For realistic corpora, this results in terabytes of index vector storage (Xu et al., 2025), creating the fundamental bottleneck that necessitates the pruning strategies discussed below.

2.2 Efficient Visual Document Retrieval

To mitigate the index vector size overhead defined in Section 2.1, recent research has diverged into two primary streams: training-based adaptation and training-free pruning.

Training-based Adaptation. Methods like Light-ColPali (Ma et al., 2025) employ knowledge distillation to train adapters that merge visual tokens into a smaller set of latent representations. While effective at high compression ratios, these approaches introduce significant operational overhead, requiring large-scale of training datasets and full-model fine-tuning, which limits their zero-shot applicability to new architectures.

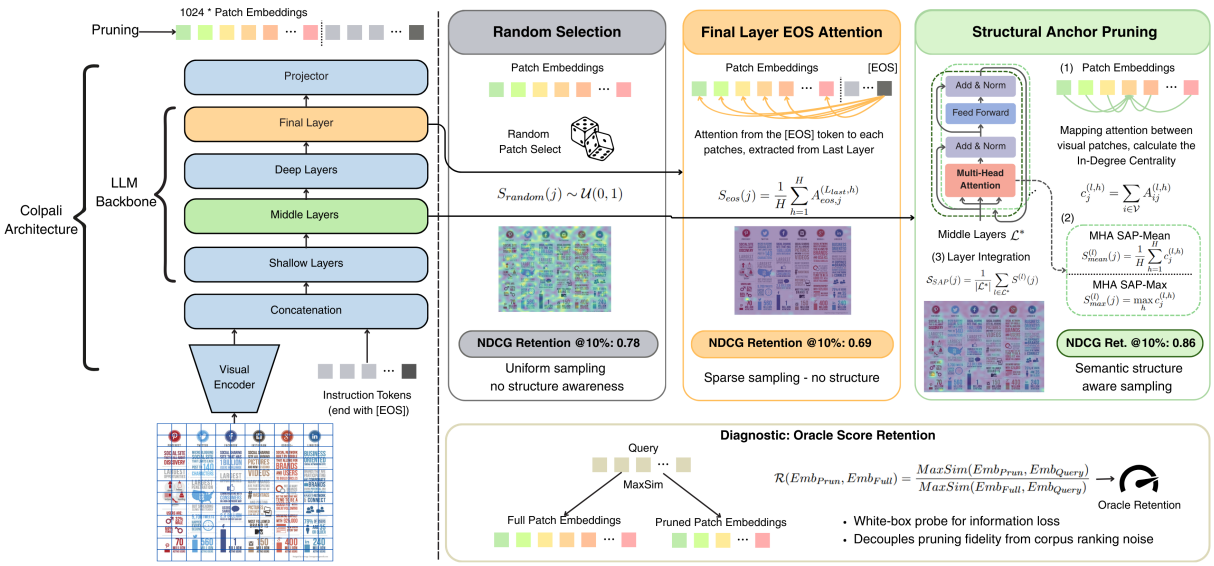

Figure 3: **Overview of SAP.** We compare three pruning paradigms on the ColPali architecture. **Left:** The shared Vision-Language backbone processes the image. **Middle:** Conventional methods (Random Selection, Final Layer EOS Attention) fail to identify critical tokens, resulting in lower retention. **Right:** Our proposed SAP method identifies semantic structural anchor patches via In-Degree Centrality in the model's middle layers, achieving high retrieval performance retention on the ViDoRe v2 benchmark by preserving the document's semantic structure. **Bottom:** We illustrate the **Oracle Score Retention** protocol, a white-box diagnostic used to validate our hypothesis. This metric directly compares the MaxSim scores of pruned versus full embeddings, isolating intrinsic information loss from corpus-dependent ranking noise.

Training-free Compression. Conversely, training-free methods select a subset of informative patches $\hat{E}_D \subset E_D$ without model adaptation. Current strategies typically rely on method signals: (1) **Random Pruning** (Ma et al., 2025) assumes a holographic distribution of visual information and selects patches via uniform pruning; (2) **Semantic Clustering** (Ma et al., 2025) aims to reduce redundancy by grouping embeddings via K-Means and indexing only the cluster centroids; (3) **EOS-Attention** (Ma et al., 2025) selects patches based on their cross-attention weights with the final `[EOS]` token; and (4) **EOS-Adaptive Pruning (DocPruner)** (Yan et al., 2025) extends this by dynamically adjusting the pruning ratio based on information density. However, these training-free methods suffer from severe performance degradation when pushed to high compression ratios (e.g., $> 80\%$ reduction) (Yan et al., 2025; Ma et al., 2025). Consequently, they argue that static, query-agnostic pruning strategies are insufficient for high ratio compression (Ma et al., 2025).

3 Methodology

To address the limitations of existing training-free pruning—specifically their degradation in high-compression, we firstly introduce **Structural Anchor Pruning**. By shifting focus from the retrieval-aligned final layers to the middle layers, SAP challenges the prevailing assumption that visual token importance is inherently query-dependent. Secondly, to rigorously validate the theoretical basis of our method, we establish the **Oracle Score Retention** protocol. Unlike standard ranking metrics which confound information loss with corpus noise, OSR provides a diagnostic metric to analyze independent retrieval score retention. The overall framework of our approach is illustrated in Figure 3.

3.1 Structural Anchor Pruning

We propose SAP, a training-free strategy designed to extract the intrinsic semantic structure of a document image. We identify semantic structural anchor patches by measuring the visual In-Degree Centrality of tokens within the **Large language Model (LLM) backbone**. We hypothesize that semantic structural patches in middle layers acting as information hubs, aggregating features from numerous other regions, constitute the core semantic representation of the document.

Visual In-Degree Centrality. We treat the self-attention mechanism within the LLM layers at any

given layer l as a directed graph, where nodes represent image patches and edges represent attention weights. Mechanistically, the attention weight A_{ij} represents the importance score of token j for token i. Consequently, the summation over all query indices, $\sum_i A_{ij}$, quantifies the total importance of token j across all visual tokens, serving as a direct proxy for its global influence.

To isolate the visual structure, we restrict our calculation to the **Visual-to-Visual** attention, masking out attention scores involving text tokens (e.g., system prompts). Let $A^{(l,h)} \in \mathbb{R}^{T \times T}$ be the full attention matrix for head h over sequence length T, and $\mathcal{V}$ be the set of indices corresponding to visual patches. The importance of a visual patch $j \in \mathcal{V}$ at layer l is defined by its column-sum:

$$c_j^{(l,h)} = \sum_{i \in \mathcal{V}} A_{ij}^{(l,h)} \tag{2}$$

A high in-degree indicates that patch j acts as a central aggregator within the visual modality.

Head Aggregation. To synthesize signals across the H attention heads at layer l, we propose two variants:

- **SAP-Mean:** Computes the average centrality, prioritizing anchors consistently active across the attention subspace.

$$S_{mean}^{(l)}(j) = \frac{1}{H} \sum_{h=1}^{H} c_j^{(l,h)} \tag{3}$$

- **SAP-Max:** Captures the peak prominence of a token within its most dominant attention head, preventing strong local signals from being diluted.

$$S_{max}^{(l)}(j) = \max_h c_j^{(l,h)} \tag{4}$$

Layer Integration. Standard pruning approaches typically rely on the final layer ($l = L_{total}$) under the assumption that it represents the most refined semantic state. However, we hypothesize that middle layers function as an *aggregation phase*, where tokens actively exchange information to build a cohesive structural understanding of the document. Conversely, the final layers shift towards an *alignment phase*, where representations are implicitly reorganized to optimize the contrastive retrieval objective (MaxSim). This final alignment often "sparsifies" the attention map to fit potential query distributions, thereby degrading the intrinsic structural signals required for effective pruning.

To capture the robust structural core before this degradation occurs, we introduce **Layer Integration**. We define the layer ensemble $\mathcal{L}^*$ as a function of the model's total depth L_{total} and relative depth hyperparameters $\alpha, \beta \in [0, 1]$:

$$\mathcal{L}^*(\alpha, \beta) = \{l \in \mathbb{N} \mid \lfloor \alpha \cdot L_{total} \rfloor \leq l \leq \lfloor \beta \cdot L_{total} \rfloor\} \tag{5}$$

where $0 \leq \alpha < \beta \leq 1$ define the boundaries of the structural window (typically the middle block). The final importance score $\mathcal{S}_{SAP}(j)$ is obtained by averaging the centrality scores across this window:

$$\mathcal{S}_{SAP}(j) = \frac{1}{|\mathcal{L}^*|} \sum_{l \in \mathcal{L}^*} S^{(l)}(j) \tag{6}$$

3.2 Oracle Score Retention

While standard evaluation metrics like Normalized Discounted Cumulative Gain (NDCG) (Wang et al., 2013) are essential for assessing retrieval effectiveness, they are insufficient for diagnosing the intrinsic fidelity of pruned representations. Formally, NDCG at position k is defined as:

$$\text{NDCG@}k = \frac{1}{\text{IDCG}_k} \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2(i+1)} \tag{7}$$

where rel_i is the relevance score, i is the ranking position, and IDCG_k is the Ideal Discounted Cumulative Gain, acting as a normalization factor to ensure the score lies in $[0, 1]$. Crucially, the logarithmic term $\log_2(i+1)$ explicitly couples the evaluation to the relative rank i. This makes the metric inherently corpus-dependent: a drop in NDCG may result from the presence of hard negatives shifting the rank i, rather than a loss of information in the document representation itself. Consequently, NDCG confounds pure information loss with the model's discriminative capacity by measuring pruning performance.

To disentangle these factors and isolate the intrinsic visual information retained by a pruning method, we introduce the **Oracle Evaluation Protocol**. We define fidelity not by ranking position, but by the preservation of the raw MaxSim score.

For a given query-document pair, we define the **Oracle Score Retention** as:

$$\mathcal{R}(\hat{E}_D, E_D) = \frac{\sum_{i=1}^{M} \max_{v \in \hat{E}_D} (q_i \cdot v)}{\sum_{i=1}^{M} \max_{v \in E_D} (q_i \cdot v)} \tag{8}$$

This metric functions as a diagnostic indicator: a retention of 1.0 confirms that the pruned patches $\hat{E}_D$ preserve the exact visual features triggered by the query, independent of the document's ranking relative to distractors.

4 Evaluation

In this section, we comprehensively benchmark the performance of SAP on large-scale visual retrieval tasks. We evaluate SAP's ability to maintain high retrieval fidelity across diverse architectures and datasets, compare it against state-of-the-art training-free and training-based baselines, and assess its computational efficiency.

4.1 Evaluation Setup

Diverse VLM Backbones. We employ three distinct VLM architectures to evaluate SAP: **ColPali** (SigLIP + PaliGemma) (Beyer et al., 2024; Faysse et al., 2024), representing the standard fixed-patch retrieval paradigm, and **ColQwen2** (NaViT + Qwen2-VL) (Wang et al., 2024; Faysse et al., 2024), representing a dynamic-resolution architecture with a deeper backbone. We also extend our evaluation to a SOTA model **Jina Embeddings v4** with Qwen2.5-VL backbone (Günther et al., 2025; Bai et al., 2025). Incorporating these architectures allows us to assess the generalizability of SAP across different VLM backbones and distinct embedding optimization recipes. See Appendix A for model architectural specifications.

Layer Integration Initialization. To ensure zero-shot adaptability across varying backbone depths, we instantiate the layer ensemble $\mathcal{L}^*$ (Eq. 6) using a simple fixed Geometric Central Window of $40\% \sim 60\%$ relative depth ($\mathcal{L}^* = \{l \mid \lfloor 0.4L_{total} \rfloor \leq l \leq \lfloor 0.6L_{total} \rfloor\}$). Specific layer indices for each model are detailed in **Appendix B**.

Baselines. We compare our **SAP-Mean** and **SAP-Max** against three distinct training-free pruning paradigms (details in Appendix C):

(1) Adaptive-EOS: An EOS attention-based method proposed by DocPruner (Yan et al., 2025), which employs document-specific thresholding based on final-layer global (`[EOS]`) attention scores. To ensure fair comparison, we apply a quantile-based calibration to align its global retention rate strictly with our fixed-ratio methods; (2) Random: The robust stochastic pruning baseline (Ma et al., 2025); and (3) Semantic Cluster: K-Means clustering on final embeddings, identified by (Ma et al., 2025) as the state-of-the-art training-free compression approach.

Datasets. We utilize the full **ViDoRe v1** (Faysse et al., 2024) and **ViDoRe v2** (Macé et al., 2025) benchmarks. These cover a wide spectrum of domains. Detailed dataset statistics are provided in Appendix D.

4.2 Main Results on ViDoRe

Table 1 presents the aggregated performance across all datasets. Appendix E shows detailed evaluation results for each sub-datasets and models.

Performance Consistency across High Compression Regimes. SAP demonstrates stability across varying degrees of sparsity. As detailed in Table 1, at the aggressive retention ratios of $\gamma = 0.20$ and $\gamma = 0.10$, our method consistently retains around 95% and 90% of the original retrieval performance across benchmarks.

Universality across Architectures. SAP delivers consistent gains across different retrieval VLM backbones. Notably, on the Jina v4 (Günther et al., 2025) architecture, SAP achieves substantial NDCG retention on both ViDoRe v1 and the more challenging ViDoRe v2, outperforming other baselines. Consequently, SAP offers a plug-and-play compression solution that generalizes zero-shot to diverse architectures without requiring any model-specific tuning.

4.3 Efficiency-Fidelity Trade-off

Figure 4 illustrates the NDCG@5 retention performance across a broad spectrum of compression ratios. SAP demonstrates remarkable robustness under a large range of compression regimes. While methods like Cluster-Merge and Adaptive-EOS suffer from rapid degradation as the keep ratio decreases, SAP maintains high fidelity from moderate ($\gamma = 0.9$) down to aggressive ($\gamma = 0.1$) sparsity levels. This stability confirms that our structural anchor identification is effective regardless of the target storage constraint.

4.4 Computational Efficiency

Beyond retrieval fidelity, SAP maintains high operational throughput. Theoretical complexity analysis and empirical benchmarks (Appendix F.2) show that SAP variants add negligible overhead to the total forward pass latency. In contrast, clustering-

Model	Method	Upper Bound	$\gamma = 0.20$			$\gamma = 0.10$			$\gamma = 0.05$		
		Full NDCG	S.Ret	NDCG	%	S.Ret	NDCG	%	S.Ret	NDCG	%
Benchmark: ViDoRe v1 (Avg. across 10 datasets)											
ColPali	EOS-Adaptive	0.85	0.86	0.76	89.03	0.79	0.69	80.60	0.72	0.62	71.32
	Random		0.91	0.80	94.27	0.85	0.76	89.36	0.78	0.71	83.22
	Cluster		0.89	0.82	96.75	0.80	0.78	91.05	0.69	0.67	78.35
	SAP-Mean (Ours)		0.94	0.83	97.43	0.89	0.79	93.08	0.83	0.75	87.47
	SAP-Max (Ours)		**0.94**	**0.83**	**98.05**	**0.89**	**0.80**	**93.52**	**0.83**	**0.75**	**87.59**
ColQwen2	EOS-Adaptive	0.88	0.84	0.77	87.44	0.75	0.70	78.98	0.65	0.63	70.05
	Random		0.86	0.82	93.44	0.78	0.77	87.13	0.68	0.69	78.09
	Cluster		0.82	0.84	95.55	0.70	0.79	90.20	0.59	0.72	81.99
	SAP-Mean (Ours)		**0.91**	**0.85**	**96.36**	**0.84**	**0.80**	**91.24**	**0.75**	**0.73**	**82.86**
	SAP-Max (Ours)		0.91	0.84	96.14	0.84	0.80	90.55	0.75	0.73	82.26
Jina Embeddings v4	EOS-Adaptive	0.90	0.86	0.85	94.49	0.79	0.77	85.39	0.70	0.65	71.93
	Random		0.89	0.86	95.32	0.82	0.81	90.20	0.74	0.75	82.19
	Cluster		0.83	0.86	96.01	0.73	0.81	90.40	0.62	0.75	83.20
	SAP-Mean (Ours)		**0.91**	**0.87**	**96.93**	**0.85**	**0.83**	**92.10**	**0.78**	**0.76**	**84.29**
	SAP-Max (Ours)		0.91	0.87	96.93	0.85	0.82	91.43	0.76	0.74	81.74
Benchmark: ViDoRe v2 (Avg. across 4 datasets)											
ColPali	EOS-Adaptive	0.56	0.86	0.44	78.60	0.79	0.38	68.96	0.70	0.32	56.94
	Random		0.90	0.49	86.89	0.84	0.44	78.04	0.77	0.39	69.28
	Cluster		0.88	0.47	84.00	0.78	0.39	69.41	0.67	0.29	52.04
	SAP-Mean (Ours)		0.93	**0.52**	**92.57**	**0.88**	**0.48**	**86.15**	0.81	**0.44**	**78.32**
	SAP-Max (Ours)		**0.94**	0.51	91.09	0.88	0.48	86.13	**0.82**	0.44	78.26
ColQwen2	EOS-Adaptive	0.54	0.84	0.47	87.27	0.75	0.41	76.50	0.66	0.33	62.35
	Random		0.87	0.46	85.61	0.78	0.41	75.96	0.69	0.36	66.16
	Cluster		0.80	0.45	84.05	0.70	0.39	72.16	0.58	0.32	58.59
	SAP-Mean (Ours)		0.89	0.48	88.61	0.82	**0.44**	**81.44**	0.74	**0.40**	**75.18**
	SAP-Max (Ours)		**0.89**	**0.49**	**90.79**	**0.82**	0.44	81.13	**0.74**	0.40	74.82
Jina Embeddings v4	EOS-Adaptive	0.58	0.89	0.53	90.91	0.82	0.47	81.47	0.73	0.38	66.41
	Random		0.90	0.51	88.46	0.83	0.46	78.49	0.75	0.39	67.83
	Cluster		0.83	0.47	80.81	0.73	0.40	67.98	0.63	0.33	57.07
	SAP-Mean (Ours)		0.91	0.55	95.60	0.85	0.51	88.49	**0.78**	**0.45**	**77.14**
	SAP-Max (Ours)		**0.91**	**0.56**	**97.18**	**0.85**	**0.52**	**89.24**	0.78	0.43	73.33

Table 1: **Main Results on ViDoRe Benchmarks.** Comparative analysis against baselines across three architectures and two benchmark suites. **Upper Bound** denotes the full model performance ($\gamma = 1.0$). The **%** column indicates the relative NDCG retention ($\frac{\text{Pruned}}{\text{Full}} \times 100$).

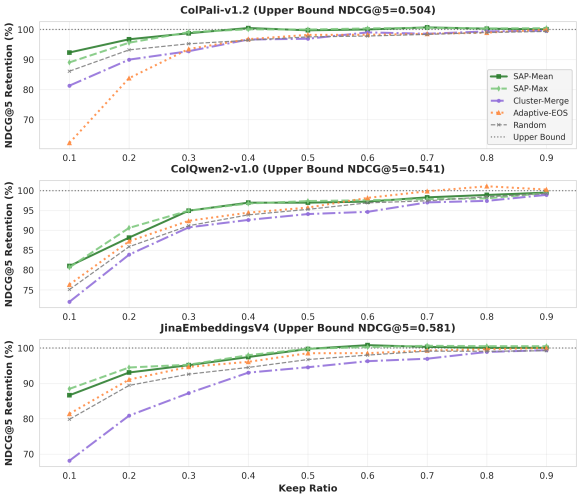

Figure 4: **Efficiency-Fidelity Trade-off.** Impact of pruning ratio on NDCG@5 Retention across ColPali, ColQwen2, and JinaEmbeddingsV4 on ViDoRe v2. SAP methods (green) exhibit exceptional stability, significantly outperforming clustering and other pruning baselines at low keep ratios.

based method incurs a significant 6% computational overhead.

4.5 Comparison with Trained Method

We benchmark SAP against **Light-ColPali** (Ma et al., 2025), a state-of-the-art method that requires supervised training to merge visual tokens.

As shown in Table 2, SAP demonstrates remarkable efficiency. While the trained method excels at extreme compression (25×) via feature fusion, SAP remains robust at high compression regimes (4× and 9×). This highlights that semantic structural anchor patches naturally capture the majority of retrieval signals, offering a compelling zero-cost alternative that eliminates the need for architectural modifications and full-model re-training.

4.6 OSR as a Reliable Proxy

As illustrated in Figure 5, we assess the efficacy of our diagnostic protocol by observing a strong linear correlation (Pearson $r = 0.635$) between

Table 2: **Training-Free vs. Trained Compression.** We compare SAP against Light-ColPali (Faysse et al., 2024) (Trained Merging method). We report NDCG@5 and the relative retention percentage. Note that the Upper Bounds (Full Model performance) differ slightly due to evaluation environments.

Backbone	Method	Training Req.	Compression Factor 4×	9×	25×
ColPali	Light-ColPali†	Yes	$0.75_{98.7}$	$\mathbf{0.75_{98.2}}$	$\mathbf{0.72_{94.8}}$
	SAP-Mean (Ours)	**No**	$0.76_{98.2}$	$0.71_{92.8}$	$0.63_{81.5}$
	SAP-Max (Ours)	**No**	$\mathbf{0.76_{98.7}}$	$0.71_{92.5}$	$0.62_{79.9}$
ColQwen2	Light-ColQwen2†	Yes	$\mathbf{0.82_{99.7}}$	$\mathbf{0.81_{98.8}}$	$\mathbf{0.80_{97.5}}$
	SAP-Mean (Ours)	**No**	$0.79_{96.9}$	$0.75_{91.9}$	$0.64_{77.9}$
	SAP-Max (Ours)	**No**	$0.80_{97.3}$	$0.75_{91.7}$	$0.65_{78.8}$

† Results cited from Light-ColPali (Faysse et al., 2024). Subscript denotes % retention of full model.

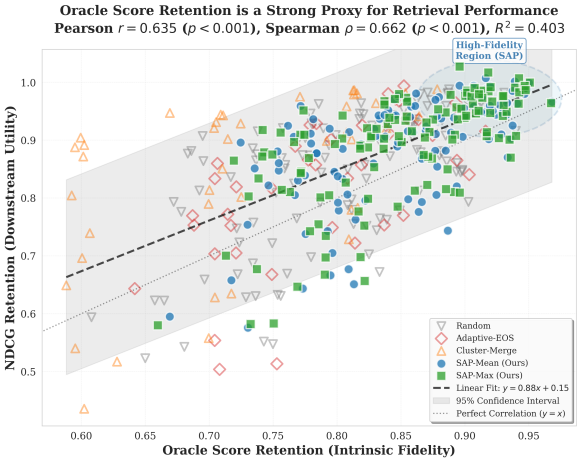

Figure 5: **Oracle Score Retention is a Strong Proxy for Retrieval Performance.** We observe a significant positive correlation between intrinsic score preservation and downstream ranking utility. Each data point corresponds to a unique evaluation configuration defined by the model architecture, compression ratio, and dataset subset.

the intrinsic Oracle Score Retention and the final NDCG performance. The plot reveals a distinct separation of methods: while baseline approaches like Adaptive-EOS and Random (hollow markers) exhibit higher variance, our SAP variants (solid markers) consistently occupy the top-right "High-Fidelity Region", where pruned representations maintaining both high NDCG and Oracle Score Retention.

5 Alignment-Aggregation Divergence

Having established the superior retrieval performance of SAP and validated the OSR as a reliable proxy, we now turn to the mechanistic question: *Why does the semantic signal necessary for pruning decouple from the final retrieval embedding?* In this section, we utilize OSR as a "white-box" probe to scan the information retention capabilities across the model's depth. This diagnostic reveals the **Alignment-Aggregation Divergence**—a phenomenon where the model's structural understanding peaks in middle layers before degrading as it aligns with the sparse retrieval objective.

To isolate the location of structural information, we apply the OSR protocol to every layer of the LLM backbone in **ColPali** (Beyer et al., 2024; Faysse et al., 2024) and **ColQwen2** (Wang et al., 2024; Faysse et al., 2024) architectures. We utilize the **ViDoRe** benchmark (Faysse et al., 2024), focusing on five diverse subsets that categorize document morphology: Text-Centric (ArxivQA (Li et al., 2024), DocVQA (Mathew et al., 2021)), Structure-Centric (TabFQuAD, TAT-DQA (Zhu et al., 2022)), and Layout-Centric (InfoVQA (Mathew et al., 2022)). This diversity allows us to test if semantic structural anchor patches remain stable across different visual modalities. As visualized in Figure 6, our analysis uncovers two distinct phases in the VLM's internal processing.

The Aggregation Phase (The Structural Plateau). In the middle layers, we observe a sustained peak in retention scores across all document morphologies—a region of stability we identify as the **Structural Plateau** (highlighted in blue in Figure 6). Mechanistically, this corresponds to the formation of structural anchor patches, where the model aggregates local visual features into high-centrality anchors to build a global understanding of the document. This morphological robustness indicates that the document's "semantic core" is naturally concentrated in these middle layers.

The Alignment Phase (Final Layers). Approaching the final layers, the retention metric exhibits a pronounced decline. We attribute this to the **Late-Interaction MaxSim objective**. We hypothesize that to maximize contrastive separability, the model reorganizes its representation to align strictly with potential query tokens, implicitly "sparsifying" the information. This suggests that while beneficial for retrieval ranking, such optimization results in the loss of dense structural context accumulated in the middle layers, rendering the final attention weights suboptimal proxies for visual importance.

SAP vs. EOS Attention This layer-wise divergence explains the limitations of prior pruning methods. As shown in Figure 6, EOS-Attention

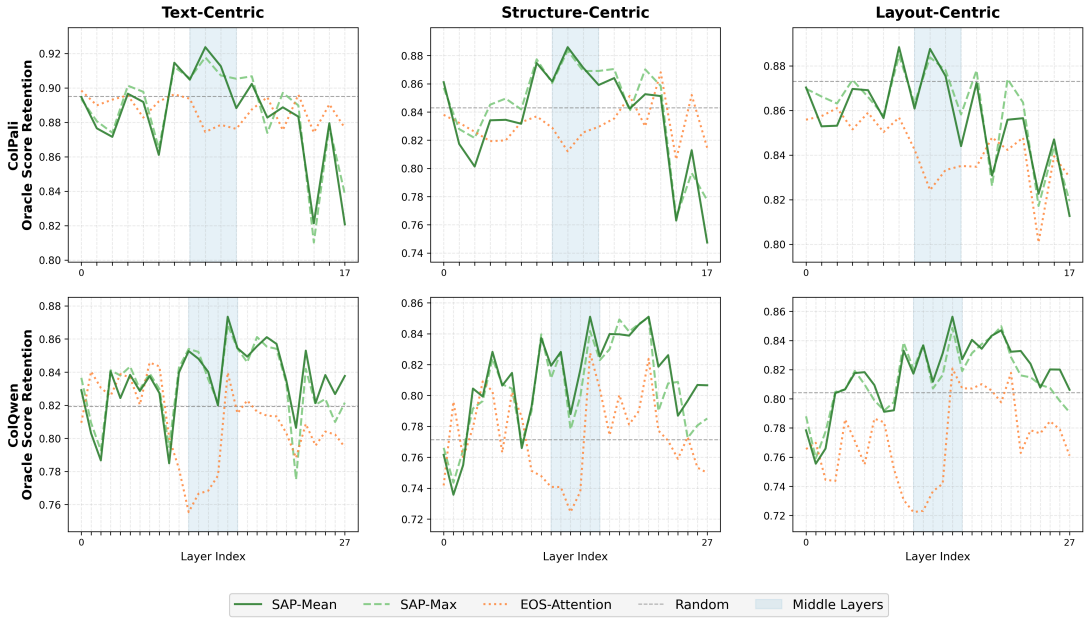

Figure 6: **Morphological Robustness Analysis.** Oracle Score Retention curves decomposed by document type at a high-retention ratio ($\gamma = 0.1$). We observe a universal "Structural Plateau" in the middle layers regardless of document morphology (Text, Tables, or Layouts), in contrast to the information loss observed in final layers.

(which relies on the final layer) consistently yields lower scores compared to SAP, and often falls below the random pruning baseline. This empirical gap serves as an indication that the MaxSim training objective decouples the global [EOS] token from the document's structural content.

6 Conclusion

In this work, we address the critical index scalability bottleneck inherent to multi-vector VLM-based VDR systems. Challenging the prevailing view that training-free pruning is insufficient for high compression due to query dependency, we propose Structural Anchor Pruning (SAP). This zero-shot, query-agnostic approach successfully reduces index storage by over 90% while maintaining robust retrieval fidelity, consistently outperforming existing baselines on the ViDoRe benchmark. Furthermore, through our Oracle Score Retention (OSR) protocol, we uncover the underlying Alignment-Aggregation Divergence, demonstrating that unlike the sparse, alignment-optimized final layers, middle-layer representations retain essential semantic structural signals, thereby offering a highly efficient and scalable solution for Visual RAG.

7 Limitations

While SAP offers a scalable solution for Visual RAG, our scope is currently confined to the multi-vector late-interaction paradigm, leaving its generalizability to broader image-text matching tasks and large-scale industrial indices to be fully explored. Methodologically, the framework relies on empirically fixed parameters for layer selection and enforces a uniform token budget across all documents. Future research could address this rigidity by developing dynamic mechanisms that adaptively select LLM backbone layers and adjust index vector capacity based on instance-specific document complexity, enabling flexible model adaptation and variable compression rates across diverse document types.

References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, and 1 others. 2024. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*.

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

Michael Günther, Saba Sturua, Mohammad Kalim Akram, Isabelle Mohr, Andrei Ungureanu, Bo Wang, Sedigheh Eslami, Scott Martens, Maximilian Werk, Nan Wang, and 1 others. 2025. jina-embeddings-v4: Universal embeddings for multimodal multilingual retrieval. In *Proceedings of the 5th Workshop on Multilingual Representation Learning (MRL 2025)*, pages 531–550.

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

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. *arXiv preprint arXiv:2403.00231*.

Yubo Ma, Jinsong Li, Yuhang Zang, Xiaobao Wu, Xiaoyi Dong, Pan Zhang, Yuhang Cao, Haodong Duan, Jiaqi Wang, Yixin Cao, and 1 others. 2025. Towards storage-efficient visual document retrieval: An empirical study on reducing patch-level embeddings. *arXiv preprint arXiv:2506.04997*.

Quentin Macé, António Loison, and Manuel Faysse. 2025. Vidore benchmark v2: Raising the bar for visual retrieval. *arXiv preprint arXiv:2505.17166*.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. 2022. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Yining Wang, Liwei Wang, Yuanzhi Li, Di He, and Tie-Yan Liu. 2013. A theoretical analysis of ndcg type ranking measures. In *Conference on learning theory*, pages 25–54. PMLR.

Mengyao Xu, Gabriel Moreira, Ronay Ak, Radek Osmulski, Yauhen Babakhin, Zhiding Yu, Benedikt Schifferer, and Even Oldridge. 2025. Llama nemoretriever colembed: Top-performing text-image retrieval model. *arXiv preprint arXiv:2507.05513*.

Yibo Yan, Guangwei Xu, Xin Zou, Shuliang Liu, James Kwok, and Xuming Hu. 2025. Docpruner: A storage-efficient framework for multi-vector visual document retrieval via adaptive patch-level embedding pruning. *arXiv preprint arXiv:2509.23883*.

Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. 2024. Vision-language models for vision tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 46(8):5625–5644.

Fengbin Zhu, Wenqiang Lei, Fuli Feng, Chao Wang, Haozhou Zhang, and Tat-Seng Chua. 2022. Towards complex document understanding by discrete reasoning. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4857–4866.

A Model Architectures

To ensure the universality of our **Alignment-Aggregation Divergence** hypothesis, we selected three Vision-Language Models (VLMs) that represent distinct design paradigms in the current landscape. Table 3 summarizes their architectural specifications.

Model	Backbone	Layers	Params
ColPali	PaliGemma-3B	18	3B
ColQwen2	Qwen2-VL-2B	28	2B
Jina v4	Qwen2.5-VL-3B	36	3B

Table 3: **Architectural Summary.** The selected models cover different LLM families (Gemma, Qwen, Llama-style) and vision encoding strategies. Note the progression from the fixed-resolution SigLIP encoder in PaliGemma to the dynamic-resolution capabilities inherent in the Qwen2-VL series.

A.1 ColPali

ColPali (`ViDoRe/colpali-v1.3`[1]) represents the pioneering architecture for late-interaction visual retrieval, establishing the foundation for the Visual Document Retrieval (ViDoRe) benchmark. It is built upon the **PaliGemma-3B** backbone, which uniquely combines a **SigLIP-So400m** vision encoder with the Gemma-2B language model. Unlike traditional pipelines that rely on OCR to extract text, ColPali employs a Visual Large Language Model (VLLM) approach to generate multi-vector representations directly from document images. This allows it to effectively index complex visual elements—such as figures, charts, and tables—thereby significantly outperforming standard dense retrieval methods on visually rich documents.

A.2 ColQwen2

ColQwen2 (`ViDoRe/colqwen2-v1.0`[2]) is based on the advanced **Qwen2-VL-2B** architecture. This model introduces significant complexity and architectural improvements over ColPali, primarily through its support for **native dynamic resolution**. While ColPali typically resizes inputs to fixed square patches (often distorting document aspect ratios), ColQwen2 leverages the Naive Dynamic Resolution mechanism inherent to Qwen2-VL. This allows the model to process images of varying dimensions and aspect ratios without information loss, resulting in superior visual fidelity and more efficient visual token usage during the indexing of high-resolution PDFs.

A.3 Jina Embeddings v4

The **Jina Embeddings v4** model (`jinaai/jina-embeddings-v4`[3]) represents the state-of-the-art application of late-interaction principles to the powerful **Qwen2.5-VL** architecture. By transitioning to the Qwen2.5 backbone, this iteration offers enhanced optical character recognition (OCR) capabilities and improved geometric reasoning for structured data. Furthermore, it incorporates Jina AI's signature **Matryoshka Representation Learning (MRL)**, enabling flexible embedding dimensions that allow users to trade off index vector size efficiency against retrieval precision. This model aims to unify multimodal retrieval by supporting extended context windows and delivering high-performance indexing for both textual and visual-heavy datasets.

B Detailed Layer Instantiation

To ensure SAP remains calibration-free and zero-shot, we select the layer ensemble $\mathcal{L}^*$ using the simple fixed Geometric Central Window:

$$\mathcal{L}^* = \{l \in \mathbb{N} \mid \lfloor 0.4 \cdot L_{total} \rfloor \leq l \leq \lfloor 0.6 \cdot L_{total} \rfloor\} \tag{9}$$

Table 4 details the specific layers selected for each architecture used in our experiments. This method effectively targets the "Structural Plateau" where visual aggregation peaks, regardless of the total depth of the backbone.

Parameters	ColPali	ColQwen2	Jina v4
Total Layers (L)	18	28	36
Range ($0.4L \sim 0.6L$)	$7.2 \sim 10.8$	$11.2 \sim 16.8$	$14.4 \sim 21.6$
Selected $\mathcal{L}^*$	$\{7, \ldots, 10\}$	$\{11, \ldots, 16\}$	$\{14, \ldots, 21\}$

Table 4: **SAP Layer Ensemble Instantiation.** The specific middle layers used for In-Degree Centrality calculation across different architectures, derived automatically from the geometric method.

C Baseline Implementation Details

We provide the formal definitions for the baseline pruning methods used in our comparative analysis. Let $E \in \mathbb{R}^{N \times d}$ denote the sequence of N visual patch embeddings. We select a subset of size K

[1] `https://huggingface.co/ViDoRe/colpali-v1.3`
[2] `https://huggingface.co/ViDoRe/colqwen2-v1.0`
[3] `https://huggingface.co/jinaai/jina-embeddings-v4`

based on the following importance scoring functions $S(j)$ for the j-th patch.

1. Random. This method assumes visual information is holographically distributed. The selection is performed via uniform pruning without replacement:

$$S_{random}(j) \sim \mathcal{U}(0, 1) \tag{10}$$

To mitigate the impact of randomness, we conducted five independent runs initialized with distinct random seeds and reported the average performance metrics.

2. EOS-Attention. This method assumes that patches attended to during text generation are most relevant. We define the score as the cross-attention weight from the final token (representing `[EOS]`) in the last layer L_{last}:

$$S_{eos}(j) = \frac{1}{H} \sum_{h=1}^{H} A_{eos,j}^{(L_{last},h)} \tag{11}$$

3. Semantic Clustering (Post-Projector). This method assumes that visual redundancy can be reduced by grouping embeddings based on their representation similarity. Following the architectural insights from Light-ColPali (Ma et al., 2025), we specifically perform this operation at the **Post-Projector** stage—immediately after the Vision-LLM's final linear projection layer.

The empirical study indicates that clustering is significantly more effective in this low-dimensional output space (e.g., 128 dimensions) compared to high-dimensional intermediate representations, as it enables more targeted feature aggregation with minimal information loss. We apply K-Means clustering to the set of projected embeddings $E = \{v_1, \ldots, v_N\}$ to partition them into K disjoint sets $\{C_1, \ldots, C_K\}$. The objective is to minimize the within-cluster sum of squares (WCSS):

$$\min_{\{\mu_1,\ldots,\mu_K\}} \sum_{k=1}^{K} \sum_{v_j \in C_k} \|v_j - \mu_k\|^2 \tag{12}$$

where μ_k is the centroid of cluster C_k. The pruned representation consists of these K centroids, effectively merging redundant visual features into a compact, representative set ready for indexing.

4. Adaptive EOS attention. While the standard EOS-Attention method applies a fixed selection ratio (Top-K) across all documents, this baseline adopts a document-aware adaptive thresholding strategy inspired by DocPruner (Yan et al., 2025). This method postulates that the information density varies across documents, and thus the number of retained tokens should be dynamic.

For a document d, we compute the mean μ_d and standard deviation σ_d of its patch importance scores S_{eos}. A patch j is retained if its score exceeds a statistical threshold:

$$S_{eos}(j) > \mu_d + k \cdot \sigma_d \tag{13}$$

where k is an adaptation factor controlling the aggressiveness of pruning.

Fairness Calibration. To ensure a rigorous comparison with fixed-ratio methods (like SAP) at a specific target retention ratio γ (e.g., 10%), we do not arbitrarily select k. Instead, we employ a calibration process. We extract the EOS attention scores from a held-out calibration set of 128 randomly sampled documents. We convert these scores into Z-scores $z_{ij} = (S_{eos}(j) - \mu_i)/\sigma_i$ and compute the global empirical quantile:

$$k = \text{Quantile}(\{z_{ij}\}_{\text{calib}}, 1 - \gamma) \tag{14}$$

This ensures that the global average retention rate of this adaptive baseline strictly aligns with the target γ, isolating the impact of the selection strategy (Adaptive vs. Fixed) from the index vector size budget.

D Dataset Details

Table 5 details the composition of the ViDoRe v1 and v2 benchmarks used in our comprehensive evaluation.

Benchmark	Subset Name	Primary Domain
ViDoRe v1	ArxivQA	Academic / STEM
	DocVQA	General Document
	InfoVQA	Infographics / Layout
	Shift Project	Environmental Reports
	Artificial Intelligence	Technical Reports
	Energy	Industry Reports
	Government Reports	Policy / Legal
	Healthcare Industry	Medical / Business
ViDoRe v2	MIT Biomedical (Multi)	Medical / Research
	Economics Macro (Multi)	Finance / Policy
	ESG Restaurant (Multi)	Business / Tables
	ESG Restaurant (Human)	Business / Tables

Table 5: **Dataset Specifications.** Overview of domains covered in the ViDoRe benchmark suite.

E Detailed Evaluation Results

The following tables present the detailed performance breakdown for ColPali (Table 6), ColQwen (Table 7), and Jina Embeddings v4 (Table 8) on the ViDoRe v1 benchmark, and Table 9 on the ViDoRe v2 benchmark.

Dataset	Method	Upper Bound	$\gamma = 0.20$			$\gamma = 0.10$			$\gamma = 0.05$		
		Full NDCG	S.Ret	NDCG	%	S.Ret	NDCG	%	S.Ret	NDCG	%
ArxivQA	EOS-Adaptive	0.82	0.90	0.77	93.98	0.84	0.72	88.50	0.76	0.65	80.06
	Random		0.93	0.80	98.20	0.89	0.78	95.94	0.83	0.76	92.53
	Cluster		0.92	0.81	99.08	0.87	**0.80**	**97.63**	0.80	**0.77**	**94.68**
	SAP-Mean		0.95	0.82	99.92	0.91	0.80	97.36	**0.86**	0.77	94.01
	SAP-Max		**0.95**	**0.82**	**99.95**	**0.91**	0.80	97.52	0.85	0.76	93.35
DocVQA	EOS-Adaptive	0.58	0.91	0.50	86.99	0.86	0.43	73.34	0.81	0.34	58.79
	Random		0.94	0.53	91.60	0.90	0.49	84.87	0.84	0.44	76.25
	Cluster		0.92	0.56	96.02	0.86	0.51	87.27	0.77	0.44	75.71
	SAP-Mean		**0.97**	**0.56**	**96.65**	**0.94**	**0.52**	**90.27**	**0.89**	**0.48**	**82.03**
	SAP-Max		0.97	0.56	96.41	0.93	0.52	90.08	0.89	0.47	81.04
InfoVQA	EOS-Adaptive	0.85	0.86	0.77	90.95	0.79	0.73	85.37	0.73	0.68	80.15
	Random		0.91	0.82	96.36	0.86	0.79	92.76	0.79	0.76	89.14
	Cluster		0.90	0.84	98.19	0.82	0.81	94.61	0.71	0.72	85.01
	SAP-Mean		0.94	0.84	98.50	0.88	0.80	93.78	**0.82**	**0.76**	**89.31**
	SAP-Max		**0.94**	**0.84**	**99.07**	**0.88**	**0.81**	**94.81**	0.82	0.76	89.07
Shift Project	EOS-Adaptive	0.78	0.82	0.61	78.03	0.74	0.44	57.18	0.65	0.36	46.85
	Random		0.90	0.66	84.67	0.83	0.59	76.49	0.76	0.52	66.92
	Cluster		0.85	0.67	86.59	0.74	0.52	67.30	0.61	0.35	45.39
	SAP-Mean		0.93	0.71	90.99	0.87	0.62	79.31	0.79	0.52	66.64
	SAP-Max		**0.93**	**0.74**	**94.91**	**0.87**	**0.64**	**82.24**	**0.79**	**0.54**	**69.71**
Artificial Intel.	EOS-Adaptive	0.97	0.87	0.94	96.40	0.81	0.88	90.58	0.73	0.81	83.26
	Random		0.90	0.95	97.83	0.84	0.92	94.36	0.77	**0.88**	**90.46**
	Cluster		0.88	**0.97**	**99.63**	0.79	**0.93**	**96.23**	0.66	0.81	82.94
	SAP-Mean		0.94	0.96	98.39	0.89	0.91	93.91	0.82	0.86	89.03
	SAP-Max		**0.95**	0.96	98.97	**0.89**	0.93	95.69	**0.83**	0.88	90.42
Energy	EOS-Adaptive	0.95	0.83	0.82	86.16	0.75	0.77	81.26	0.68	0.72	76.15
	Random		0.90	0.92	96.98	0.84	0.88	92.94	0.76	0.81	85.12
	Cluster		0.88	**0.94**	**99.08**	0.80	**0.92**	**97.17**	0.67	0.82	86.89
	SAP-Mean		**0.95**	0.93	98.40	0.88	0.89	94.17	**0.81**	**0.87**	**91.42**
	SAP-Max		0.94	0.93	98.29	**0.88**	0.91	95.34	0.80	0.86	90.30
Gov. Reports	EOS-Adaptive	0.96	0.86	0.90	93.69	0.78	0.85	87.81	0.70	0.74	76.43
	Random		0.90	0.92	95.95	0.84	0.88	91.63	0.76	0.83	85.73
	Cluster		0.88	0.95	98.34	0.78	0.92	95.33	0.65	0.78	80.55
	SAP-Mean		0.95	0.94	98.02	0.90	0.95	98.52	0.83	**0.90**	**93.70**
	SAP-Max		**0.95**	**0.96**	**99.37**	**0.90**	**0.95**	**98.57**	**0.84**	0.89	92.79
Healthcare	EOS-Adaptive	0.97	0.86	0.91	93.91	0.80	0.88	90.83	0.73	0.81	83.59
	Random		0.91	0.94	96.89	0.84	0.91	93.86	0.78	0.87	90.38
	Cluster		0.88	0.94	97.66	0.78	0.91	93.84	0.64	0.70	72.10
	SAP-Mean		**0.95**	**0.95**	**98.78**	0.91	**0.95**	**97.89**	**0.86**	0.94	96.97
	SAP-Max		0.95	0.95	98.65	**0.91**	0.93	96.29	0.85	**0.94**	**97.31**
TabFQuad	EOS-Adaptive	0.87	0.82	0.81	93.56	0.74	0.77	88.04	0.66	0.68	77.91
	Random		0.91	0.85	97.85	0.85	0.83	95.23	0.78	0.81	92.83
	Cluster		0.91	0.87	100.45	0.85	**0.86**	**99.04**	0.78	**0.85**	**98.23**
	SAP-Mean		0.94	0.87	100.16	0.88	0.85	97.47	0.79	0.80	92.00
	SAP-Max		**0.94**	**0.88**	**100.71**	**0.89**	0.85	97.85	**0.80**	0.82	94.26
TAT-DQA	EOS-Adaptive	0.72	0.85	0.55	76.65	0.78	0.45	63.08	0.72	0.36	50.07
	Random		0.89	0.62	86.33	0.81	0.54	75.50	0.73	0.45	62.85
	Cluster		0.86	0.66	92.44	0.76	0.59	82.11	0.64	0.44	61.96
	SAP-Mean		**0.93**	**0.68**	**94.44**	**0.88**	**0.63**	**88.16**	**0.82**	**0.57**	**79.60**
	SAP-Max		0.93	0.67	94.17	0.87	0.62	86.84	0.82	0.56	77.67

Table 6: **Detailed Performance: ColPali on ViDoRe v1.** Detailed metrics across all 10 datasets. The strongest method in each column is **bolded**.

Dataset	Method	Upper Bound	$\gamma = 0.20$			$\gamma = 0.10$			$\gamma = 0.05$		
		Full NDCG	S.Ret	NDCG	%	S.Ret	NDCG	%	S.Ret	NDCG	%
ArxivQA	EOS-Adaptive	0.86	0.84	0.74	85.69	0.75	0.67	78.10	0.64	0.55	64.13
	Random		0.91	0.83	95.93	0.84	0.80	92.94	0.76	0.75	87.17
	Cluster		0.87	**0.83**	**96.48**	0.79	**0.83**	**95.54**	0.68	**0.79**	**91.24**
	SAP-Mean		**0.92**	0.82	95.06	0.86	0.79	91.37	0.77	0.73	83.93
	SAP-Max		0.92	0.82	94.71	**0.86**	0.79	91.54	**0.77**	0.73	84.55
DocVQA	EOS-Adaptive	0.59	0.87	0.49	83.85	0.79	0.39	66.94	0.71	0.29	48.66
	Random		0.86	0.51	87.26	0.77	0.44	75.49	0.66	0.37	62.53
	Cluster		0.82	0.56	94.29	0.71	0.51	86.52	0.59	**0.46**	**77.71**
	SAP-Mean		0.93	0.56	94.70	0.87	0.51	86.79	**0.79**	0.44	74.29
	SAP-Max		**0.93**	**0.57**	**96.08**	**0.87**	**0.51**	**87.44**	0.79	0.43	73.46
InfoVQA	EOS-Adaptive	0.91	0.83	0.82	89.67	0.74	0.77	84.81	0.65	0.71	77.74
	Random		0.86	0.85	93.58	0.78	0.80	88.28	0.69	0.74	81.63
	Cluster		0.81	**0.87**	**95.24**	0.69	0.81	89.03	0.57	0.72	78.74
	SAP-Mean		**0.90**	0.86	94.19	**0.83**	**0.82**	**90.38**	**0.75**	**0.77**	**84.75**
	SAP-Max		0.90	0.85	93.87	0.82	0.81	88.73	0.74	0.76	83.37
Shift Project	EOS-Adaptive	0.85	0.83	0.67	78.47	0.75	0.58	67.85	0.67	0.51	60.11
	Random		0.84	0.74	87.33	0.76	0.64	75.65	0.66	0.52	61.47
	Cluster		0.79	0.75	87.68	0.66	0.65	76.06	0.53	0.50	58.64
	SAP-Mean		**0.91**	**0.85**	**99.41**	**0.83**	**0.78**	**91.92**	**0.73**	**0.64**	**75.39**
	SAP-Max		0.90	0.83	97.68	0.82	0.75	88.55	0.71	0.60	70.02
Artificial Intel.	EOS-Adaptive	0.98	0.81	0.85	86.68	0.74	0.82	83.13	0.65	0.79	80.08
	Random		0.86	0.96	97.36	0.78	0.92	93.62	0.68	0.84	85.45
	Cluster		0.81	**0.98**	**99.10**	0.70	**0.94**	**95.70**	0.59	**0.88**	**89.31**
	SAP-Mean		0.91	0.96	97.44	**0.83**	0.93	94.30	0.74	0.84	85.11
	SAP-Max		**0.91**	0.97	98.16	0.82	0.91	92.08	**0.74**	0.87	87.91
Energy	EOS-Adaptive	0.96	0.82	0.82	85.32	0.71	0.70	72.86	0.60	0.65	68.10
	Random		0.86	0.89	93.07	0.78	0.85	89.06	0.68	0.77	80.66
	Cluster		0.82	**0.93**	**97.00**	0.70	**0.90**	**94.38**	0.58	0.83	86.49
	SAP-Mean		**0.92**	0.91	94.87	**0.85**	0.85	88.92	0.75	0.82	86.05
	SAP-Max		0.91	0.90	94.62	0.84	0.86	90.25	**0.75**	**0.84**	**87.51**
Gov. Reports	EOS-Adaptive	0.95	0.84	0.92	97.15	0.74	0.86	90.99	0.64	0.77	81.31
	Random		0.87	0.93	98.01	0.78	0.90	94.88	0.69	0.83	87.38
	Cluster		0.81	0.91	96.23	0.69	0.87	92.41	0.57	0.83	87.74
	SAP-Mean		**0.92**	**0.94**	**98.91**	**0.84**	**0.92**	**97.02**	**0.76**	**0.87**	**92.49**
	SAP-Max		0.91	0.93	98.54	0.84	0.91	96.75	0.74	0.87	91.75
Healthcare	EOS-Adaptive	0.98	0.84	0.94	96.25	0.76	0.91	92.64	0.67	0.83	85.26
	Random		0.87	0.96	98.17	0.79	0.93	94.66	0.69	0.87	88.40
	Cluster		0.81	**0.98**	**100.33**	0.69	0.93	94.66	0.57	0.85	87.06
	SAP-Mean		**0.92**	0.97	99.37	**0.86**	**0.96**	**98.09**	**0.77**	**0.94**	**95.91**
	SAP-Max		0.92	0.97	99.16	0.85	0.96	97.98	0.77	0.93	95.12
TabFQuad	EOS-Adaptive	0.88	0.87	0.82	92.97	0.78	0.76	85.79	0.68	0.70	79.32
	Random		0.87	0.86	97.26	0.79	0.82	93.05	0.70	0.77	87.07
	Cluster		0.85	0.86	97.80	0.75	0.85	96.60	0.64	0.83	93.81
	SAP-Mean		**0.92**	**0.87**	**99.03**	**0.86**	0.84	95.82	**0.78**	0.80	91.14
	SAP-Max		0.92	0.87	98.44	0.85	**0.85**	**96.69**	0.77	0.80	90.94
TAT-DQA	EOS-Adaptive	0.81	0.83	0.64	78.35	0.72	0.54	66.68	0.61	0.45	55.84
	Random		0.82	0.70	86.41	0.72	0.60	73.66	0.61	0.48	59.17
	Cluster		0.79	**0.74**	**91.31**	0.67	**0.66**	**81.15**	0.54	**0.56**	**69.15**
	SAP-Mean		**0.90**	0.74	90.61	**0.80**	0.63	77.84	**0.67**	0.48	59.50
	SAP-Max		0.89	0.73	90.13	0.79	0.61	75.45	0.66	0.47	57.99

Table 7: **Detailed Performance: ColQwen on ViDoRe v1.** Detailed metrics across all 10 datasets. The strongest method in each column is **bolded**.

Dataset	Method	Upper Bound	$\gamma = 0.20$			$\gamma = 0.10$			$\gamma = 0.05$		
		Full NDCG	S.Ret	NDCG	%	S.Ret	NDCG	%	S.Ret	NDCG	%
ArxivQA	EOS-Adaptive	0.88	0.90	0.84	94.94	0.82	0.76	85.73	0.72	0.62	70.50
	Random		0.92	0.85	96.72	0.86	0.82	93.02	0.79	0.77	86.97
	Cluster		0.88	**0.87**	**98.80**	0.81	**0.85**	**95.83**	0.72	**0.82**	**92.86**
	SAP-Mean		0.93	0.85	96.12	0.87	0.81	91.69	**0.80**	0.75	84.56
	SAP-Max		**0.93**	0.86	97.52	**0.87**	0.81	91.42	0.79	0.71	80.10
DocVQA	EOS-Adaptive	0.61	0.87	0.56	90.81	0.80	0.46	74.90	0.70	0.34	55.38
	Random		0.87	0.54	87.91	0.79	0.49	79.23	0.70	0.40	65.91
	Cluster		0.84	0.56	92.06	0.73	0.49	80.17	0.60	0.43	69.62
	SAP-Mean		**0.93**	0.59	96.43	**0.88**	**0.56**	**91.06**	**0.80**	**0.49**	**79.21**
	SAP-Max		0.93	**0.60**	**97.62**	0.87	0.55	90.44	0.78	0.46	74.28
InfoVQA	EOS-Adaptive	0.92	0.84	0.84	91.15	0.75	0.75	81.73	0.64	0.59	64.34
	Random		0.87	0.88	95.92	0.80	**0.84**	**90.89**	0.72	**0.77**	**83.62**
	Cluster		0.83	**0.89**	**96.39**	0.71	0.83	90.20	0.59	0.74	80.44
	SAP-Mean		**0.90**	0.88	95.56	**0.83**	0.83	90.72	**0.75**	0.76	82.18
	SAP-Max		0.90	0.88	95.43	0.83	0.83	90.66	0.73	0.74	80.27
Shift Project	EOS-Adaptive	0.89	0.85	0.87	96.77	0.78	0.77	85.72	0.72	0.67	75.24
	Random		0.88	0.84	93.46	0.81	0.76	84.96	0.72	0.65	72.33
	Cluster		0.81	0.82	91.32	0.70	0.73	81.37	0.59	0.58	64.90
	SAP-Mean		0.92	0.90	100.73	**0.86**	**0.86**	**95.92**	**0.78**	**0.79**	**87.77**
	SAP-Max		**0.92**	**0.91**	**101.70**	0.85	0.84	94.08	0.77	0.75	84.32
Artificial Intel.	EOS-Adaptive	0.99	0.84	0.97	98.03	0.77	0.88	88.94	0.69	0.76	76.94
	Random		0.89	0.97	98.21	0.82	0.94	94.61	0.73	0.88	88.59
	Cluster		0.81	0.97	97.90	0.71	0.92	93.11	0.60	0.90	90.43
	SAP-Mean		**0.90**	**0.98**	**98.39**	**0.84**	0.94	95.18	**0.77**	0.90	90.60
	SAP-Max		0.90	0.97	98.19	0.84	**0.95**	**96.16**	0.76	**0.91**	**91.31**
Energy	EOS-Adaptive	0.96	0.85	0.90	93.77	0.78	0.83	86.57	0.69	0.72	75.31
	Random		0.89	0.94	98.05	0.82	0.90	93.67	0.74	0.83	86.51
	Cluster		0.81	**0.95**	**98.59**	0.71	**0.91**	**94.28**	0.60	**0.84**	**87.17**
	SAP-Mean		**0.90**	0.93	97.31	**0.84**	0.87	90.62	**0.77**	0.80	83.04
	SAP-Max		0.90	0.91	94.22	0.84	0.85	88.49	0.76	0.78	81.04
Gov. Reports	EOS-Adaptive	0.97	0.85	0.94	97.11	0.78	0.90	92.61	0.70	0.81	83.35
	Random		0.89	0.95	97.97	0.82	**0.92**	**94.48**	0.74	0.85	87.56
	Cluster		0.82	**0.95**	**98.09**	0.71	0.92	94.41	0.60	0.87	89.15
	SAP-Mean		**0.91**	0.94	97.33	**0.85**	0.92	94.29	**0.78**	0.87	89.44
	SAP-Max		0.91	0.95	97.56	0.84	0.91	93.89	0.77	**0.87**	**89.68**
Healthcare	EOS-Adaptive	0.98	0.85	**0.97**	**99.38**	0.78	0.91	92.88	0.71	0.84	85.97
	Random		0.89	0.97	98.78	0.82	0.95	96.39	0.74	0.90	91.53
	Cluster		0.81	0.97	98.64	0.70	0.90	92.10	0.60	0.87	88.77
	SAP-Mean		0.91	0.97	99.12	**0.85**	**0.96**	**97.63**	**0.78**	**0.92**	**93.65**
	SAP-Max		**0.91**	0.97	98.99	0.85	0.95	97.12	0.77	0.89	90.93
TabFQuad	EOS-Adaptive	0.96	0.89	0.92	96.47	0.82	0.88	92.56	0.72	0.78	81.92
	Random		0.90	0.94	98.11	0.84	0.92	96.01	**0.76**	0.88	92.09
	Cluster		0.85	0.94	98.16	0.77	**0.93**	**97.50**	0.67	**0.90**	**94.70**
	SAP-Mean		0.91	**0.94**	**98.17**	0.84	0.90	94.15	0.75	0.84	88.07
	SAP-Max		**0.91**	0.93	97.31	**0.85**	0.89	93.49	0.74	0.83	87.17
TAT-DQA	EOS-Adaptive	0.79	0.89	0.68	86.50	0.81	0.57	72.21	0.71	0.40	50.38
	Random		0.88	0.70	88.11	0.81	0.62	78.77	0.72	0.53	66.82
	Cluster		0.81	0.71	90.13	0.71	**0.67**	**85.06**	0.61	**0.58**	**73.93**
	SAP-Mean		**0.92**	0.71	90.18	**0.86**	0.63	79.78	**0.77**	0.51	64.36
	SAP-Max		0.92	**0.72**	**90.71**	0.85	0.62	78.51	0.75	0.46	58.33

Table 8: **Detailed Performance: Jina Embeddings v4 on ViDoRe v1.** Detailed metrics across all 10 datasets. The strongest method in each column is **bolded**.

Dataset	Method	Upper Bound	$\gamma = 0.20$			$\gamma = 0.10$			$\gamma = 0.05$		
		Full NDCG	S.Ret	NDCG	%	S.Ret	NDCG	%	S.Ret	NDCG	%
Model: ColPali											
MIT Biomedical	EOS-Adaptive	0.57	0.87	0.48	83.58	0.80	0.42	72.62	0.73	0.36	63.28
	Random		0.93	0.55	95.98	0.89	0.52	90.80	0.82	0.48	83.37
	Cluster		0.92	0.55	95.54	0.86	0.51	89.29	0.78	0.46	79.61
	SAP-Mean		**0.94**	**0.55**	**96.30**	**0.89**	**0.52**	**91.31**	**0.83**	**0.49**	**85.99**
	SAP-Max		0.94	0.55	96.29	0.89	0.52	90.92	0.83	0.49	85.34
Econ. Macro	EOS-Adaptive	0.51	0.83	0.43	83.66	0.76	0.40	78.39	0.69	0.36	69.73
	Random		0.88	0.45	87.83	0.81	0.41	79.75	0.73	0.39	75.75
	Cluster		0.87	0.44	85.07	0.76	0.36	70.91	0.64	0.27	52.47
	SAP-Mean		0.92	**0.48**	**94.41**	**0.87**	**0.45**	**87.33**	0.80	**0.43**	**83.32**
	SAP-Max		**0.93**	0.46	90.36	0.87	0.45	87.13	**0.81**	0.42	82.67
ESG Rest. (Multi)	EOS-Adaptive	0.56	0.87	0.42	76.09	0.80	0.37	65.79	0.71	0.28	51.04
	Random		0.90	0.47	84.43	0.83	0.42	74.43	0.75	0.35	63.27
	Cluster		0.86	0.42	76.02	0.76	0.30	54.10	0.62	0.17	30.79
	SAP-Mean		0.93	**0.49**	**87.03**	0.88	0.45	80.53	0.81	0.38	67.71
	SAP-Max		**0.94**	0.49	86.97	**0.89**	**0.47**	**84.95**	**0.82**	**0.39**	**69.81**
ESG Rest. (Human)	EOS-Adaptive	0.60	0.87	0.43	71.07	0.79	0.35	59.04	0.69	0.26	43.70
	Random		0.91	0.47	79.33	0.84	0.41	68.22	0.76	0.33	55.37
	Cluster		0.86	0.47	79.36	0.76	0.38	63.35	0.64	0.27	45.29
	SAP-Mean		**0.94**	**0.55**	**92.53**	0.88	**0.51**	**85.43**	0.81	**0.46**	**76.28**
	SAP-Max		0.94	0.54	90.74	**0.89**	0.49	81.54	**0.82**	0.45	75.20
Model: ColQwen2											
MIT Biomedical	EOS-Adaptive	0.54	0.82	0.45	83.45	0.75	0.41	75.32	0.67	0.36	66.86
	Random		0.90	0.51	94.81	0.83	0.48	89.18	0.76	0.45	83.03
	Cluster		0.85	0.51	94.23	0.75	0.48	87.90	0.64	0.43	79.20
	SAP-Mean		0.91	0.51	94.87	**0.86**	**0.50**	**91.54**	**0.78**	**0.47**	**87.72**
	SAP-Max		**0.91**	**0.51**	**95.06**	0.85	0.48	89.68	0.78	0.47	86.92
Econ. Macro	EOS-Adaptive	0.48	0.80	0.43	89.22	0.71	0.39	80.70	0.62	0.36	74.41
	Random		0.85	0.42	86.70	0.77	0.37	76.50	0.68	0.34	69.95
	Cluster		0.79	0.41	85.31	0.67	0.34	71.11	0.55	0.27	57.17
	SAP-Mean		**0.88**	**0.46**	**97.00**	0.80	0.42	88.51	**0.72**	**0.43**	**89.62**
	SAP-Max		0.88	0.45	93.56	**0.80**	**0.43**	**89.32**	0.72	0.41	86.46
ESG Rest. (Multi)	EOS-Adaptive	0.57	**0.89**	**0.57**	**100.63**	0.81	**0.49**	**85.20**	0.70	0.36	62.74
	Random		0.86	0.46	80.19	0.78	0.39	68.26	0.67	0.30	52.78
	Cluster		0.79	0.45	79.06	0.68	0.37	64.06	0.58	0.27	48.12
	SAP-Mean		0.88	0.50	88.21	0.81	0.46	80.60	0.72	0.38	65.79
	SAP-Max		0.89	0.53	93.59	**0.82**	0.46	79.86	**0.74**	**0.39**	**67.70**
ESG Rest. (Human)	EOS-Adaptive	0.57	0.84	0.43	75.76	0.74	0.37	64.78	0.64	0.26	45.38
	Random		0.85	**0.46**	80.76	0.76	**0.40**	**69.90**	0.66	**0.34**	**58.86**
	Cluster		0.79	0.44	77.62	0.68	0.37	65.56	0.57	0.29	49.87
	SAP-Mean		0.89	0.43	74.38	0.81	0.37	65.11	0.73	0.33	57.60
	SAP-Max		**0.90**	0.46	**80.94**	**0.82**	0.38	65.67	**0.73**	0.33	58.20
Model: Jina Embeddings v4											
MIT Biomedical	EOS-Adaptive	0.61	0.90	0.58	94.49	0.81	0.51	83.41	0.70	0.43	70.35
	Random		0.91	0.58	95.83	0.86	0.55	90.55	0.79	**0.51**	**84.40**
	Cluster		0.86	0.57	94.12	0.79	0.55	89.86	0.70	0.48	78.97
	SAP-Mean		0.92	0.58	95.97	0.87	0.56	92.36	0.80	0.51	83.77
	SAP-Max		**0.92**	**0.59**	**96.67**	**0.87**	**0.57**	**93.04**	0.79	0.49	80.50
Econ. Macro	EOS-Adaptive	0.55	0.86	0.51	93.57	0.80	0.50	90.14	0.71	0.42	77.19
	Random		0.88	0.48	87.39	0.81	0.43	77.96	0.73	0.38	68.70
	Cluster		0.81	0.43	77.96	0.70	0.35	62.81	0.60	0.30	54.01
	SAP-Mean		0.90	0.55	99.44	**0.84**	**0.52**	**94.34**	**0.77**	0.44	80.50
	SAP-Max		**0.90**	**0.56**	**102.73**	0.83	0.51	91.97	0.76	**0.45**	**81.42**
ESG Rest. (Multi)	EOS-Adaptive	0.53	0.90	0.44	84.04	0.84	0.40	75.34	0.75	0.35	66.78
	Random		0.89	0.46	86.05	0.82	0.40	75.20	0.74	0.33	63.05
	Cluster		0.82	0.41	78.30	0.72	0.34	63.48	0.63	0.27	51.72
	SAP-Mean		0.91	0.51	95.81	0.85	0.47	89.65	**0.78**	**0.39**	**73.79**
	SAP-Max		**0.91**	**0.52**	**97.93**	**0.85**	**0.48**	**90.69**	0.77	0.34	64.66
ESG Rest. (Human)	EOS-Adaptive	0.64	**0.92**	**0.58**	**91.55**	0.85	0.49	77.01	0.75	0.33	51.34
	Random		0.89	0.54	84.56	0.83	0.45	70.26	0.74	0.35	55.18
	Cluster		0.81	0.46	72.86	0.70	0.35	55.78	0.60	0.28	43.57
	SAP-Mean		0.92	0.58	91.16	**0.86**	0.49	77.62	**0.80**	**0.45**	**70.52**
	SAP-Max		0.92	0.58	91.39	0.86	**0.52**	**81.28**	0.79	0.42	66.75

Table 9: **Detailed Performance on ViDoRe v2.** SAP consistently outperforms baselines across most datasets and models. The best performance in each column is **bolded**.

F Computational Complexity & Efficiency Analysis

A primary concern for any indexing strategy is the additional latency introduced during the document processing phase. In this section, we formally analyze the computational overhead of Structural Anchor Pruning (SAP) and provide empirical benchmarks on the ViDoRe v2 dataset.

F.1 Theoretical Complexity

Let N be the number of visual patches (e.g., 1024 for standard inputs), L the number of transformer layers, and d the hidden dimension.

Backbone Cost. The computational cost of the standard forward pass is dominated by the self-attention mechanism and feed-forward networks. The complexity for the attention mechanism alone across all layers is $\mathcal{O}(L \cdot N^2 \cdot d)$.

SAP Overhead. SAP operates by extracting attention matrices $A^{(l,h)}$ from a subset of layers $\mathcal{L}^*$. The operations required are:

1. **Extraction:** Accessing attention logits (effectively zero FLOPs, bounded by memory bandwidth).
2. **Aggregation (In-Degree):** Summing columns of the attention matrix. For a selected layer set $|\mathcal{L}^*|$ and heads H, the complexity is $C_{SAP} = \mathcal{O}(|\mathcal{L}^*| \cdot H \cdot N^2)$.

Comparing the two, the ratio of SAP overhead to the attention computation is approximately:

$$\frac{C_{SAP}}{C_{Attn}} \approx \frac{|\mathcal{L}^*| \cdot H \cdot N^2}{L \cdot H \cdot N^2 \cdot d} = \frac{|\mathcal{L}^*|}{L \cdot d} \quad (15)$$

For the `jina-embeddings-v4` model used in our experiments, with $d = 1280$, this ratio is exceedingly small ($< 10^{-3}$), implying the theoretical cost is negligible.

F.2 Empirical Benchmarks

To validate our theoretical analysis, we conducted a rigorous latency benchmark using the **ViDoRe v2** dataset (subsets: *esg_reports, biomedical_lectures, economics_reports*). Experiments were performed on a single **NVIDIA H200 (141GB)** GPU. The backbone model is `jina-embeddings-v4` (hidden size $d = 1280$).

We measure the **Pruning Latency** (time taken to compute masks and select tokens) and compare it against the **Full Forward Pass** time. The results are summarized in Table 10.

Method	Avg Time (ms)	Overhead (+ %)
Full Forward Pass	206.05	*(Baseline)*
Random	0.04	+0.02%
Adaptive-EOS	0.08	+0.04%
SAP-Max (Ours)	**0.05**	**+0.03%**
SAP-Mean (Ours)	**0.06**	**+0.03%**
Cluster-Merge	12.08	+5.86%

Table 10: **Efficiency Benchmark on ViDoRe v2.** Latency represents the time per page for the pruning operation (mask generation) only. **Overhead** is calculated relative to the Full Forward Pass time (206.05 ms). SAP variants introduce negligible overhead ($< 0.03\%$), whereas clustering-based methods incur a significant penalty ($\approx 6\%$).

Results Analysis. As shown in Table 10, the Full Forward Pass requires approximately 206ms per page.

- **Negligible Overhead:** SAP-Mean and SAP-Max require only 0.06ms and 0.05ms respectively. This corresponds to an overhead of approximately $\mathbf{0.03}\%$ relative to the model inference. In a real-world pipeline, this is imperceptible.
- **Comparison to Clustering:** Iterative methods like *Cluster-Merge* (K-Means) are significantly slower, taking $\approx$ 12ms per page. While feasible, this represents a $\sim 200\times$ slowdown compared to SAP and adds nearly 6% to the total indexing time.
- **Comparison to Random:** SAP achieves comparable speed to *Random* selection (0.04ms) while providing the semantic benefits detailed in Section 4.

These results confirm that SAP is a highly scalable solution suitable for high-throughput Visual RAG systems processing millions of documents.

G Detailed Comparison with Trained Methods

In Table 11, we provide a fine-grained breakdown of the comparison between SAP and Light-ColPali/Light-ColQwen2 (Faysse et al., 2024) across individual datasets.

Table 11: **Detailed Dataset Breakdown: SAP vs. Light-Baselines.** We compare the trained token merging baselines (Top) with our training-free SAP variants (Bottom) across 6 datasets. We report NDCG@5 and the percentage of the full model's performance retained (subscript).

Model	Method	Factor	InfoVQA	DocVQA	ArxivQA	TabFQuAD	TAT-DQA	ShiftProj	Average
Trained Baselines (Token Merging)									
Light ColPali	Merging	4×	$0.83_{98.1}$	$0.53_{97.4}$	$0.84_{98.8}$	$0.87_{101.4}$	$0.73_{100.7}$	$0.73_{96.0}$	$0.75_{98.7}$
	Merging	9×	$0.82_{97.3}$	$0.55_{100.0}$	$0.84_{98.1}$	$0.85_{99.1}$	$0.71_{98.1}$	$0.73_{96.4}$	$0.75_{98.2}$
	Merging	25×	$0.81_{96.2}$	$0.51_{92.2}$	$0.83_{97.1}$	$0.83_{97.0}$	$0.67_{92.9}$	$0.71_{93.6}$	$0.72_{94.8}$
Light ColQwen2	Merging	4×	$0.90_{97.8}$	$0.57_{102.2}$	$0.89_{100.7}$	$0.90_{99.7}$	$0.81_{99.3}$	$0.87_{98.4}$	$0.82_{99.7}$
	Merging	9×	$0.90_{98.0}$	$0.56_{101.3}$	$0.87_{98.5}$	$0.89_{98.1}$	$0.79_{97.5}$	$0.87_{98.6}$	$0.81_{98.8}$
	Merging	25×	$0.89_{97.2}$	$0.55_{98.6}$	$0.86_{98.2}$	$0.89_{98.7}$	$0.79_{97.0}$	$0.84_{95.4}$	$0.80_{97.5}$
Ours (Training-Free Structural Anchor Pruning)									
ColPali	SAP-Mean	4×	$0.84_{98.8}$	$0.57_{98.0}$	$0.82_{99.8}$	$0.87_{100.0}$	$0.69_{96.4}$	$0.75_{96.5}$	$0.76_{98.2}$
		9×	$0.81_{94.6}$	$0.53_{92.0}$	$0.80_{98.4}$	$0.85_{97.4}$	$0.64_{89.5}$	$0.66_{84.6}$	$0.71_{92.8}$
		25×	$0.75_{88.0}$	$0.46_{79.9}$	$0.74_{90.8}$	$0.79_{90.5}$	$0.54_{75.1}$	$0.50_{64.5}$	$0.63_{81.5}$
	SAP-Max	4×	$0.85_{99.2}$	$0.57_{97.8}$	$0.82_{99.7}$	$0.87_{100.3}$	$0.69_{96.2}$	$0.77_{98.7}$	$0.76_{98.7}$
		9×	$0.81_{95.2}$	$0.53_{91.5}$	$0.81_{98.5}$	$0.86_{98.8}$	$0.63_{87.7}$	$0.65_{83.4}$	$0.71_{92.5}$
		25×	$0.74_{87.2}$	$0.45_{76.8}$	$0.74_{91.0}$	$0.78_{89.8}$	$0.53_{73.5}$	$0.48_{61.3}$	$0.62_{79.9}$
ColQwen2	SAP-Mean	4×	$0.88_{96.4}$	$0.56_{95.1}$	$0.83_{96.2}$	$0.88_{100.1}$	$0.77_{94.8}$	$0.84_{99.1}$	$0.79_{96.9}$
		9×	$0.84_{92.6}$	$0.53_{90.1}$	$0.80_{92.5}$	$0.85_{96.8}$	$0.69_{85.0}$	$0.80_{94.1}$	$0.75_{91.9}$
		25×	$0.76_{83.9}$	$0.42_{72.1}$	$0.72_{83.0}$	$0.78_{88.7}$	$0.52_{63.7}$	$0.64_{75.7}$	$0.64_{77.9}$
	SAP-Max	4×	$0.88_{96.2}$	$0.56_{95.4}$	$0.83_{96.3}$	$0.88_{99.8}$	$0.77_{94.8}$	$0.86_{101.2}$	$0.80_{97.3}$
		9×	$0.83_{91.6}$	$0.53_{89.8}$	$0.79_{91.5}$	$0.85_{96.9}$	$0.71_{86.6}$	$0.80_{93.7}$	$0.75_{91.7}$
		25×	$0.76_{83.1}$	$0.44_{74.9}$	$0.73_{84.1}$	$0.78_{88.7}$	$0.52_{63.5}$	$0.67_{78.5}$	$0.65_{78.8}$