Structural Anchor Pruning: Training-Free Multi-Vector Compression for Visual Document Retrieval

Anonymous ACL submission

Abstract

Recent Vision-Language Models (e.g., ColPali) enable fine-grained Visual Document Retrieval (VDR) but incur prohibitive multi-vector index storage overhead. Existing training-free pruning methods either rely on heuristic layer choices or degrade sharply under aggressive compression, leading prior work to argue that effective high-compression pruning requires query-dependent training. We challenge this view with Structural Anchor Pruning (SAP), a self-calibrating, training-free, and query-agnostic index-time pruning framework with three components: (i) Score Retention (SR), a white-box per-layer compression diagnostic; (ii) SR-guided window selection, a procedure that automatically locates the structural pruning region for any backbone with no per-model hyperparameters; and (iii) a visual in-degree centrality scorer that identifies anchor patches within the selected window. On the ViDoRe v1/v2 benchmarks across three architectures spanning 18, 28, and 36 backbone layers, SAP retains over 90% of NDCG@5 while pruning more than 90% of visual tokens, without any per-model parameter tuning. Our layer-resolved SR analysis reveals an Alignment-Aggregation Divergence: the document's visual structure is preserved as a stable "Structural Plateau" within the backbone, but the final layers reshape this representation into a sparse, query-aligned form that is no longer suitable for pruning. This is the mechanistic reason SAP succeeds where final-layer methods fail.

1 Introduction

Visual Document Retrieval (VDR) has shifted from traditional pipelines to end-to-end Vision-Language Models (VLMs) (Zhang et al., 2024a). 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.

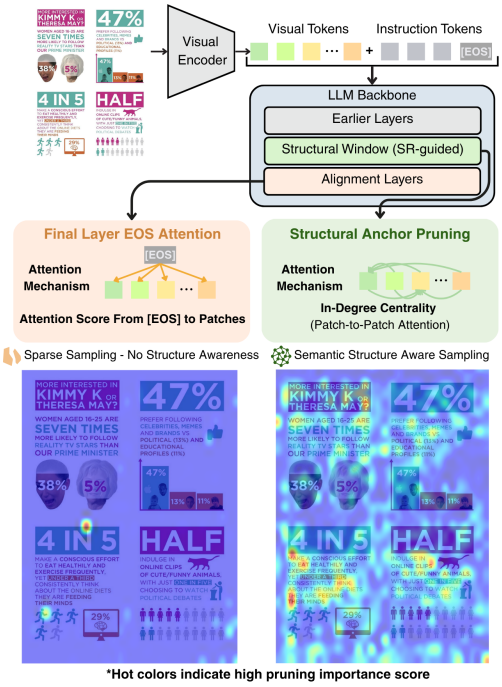

Figure 1: Comparison of Pruning Mechanisms. Final-layer EOS attention (left) yields sparse, structure-agnostic sampling. Our Structural Anchor Pruning (right) applies visual in-degree centrality on patch-to-patch attention within a structural window that is automatically located per architecture by our SR-guided selection procedure, yielding a structure-aware sample of anchor patches.

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 perfor-

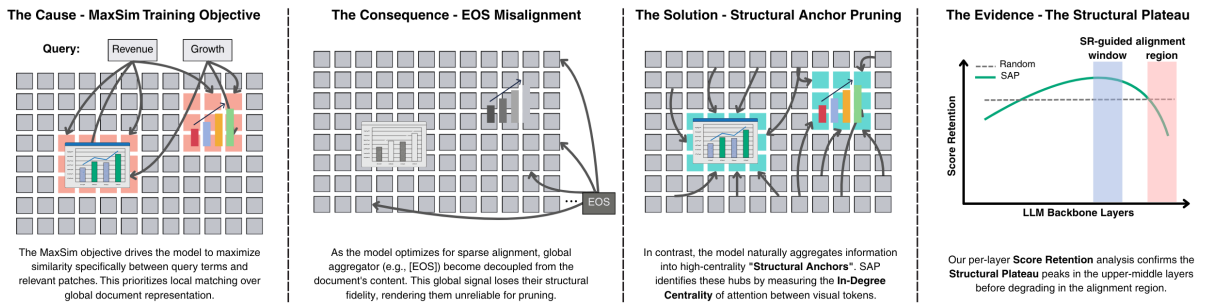

Figure 2: **The Alignment-Aggregation Divergence.** Final-layer global signals decay under MaxSim alignment, while a contiguous Structural Plateau within the backbone aggregates information into high-centrality anchor patches. SAP exploits this by measuring **visual in-degree centrality** between visual tokens; per-backbone Score Retention curves are reported in Figure 5.

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

In this work, we challenge this conclusion. At the core of our approach is **Score Retention (SR)**, a white-box, label-free per-layer compression diagnostic that quantifies per-pair MaxSim fidelity decoupled from corpus-level ranking effects. We use SR as the driving signal of an **SR-guided window selection** procedure that automatically locates the structural pruning region for any backbone without per-model tuning. Within the selected window, **visual in-degree centrality** identifies anchor patches in a query-agnostic manner. We call the resulting end-to-end, training-free, query-agnostic, index-time framework **Structural Anchor Pruning (SAP)** (Figure 1). On the ViDoRe v1/v2 benchmarks (Faysse et al., 2024; Macé et al., 2025), SAP consistently outperforms EOS-Adaptive (Yan et al., 2025), Random (Ma et al., 2025), and Semantic Clustering (Ma et al., 2025) across three backbones of widely different depth (18, 28, 36 layers), retaining over 90% of NDCG@5 while reducing stored vectors by more than 90%.

Beyond its operational role in window selection, SR also serves as a layer-resolved probe for analyzing VLM backbones. Applied across the full depth of three architectures, it reveals an **Alignment-Aggregation Divergence** (Figure 2): a contiguous block of layers preserves the document's visual structure as a stable Structural Plateau, while the final layers reshape this representation into a sparse, retrieval-aligned form that is no longer suitable for pruning. This explains why SR-guided selection consistently lands at the boundary between the two regions, and why final-layer pruning methods (e.g., EOS-attention) degrade sharply under aggressive compression.

2 Related Work and Preliminaries

2.1 VDR Multi-Vector Late Interaction

Unlike dense retrievers that map a document to a single vector, multi-vector retrievers such as ColPali (Faysse et al., 2024) employ the **Multi-Vector Late Interaction** mechanism (Khattab and Zaharia, 2020). A document image D is encoded into a bag of visual patch embeddings $E_D = \{v_1, \ldots, v_N\} \in \mathbb{R}^{N \times d}$ ($N \approx 1024$ patches per image). Given a text query Q tokenized as $\{q_1, \ldots, q_M\}$, the relevance score is computed via the MaxSim operator:

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

This preserves fine-grained layout details but requires indexing the full E_D, so storage scales linearly with N; on realistic corpora the index reaches terabytes (Xu et al., 2025).

2.2 Efficient Visual Document Retrieval

Two streams of work address this overhead: 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.

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).

Concurrent training-free work introduces a Prune-then-Merge framework (Yan et al., 2026), addressing the information loss of pure pruning with a two-stage pipeline: adaptive pruning is followed by hierarchical merging that aggregates the discarded patches into summary representations. SAP is procedurally simpler, a single-stage, fixed-budget per-document top-k selection driven by attention-graph centrality read from a backbone window automatically located by SR. The two designs target different stages of the indexing pipeline and could in principle be stacked.

Inference-time vs. Index-time Pruning. A separate line of work targets visual token pruning at inference time to accelerate single-pass VLM forward latency, such as Token Merging (Bolya et al., 2022) for vision transformers and Sparse-VLM (Zhang et al., 2024b) for vision-language models. These methods drop or merge tokens dynamically during the forward pass and discard them after inference. SAP addresses a different setting: persistent index-time compression that prunes once and reuses the result query-agnostically across all future queries, optimizing storage rather than per-pass latency. We therefore compare only against index-time baselines.

3 Methodology

We first introduce Structural Anchor Pruning, which scores tokens by visual in-degree centrality within a structural window of the LLM backbone. We then introduce Score Retention (SR), a white-box compression diagnostic that quantifies per-pair MaxSim fidelity independently of corpus-level ranking. Finally, we present SR-guided window selection, a label-free procedure that uses SR to automatically locate the structural window for any backbone, eliminating per-model tuning. The overall framework 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 the backbone's Structural Plateau act as information hubs, aggregating features from numerous other regions and constituting the core semantic representation of the document.

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 source 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 adopt averages centralities across heads and prioritizes anchors

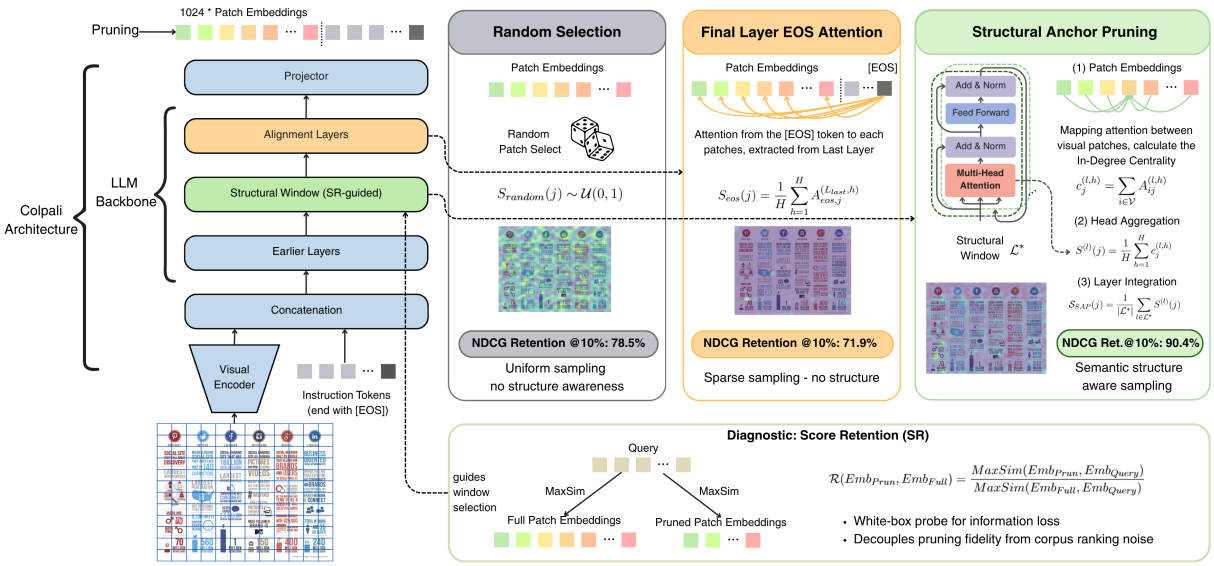

Figure 3: **Overview of SAP.** We compare three pruning paradigms on the ColPali architecture. **Left:** the shared Vision-Language backbone processes the image. **Middle:** conventional methods (Random Selection, Final Layer EOS Attention) fail to identify critical tokens, resulting in lower retention. **Right:** SAP identifies semantic structural anchor patches via **visual in-degree centrality** within the **Structural Window** of the backbone, automatically located by our SR-guided procedure. **Bottom:** the **Score Retention (SR)** diagnostic directly compares the MaxSim scores of pruned versus full embeddings, isolating intrinsic information loss from corpus-dependent ranking noise; SR itself drives the Structural Window selection. NDCG@5 retention numbers shown are ColPali averages on ViDoRe v2 at $\gamma = 0.10$ (Table 5).

consistently active across the attention subspace:

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

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 a two-phase structure within the backbone: an aggregation phase, where tokens actively exchange information to build a cohesive structural understanding of the document, followed by an alignment phase in the final layers, 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 boundaries $\alpha, \beta \in [0, 1]$:

$$\mathcal{L}^*(\alpha, \beta) = \{l \in \mathbb{N} \mid \lfloor \alpha \cdot L_{total} \rfloor \leq l < \lfloor \beta \cdot L_{total} \rfloor\} \tag{4}$$

where $0 \leq \alpha < \beta \leq 1$ define the boundaries of the structural window. We delegate the choice of (α, β) to a label-free, automatic procedure introduced in Section 3.3. The final importance score $\mathcal{S}_{SAP}(j)$ is obtained by averaging centrality scores across this window:

$$\mathcal{S}_{SAP}(j) = \frac{1}{|\mathcal{L}^*|}\sum_{l \in \mathcal{L}^*} S^{(l)}(j) \tag{5}$$

3.2 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:

$$\text{NDCG@}k = \frac{1}{\text{IDCG}_k}\sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2(i+1)} \tag{6}$$

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 **Score Retention (SR)** as a white-box compression diagnostic. We define fidelity not by ranking position, but by the preservation of the raw MaxSim score:

$$\mathcal{R}(\hat{E}_D, E_D) = \frac{\sum_{i=1}^{M} \max_{v \in \hat{E}_D}(q_i \cdot v)}{\sum_{i=1}^{M} \max_{v \in E_D}(q_i \cdot v)} \quad (7)$$

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.

SR is intentionally distinct from NDCG. Whereas NDCG measures corpus-level ranking quality relative to gold relevance judgments, SR measures per-pair score fidelity for individual (query, document) pairs, decoupled from any corpus-level ranking effect. The two are complementary, not interchangeable: a high SR does not guarantee correct ranking against distractors, and a low SR can still leave rankings unchanged. SR is purpose-built for one specific question: which layers preserve the most score fidelity under a fixed compression budget? We use it in two ways: (i) as the driving signal for our automatic window-selection procedure, and (ii) as a layer-resolved probe to localize where structural information concentrates within the backbone.

3.3 SR-Guided Window Selection

A naive instantiation of SAP would require selecting the structural window (α, β) heuristically. We propose a label-free, training-free procedure that uses SR itself to locate the window for any backbone, parameterized only by a calibration set size N and a relative window width $\rho \in (0, 1]$.

The procedure operates as follows. Given an unlabeled calibration set $\mathcal{C} = \{(I_n, Q_n)\}_{n=1}^{N}$ and a target retention ratio γ, for each layer $l \in \{0, \ldots, L_{total} - 1\}$ we compute, for every pair, the centrality $S^{(l)}$ from layer l alone (Eq. 3), retain the top-$\lceil \gamma |E_I| \rceil$ patches of the image embeddings, and record SR (Eq. 7) between the pruned and full embeddings. Averaging across calibration pairs yields a per-layer score retention curve $\mathcal{R}(l)$. We compute the median $m = \text{median}\{\mathcal{R}(l)\}_{l=0}^{L_{total}-1}$ and identify the alignment region as the longest suffix $[l^*, L_{total} - 1]$ for which $\mathcal{R}(l) < m$ throughout, i.e., the contiguous tail where retention has fallen below the global median. The structural window is placed immediately before this region, with width $k = \lceil \rho \cdot L_{total} \rceil$ layers:

$$\beta = \frac{l^*}{L_{total}}, \qquad \alpha = \frac{l^* - k}{L_{total}}. \quad (8)$$

Formal pseudocode is provided in Appendix D.

The procedure has three properties. It requires no labels, no training, and no NDCG evaluation. It is identical across backbones, introducing no per-architecture hyperparameter. And it directly operationalizes the Alignment-Aggregation Divergence (Section 5) by placing the window at the empirically detected boundary between aggregation and alignment phases.

The calibration queries are unlabeled and disjoint from the deployment query stream: they may come from any held-out generic query source and are used only once, offline. Once (α, β) has been selected, the window is applied query-agnostically at index time: SAP's centrality scorer never sees a query, neither at calibration nor at deployment.

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

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

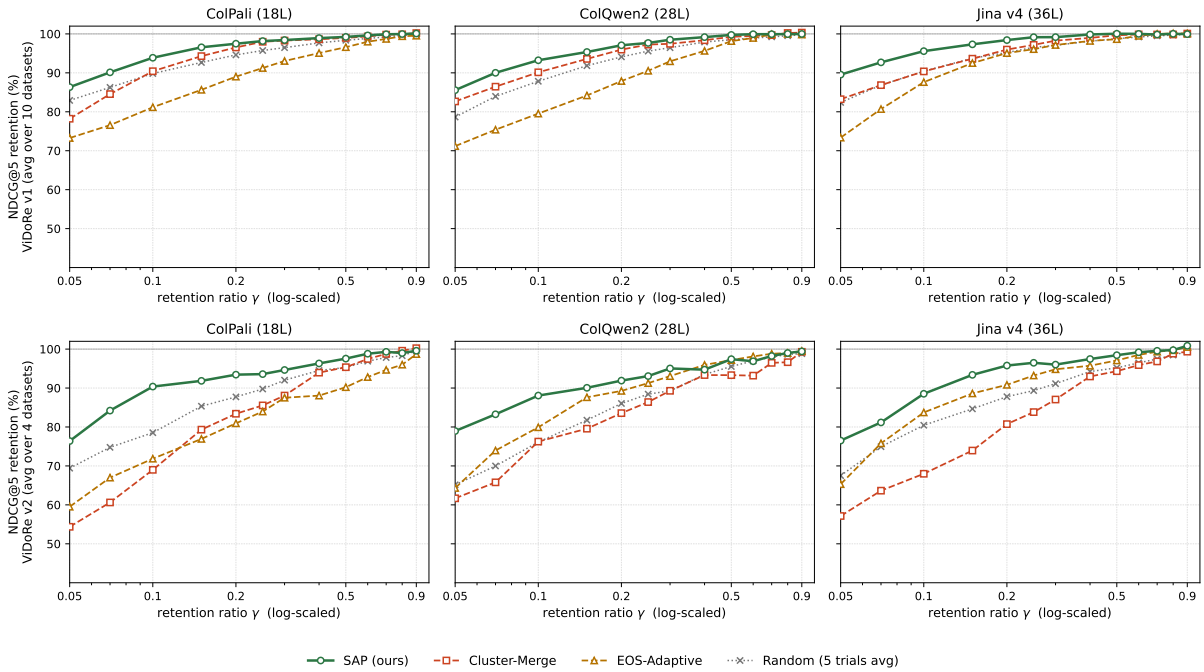

Figure 4: **Efficiency-Fidelity Trade-off.** NDCG@5 retention vs. retention ratio γ across ColPali, ColQwen2, and Jina v4 on ViDoRe v1 (top) and v2 (bottom). SAP (green) consistently dominates training-free baselines at every γ on every backbone-benchmark pair, with the largest gains under aggressive compression ($\gamma \leq 0.10$). SAP uses the SR-guided window.

embedding optimization recipes. See Appendix A for model architectural specifications.

We instantiate the layer ensemble $\mathcal{L}^*$ (Eq. 6) automatically using the SR-guided window selection procedure, with hyperparameters set to $\rho = 0.2$ (relative window width) and $N = 500$ (calibration set size). The calibration set consists of 500 (image, query) pairs uniformly sampled from the ColPali training corpus (Faysse et al., 2024), which is disjoint from all ViDoRe v1 and v2 evaluation splits. The resulting per-architecture windows are listed in Table 4. This calibration runs once per backbone, offline, and the pruned indices are reused for all future user queries with no further re-calibration.

We compare our SAP against three distinct training-free pruning paradigms (details in Appendix B): (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.

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 C.

4.2 Main Results

Figure 4 reports NDCG@5 retention as a function of the retention ratio γ for all three backbones on both ViDoRe v1 and v2. Three observations emerge across all six panels.

The SAP curve (green) matches or exceeds every training-free baseline at every $\gamma \in [0.05, 0.9]$ on every (backbone, benchmark) pair. At high retention ratios ($\gamma \geq 0.5$) all methods cluster near 95–100% of full-model NDCG@5 and the differences are small. The separation widens monotonically as compression intensifies, and by $\gamma = 0.05$ baselines fan out across a ~30-point band while SAP remains comfortably on top.

At the practical deployment regime $\gamma \in [0.05, 0.10]$, SAP retains >90% NDCG@5 on ViDoRe v1 and >76% on the harder v2, while Adaptive-EOS and Cluster-Merge fall to 54–74%. On v2 at $\gamma = 0.05$, the SAP-vs-EOS gap reaches 22–25 percentage points across backbones, closing

the gap that prior work (Ma et al., 2025; Yan et al., 2025) cited as the principal limitation of training-free pruning.

The SAP curve shape is qualitatively identical across backbones of widely different depth (18, 28, 36 layers), confirming that the SR-guided procedure tracks the optimal window per architecture without any per-model tuning. A three-γ case study with raw NDCG@5 values, Score Retention is provided in Appendix F. And per-cutoff (NDCG@$\{1, 5, 10, 50, 100\}$) retention is provided in Appendix H.

4.3 SR-Guided Window Validation

We validate the SR-guided window selection procedure (Section 3.3) by comparing it against a brute-force search over 9 sliding windows of fixed width $0.2\,L_{total}$, stepped by $0.1\,L_{total}$, covering relative depth ranges from 0–20% to 80–100%. Table 1 reports NDCG@5 retention at $\gamma = 0.10$ on ViDoRe v1 and v2 for each of the 9 windows on all three backbones. Full results for $\gamma = 0.20$ and $\gamma = 0.05$ are deferred to Appendix K.

The SR-guided procedure selects the 60–80% window for ColPali and ColQwen2, and the 70–90% window for Jina v4 (bold cells in the Mean rows of Table 1). On the mean of ViDoRe v1 and v2, this selection coincides exactly with the brute-force-best window for all three backbones. Per-bench, ColQwen2 matches the brute-force best on both v1 and v2; Jina v4 matches the best on v1 and lies 0.2 NDCG@5 points behind the 50–70% best on v2; ColPali matches the best on v2 and lies 0.3 points behind the 50–70% best on v1. The largest per-bench gap to the brute-force optimum across the three backbones is therefore 0.3 NDCG@5, achieved without any retrieval labels (Table 4, Appendix E).

The brute-force optimum lies at a different relative depth on each backbone: 50–70% for ColPali (18 layers), 60–80% for ColQwen2 (28 layers), and 70–90% for Jina v4 (36 layers). In absolute terms, the optimum always ends approximately 2–4 layers before the last layer, regardless of backbone depth. The SR-guided procedure recovers this shifted optimum on all three backbones without any per-model intervention; we examine the mechanism behind the shift in Section 5.

Holding the window position fixed by the drop-alignment rule and sweeping the width k from 1 to L_{total} layers shows NDCG@5 retention is insensitive to k over a broad plateau that includes our default $k = \lceil 0.2\,L_{total} \rceil$ for all three backbones (Figure 6, Appendix). SAP is therefore robust to moderate deviations from the default window width.

Table 1: **Brute-force window ablation at $\gamma = 0.10$.** NDCG@5 retention (%) for SAP at nine sliding windows of fixed width $0.2\,L_{total}$, stepped by $0.1\,L_{total}$ across the LLM backbone. † marks the window selected by our SR-guided procedure; the brute-force best window per row is in **bold**. The SR-guided window coincides with the bold-and-dagger cell in each Mean row, i.e., it is optimal on the v1+v2 mean for all three backbones. Full results at $\gamma = 0.20$ and $\gamma = 0.05$ in Appendix K.

Model	Bench	0–20	10–30	20–40	30–50	40–60	50–70	60–80	70–90	80–100
ColPali	v1	88.2	91.8	92.6	93.1	92.7	**94.2**	93.9†	91.9	90.0
	v2	82.9	86.8	87.9	90.6	89.9	88.3	**90.9**†	85.1	81.5
	Mean	85.6	89.3	90.2	91.9	91.3	91.3	**92.4**†	88.5	85.7
ColQwen2	v1	85.2	88.6	90.0	90.7	91.3	92.0	**93.3**†	92.2	91.5
	v2	82.6	82.7	82.1	82.0	81.4	84.3	**88.5**†	85.2	80.9
	Mean	83.9	85.6	86.1	86.3	86.4	88.1	**90.9**†	88.7	86.2
Jina v4	v1	88.3	89.8	91.3	91.5	92.1	93.6	94.9	**95.6**†	94.6
	v2	82.4	81.7	83.6	87.8	88.5	**88.7**	87.5	88.5†	85.6
	Mean	85.3	85.7	87.5	89.7	90.3	91.1	91.2	**92.0**†	90.1

4.4 Computational Efficiency

Beyond retrieval fidelity, SAP maintains high operational throughput. Theoretical complexity analysis and empirical benchmarks (Appendix L.2) show that SAP adds negligible overhead to the total forward pass latency. In contrast, the clustering-based method incurs a significant 6% computational overhead.

On the ViDoRe v2 union (3,006 documents, 1,152 queries), SAP at $\gamma = 0.10$ shrinks the ColPali index from 751 MB to 75 MB (10× reduction) and accelerates end-to-end MaxSim retrieval from 6.1 ms to 0.78 ms per query (7.9× speedup), while retaining >90% NDCG@5. At $\gamma = 0.05$ the index drops to 37 MB (20×) and latency to 0.47 ms (13×). The full benchmark across all γ values is provided in Appendix L.3.

4.5 Comparison with Trained Method

We also benchmark SAP against **Light-ColPali** (Ma et al., 2025), a trained token-merging baseline. At moderate compression (4×, $\gamma = 0.25$) SAP matches or exceeds Light-ColPali without any training; at 9× ($\gamma = 0.10$) SAP stays within a few NDCG@5 points; only at extreme 25× ($\gamma = 0.05$) does the gap widen, consistent with the expected advantage of learned feature fusion in the most aggressive regime. The full comparison table is provided in Appendix M.

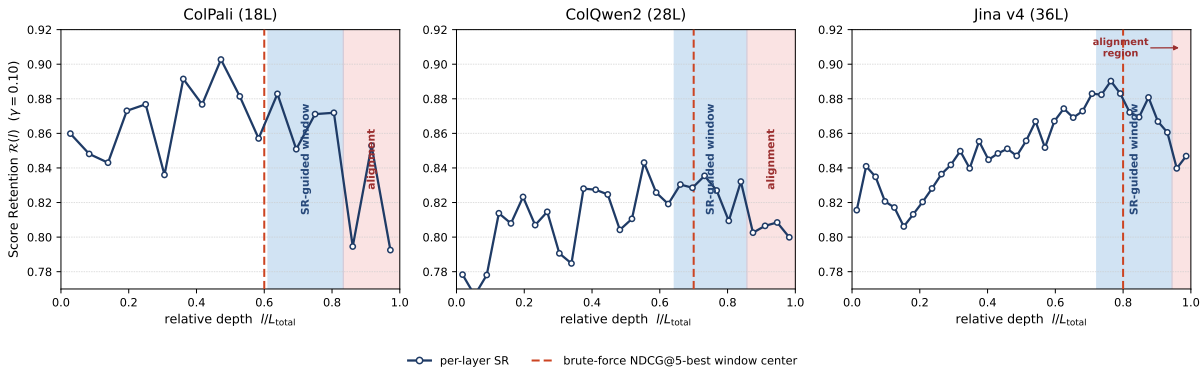

Figure 5: **Per-layer Score Retention and SR-guided window selection across three backbones.** For each backbone, $\mathcal{R}(l)$ at $\gamma = 0.10$ is averaged over the 500-pair calibration set. The blue band marks the window selected by our SR-guided procedure; the red band marks the alignment region; the vertical dashed line marks the brute-force NDCG@5-best window center (Table 1). The two window agree to within $\sim$1% NDCG@5. The Alignment-Aggregation Divergence is visible in all three backbones as a structural plateau followed by a sharp suffix drop in the alignment region.

4.6 Score Retention vs. NDCG

We assess the empirical relationship between SR and downstream NDCG. Across all (model, benchmark, method, γ, dataset) configurations we observe a moderate positive Pearson correlation ($r \approx 0.60$). This aligns with what theory predicts: SR measures per-pair score fidelity, NDCG measures corpus-level ranking quality relative to hard negatives, and these are conceptually distinct quantities. The full SR–NDCG scatter is in Appendix N.

5 Alignment-Aggregation Divergence

Having established the retrieval performance of SAP, we turn to the mechanistic question: why does the semantic signal needed for pruning decouple from the final retrieval embedding? We apply the SR protocol to every layer of the three backbones using the same calibration set from evaluation. Figure 5 reveals two distinct phases in every backbone, regardless of depth.

Before the alignment region, the per-layer SR forms a sustained plateau (blue band in Figure 5). Attention here aggregates local visual features into high-centrality anchor patches that constitute the document's "semantic core". The plateau appears consistently across all three evaluated backbones despite a 2$\times$ depth range from 18 to 36 layers.

Approaching the final layers, SR drops sharply (red band). We attribute this to the contrastive MaxSim objective: to maximize separability against hard negatives, the model reorganizes representations into a sparse, query-aligned form. This optimization is beneficial for ranking but destroys the dense structural signal needed for pruning, which is why final-layer EOS-Attention consistently falls below random selection in our main results (Figure 4). The empirically detected boundary between the two phases is precisely what the SR-guided window selection procedure operationalizes.

6 Conclusion

We propose Structural Anchor Pruning (SAP), a self-calibrating, training-free, and query-agnostic index-time compression framework for multi-vector visual document retrieval. SAP couples a white-box Score Retention (SR) diagnostic with an SR-guided window selection procedure that automatically locates the structural pruning region of any backbone without labels or per-model tuning, and uses visual in-degree centrality to identify anchor patches within that window. SAP reduces index storage by over 90% while retaining >90% NDCG@5 across three architectures spanning 18 to 36 layers. Our layer-resolved analysis uncovers the Alignment-Aggregation Divergence: a stable Structural Plateau within the backbone preserves the document's visual structure, while the final layers compress this representation into a sparse, retrieval-aligned form that is no longer suitable for pruning.

7 Limitations

Our evaluation is confined to the multi-vector late-interaction VDR paradigm. Applicability to single-vector dense retrievers, generic image-text matching, and end-to-end retrieval-augmented generation

pipelines is unexplored.

The calibration set is drawn from a VDR-relevant corpus (the ColPali training data), and our per-layer SR analysis is averaged over this distribution. The Structural Plateau location and the SR-guided window may therefore shift under more aggressive visual-domain changes, such as natural-image retrieval, scientific figures, or satellite imagery, where the visual statistics and the relevant feature scales differ markedly from document pages. Verifying domain robustness under such shifts is left for future work.

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*.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. 2022. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*.

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.

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*.

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, Mingdong Ou, Yi Cao, Xin Zou, Jiahao Huo, Shuliang Liu, James Kwok, and Xuming Hu. 2026. Sculpting the vector space: Towards efficient multi-vector visual document retrieval via prune-then-merge framework. *Preprint*, arXiv:2602.19549.

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. 2024a. Vision-language models for vision tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 46(8):5625–5644.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and 1 others. 2024b. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*.

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 2 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 2: **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 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 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:

$$S_{random}(j) \sim \mathcal{U}(0, 1) \tag{9}$$

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}:

$$S_{eos}(j) = \frac{1}{H} \sum_{h=1}^{H} A_{eos,j}^{(L_{last},h)} \tag{10}$$

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

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):

$$\min_{\{\mu_1,\ldots,\mu_K\}} \sum_{k=1}^{K} \sum_{v_j \in C_k} \|v_j - \mu_k\|^2 \tag{11}$$

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:

$$S_{eos}(j) > \mu_d + k \cdot \sigma_d \tag{12}$$

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:

$$k = \text{Quantile}(\{z_{ij}\}_{\text{calib}}, 1 - \gamma) \tag{13}$$

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.

C Dataset Details

Table 3 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 3: **Dataset Specifications.** Overview of domains covered in the ViDoRe benchmark suite.

D SR-Guided Window Selection: Formal Algorithm

We provide formal pseudocode for the SR-guided window selection procedure described in Section 3.3.

Calibration cost. On a V100-32GB GPU, the procedure completes within ∼30 min for ColPali (18 layers) and within ∼2 h for Jina v4 (36 layers) with $N = 500$ pairs. Since the window selection runs once per backbone, offline, this cost is amortized across all subsequent queries.

E Per-Architecture Window Instantiation

The SR-guided window selection procedure is run once per backbone on 500 (image, query) pairs uniformly sampled from the ColPali training corpus (`vidore/colpali_train_set`), which is disjoint from all ViDoRe evaluation splits. Table 4 reports the resulting per-architecture windows together with a head-to-head comparison against the brute-force best window from Table 1. For all

Algorithm 1 SR-Guided Window Selection

Require: Backbone $\mathcal{M}$ with L_{total} layers; calibration set $\mathcal{C} = \{(I_n, Q_n)\}_{n=1}^{N}$; retention ratio γ; relative window width ρ.
Ensure: Window (α, β).

```
r_l ← 0 for all l ∈ {0, ..., L_total − 1}
for (I, Q) ∈ C do
    Forward I through M to obtain per-layer attention tensors {A^(l)} and full visual embeddings E_I.
    Encode query: E_Q ← M_txt(Q).
    full ← MaxSim(E_Q, E_I)
    for l ∈ {0, ..., L_total − 1} do
        c ← S^(l) from A^(l)                          ▷ Eq. 3
        Ê_I ← top-⌈γ|E_I|⌉(E_I, c)
        r_l ← r_l + MaxSim(E_Q, Ê_I)/full             ▷ SR, Eq. 7
    end for
end for
R(l) ← r_l/N for all l
m ← median{R(l)}_{l=0}^{L_total−1}
l* ← smallest index s.t. R(l) < m for all l ∈ [l*, L_total − 1]
k ← ⌈ρ · L_total⌉
β ← l*/L_total;   α ← (l* − k)/L_total
return (α, β)
```

three backbones the SR-guided window matches the brute-force optimum (NDCG gap 0.0). The alignment region (suffix layers with $\mathcal{R}(l) < m$) confirms an architecture-invariant pattern: 2–4 final layers dominated by retrieval-objective alignment.

Table 4: **SR-guided window selection per backbone** ($\gamma = 0.10$, 500-pair calibration set). The procedure automatically determines the window position; width is fixed at $k = \lceil 0.2\,L \rceil$. "NDCG gap" is the NDCG@5 retention gap between the SR-guided window and the brute-force best from Table 1.

Backbone	L	l^*	Layers	Snap	Gap
ColPali	18	15	11–14	60–80%	−0.3
ColQwen2	28	24	18–23	60–80%	0.0
Jina v4	36	34	26–33	70–90%	0.0

F Three Typical Compression Ratios: Detailed Case Study

To complement the continuous γ sweep in Figure 4, Table 5 reports the three operational regimes most commonly cited in prior work (Ma et al., 2025; Yan et al., 2025): moderate compression ($\gamma = 0.20$, $5\times$), aggressive ($\gamma = 0.10$, $10\times$), and extreme ($\gamma = 0.05$, $20\times$). For each (backbone, benchmark, method, γ) we report Score Retention (Eq. 7), absolute NDCG@5, and the NDCG@5 retention percentage relative to the full-model upper bound.

Performance Consistency at $\gamma = 0.10$. SAP retains over 90% of NDCG@5 across both benchmarks and all three backbones at $\gamma = 0.10$, the standard $10\times$ compression target. At $\gamma = 0.05$ on ViDoRe v2, where baselines drop to 54–69%, SAP maintains 76–79% — a 22–25 point margin that closes the high-compression gap previously cited as a barrier to training-free deployment.

Universality at the same γ. The same three γ values yield qualitatively identical method rankings across backbones spanning 18 to 36 layers: SAP > Cluster/Random > EOS-Adaptive at $\gamma = 0.20$, and the gap widens as γ shrinks. No per-architecture tuning of SAP is performed.

Cross-Cutoff Stability. To verify the $\gamma = 0.10$ numbers are not artifacts of NDCG@5 specifically, Appendix H reports NDCG@$\{1, 5, 10, 50, 100\}$ retention for SAP at all three typical γ values. The spread across cutoffs is below 2.5% at $\gamma = 0.20$ and below 8% at $\gamma = 0.05$ on both benchmarks, confirming SAP exhibits no cutoff-sensitive degradation. Full per-dataset numbers across all 14 ViDoRe datasets are provided next in Appendix G.

Model	Method	Upper Bound	$\gamma = 0.20$			$\gamma = 0.10$			$\gamma = 0.05$		
		Full NDCG@5	S.Ret	NDCG@5	%	S.Ret	NDCG@5	%	S.Ret	NDCG@5	%
Benchmark: ViDoRe v1 (Avg. across 10 datasets)											
ColPali	EOS-Adaptive	0.85	0.86	0.76	89.02	0.79	0.70	81.16	0.72	0.63	73.26
	Random		0.91	0.81	94.61	0.85	0.77	89.82	0.78	0.71	82.92
	Cluster		0.89	0.82	96.53	0.80	0.77	90.42	0.69	0.67	78.22
	SAP (Ours)		**0.94**	**0.83**	**97.48**	**0.89**	**0.80**	**93.88**	**0.81**	**0.74**	**86.34**
ColQwen2	EOS-Adaptive	0.88	0.84	0.77	87.90	0.75	0.70	79.54	0.66	0.63	71.18
	Random		0.86	0.83	94.10	0.78	0.78	87.83	0.68	0.70	78.65
	Cluster		0.82	0.84	95.97	0.70	0.79	90.14	0.59	0.73	82.66
	SAP (Ours)		**0.92**	**0.85**	**97.08**	**0.86**	**0.82**	**93.23**	**0.78**	**0.75**	**85.56**
Jina Embeddings v4	EOS-Adaptive	0.90	0.87	0.85	95.07	0.79	0.79	87.61	0.71	0.67	73.36
	Random		0.89	0.86	95.21	0.82	0.82	90.42	0.74	0.75	82.40
	Cluster		0.83	0.86	95.97	0.73	0.81	90.37	0.62	0.75	83.23
	SAP (Ours)		**0.91**	**0.88**	**98.42**	**0.85**	**0.86**	**95.57**	**0.78**	**0.81**	**89.55**
Benchmark: ViDoRe v2 (Avg. across 4 datasets)											
ColPali	EOS-Adaptive	0.56	0.87	0.45	80.93	0.80	0.40	71.85	0.72	0.33	59.52
	Random		0.90	0.49	87.72	0.84	0.44	78.54	0.77	0.38	69.42
	Cluster		0.88	0.46	83.40	0.79	0.38	69.00	0.67	0.30	54.37
	SAP (Ours)		**0.94**	**0.52**	**93.44**	**0.89**	**0.50**	**90.38**	**0.81**	**0.42**	**76.39**
ColQwen2	EOS-Adaptive	0.54	0.84	0.48	89.26	0.76	0.43	79.92	0.67	0.35	64.33
	Random		0.87	0.46	86.04	0.78	0.41	76.12	0.69	0.35	65.15
	Cluster		0.80	0.45	83.56	0.70	0.41	76.25	0.59	0.33	61.68
	SAP (Ours)		**0.91**	**0.50**	**91.91**	**0.85**	**0.48**	**88.07**	**0.77**	**0.43**	**78.96**
Jina Embeddings v4	EOS-Adaptive	0.58	0.90	0.53	90.82	0.82	0.49	83.73	0.73	0.38	65.31
	Random		0.89	0.51	87.78	0.83	0.47	80.44	0.75	0.39	67.51
	Cluster		0.83	0.47	80.74	0.73	0.40	67.99	0.63	0.33	57.14
	SAP (Ours)		**0.92**	**0.56**	**95.78**	**0.87**	**0.51**	**88.52**	**0.81**	**0.44**	**76.51**

Table 5: **Three typical compression ratios.** NDCG@5 retention (%) for training-free compression methods at $\gamma \in \{0.20, 0.10, 0.05\}$ across three VLM backbones and two benchmark suites. **S.Ret** (Score Retention, Eq. 7) diagnoses per-pair MaxSim fidelity; the **%** column reports $(\text{NDCG@5}_{\text{pruned}}/\text{NDCG@5}_{\text{full}}) \times 100$. SAP uses our SR-guided window (ColPali & ColQwen2: 60–80%, Jina v4: 70–90%; see Section 3.3). Best per column in **bold**.

G Detailed Evaluation Results

Extending the aggregated case study in Appendix F, we report the full per-dataset NDCG@5 retention for SAP at the SR-guided window against the three training-free baselines on each of the 14 ViDoRe datasets. Tables 6–8 cover ViDoRe v1 for ColPali, ColQwen2, and Jina v4 respectively; Tables 9–11 cover ViDoRe v2 for the same three backbones. SAP's relative advantage varies across datasets; the variance is largely explained by the per-architecture optimal window position revealed by our SR-guided procedure .

Dataset	Method	Upper Bound	$\gamma = 0.20$			$\gamma = 0.10$			$\gamma = 0.05$		
		Full NDCG@5	S.Ret	NDCG@5	%	S.Ret	NDCG@5	%	S.Ret	NDCG@5	%
ArxivQA	EOS-Adaptive	0.82	0.91	0.77	93.82	0.85	0.73	89.08	0.78	0.68	83.67
	Random		0.93	0.80	98.04	0.89	0.78	95.90	0.83	0.75	91.29
	Cluster		0.92	0.81	99.07	0.87	0.80	98.07	**0.80**	**0.79**	**96.25**
	SAP (Ours)		**0.95**	**0.82**	**99.75**	**0.90**	**0.81**	**98.87**	0.84	0.77	94.25
DocVQA	EOS-Adaptive	0.59	0.91	0.50	84.18	0.86	0.43	72.33	0.81	0.35	59.41
	Random		0.94	0.54	91.65	0.90	0.50	85.03	0.84	0.43	73.58
	Cluster		0.92	0.55	93.44	0.86	0.51	86.09	0.77	0.45	76.05
	SAP (Ours)		**0.97**	**0.57**	**96.51**	**0.93**	**0.52**	**88.71**	**0.86**	**0.46**	**78.82**
InfoVQA	EOS-Adaptive	0.85	0.86	0.77	89.92	0.79	0.72	84.87	0.73	0.68	79.50
	Random		0.91	0.82	96.37	0.86	0.79	93.16	**0.79**	**0.76**	**89.42**
	Cluster		**0.90**	**0.84**	**98.04**	0.82	0.81	94.64	0.71	0.74	87.10
	SAP (Ours)		0.94	0.83	97.63	**0.88**	**0.81**	**94.96**	0.80	0.75	87.84
Shift Project	EOS-Adaptive	0.77	0.83	0.60	77.60	0.75	0.48	62.41	0.66	0.40	51.41
	Random		0.90	0.68	88.04	0.84	0.61	78.99	0.76	0.51	65.23
	Cluster		0.85	0.68	87.25	0.74	0.53	68.78	0.60	0.30	38.82
	SAP (Ours)		**0.94**	**0.74**	**95.29**	**0.88**	**0.66**	**85.45**	**0.81**	**0.57**	**73.24**
SynDocQA-AI	EOS-Adaptive	0.97	0.87	0.94	96.67	0.80	0.88	90.58	0.74	0.86	88.11
	Random		0.90	0.95	97.96	0.84	0.92	95.04	0.76	0.87	89.56
	Cluster		**0.88**	**0.96**	**99.04**	0.78	0.92	94.37	0.67	0.82	84.51
	SAP (Ours)		0.94	0.95	98.08	**0.89**	**0.93**	**95.77**	**0.81**	**0.90**	**92.06**
SynDocQA-Energy	EOS-Adaptive	0.96	0.84	0.87	90.69	0.76	0.77	79.94	0.69	0.75	78.23
	Random		0.90	0.93	96.87	0.84	0.90	93.50	0.76	0.85	88.23
	Cluster		**0.89**	**0.95**	**98.76**	0.80	0.92	95.66	0.67	0.80	83.65
	SAP (Ours)		0.94	0.93	97.15	**0.89**	**0.92**	**96.30**	**0.80**	**0.85**	**89.09**
SynDocQA-Gov	EOS-Adaptive	0.96	0.85	0.90	92.92	0.79	0.86	88.97	0.70	0.74	76.95
	Random		0.91	0.92	95.05	0.84	0.88	91.38	0.77	0.84	86.68
	Cluster		0.88	0.94	97.42	0.78	0.90	93.57	0.65	0.78	81.13
	SAP (Ours)		**0.95**	**0.95**	**98.60**	**0.90**	**0.94**	**97.36**	**0.80**	**0.86**	**89.21**
SynDocQA-Health	EOS-Adaptive	0.97	0.87	0.93	96.35	0.81	0.89	91.89	0.74	0.80	82.64
	Random		0.91	0.94	97.57	0.84	0.91	94.16	0.78	0.87	90.04
	Cluster		0.88	0.95	98.37	0.77	0.88	91.05	0.65	0.70	72.73
	SAP (Ours)		**0.95**	**0.95**	**98.74**	**0.90**	**0.95**	**98.53**	**0.82**	**0.90**	**93.35**
TabFQuAD	EOS-Adaptive	0.87	0.82	0.79	90.95	0.75	0.76	86.90	0.67	0.70	80.39
	Random		0.91	0.85	97.92	0.85	0.83	95.40	0.77	0.79	91.13
	Cluster		**0.91**	**0.88**	**100.53**	**0.85**	**0.87**	**100.11**	**0.77**	**0.86**	**98.70**
	SAP (Ours)		0.94	0.86	99.13	0.89	0.85	97.10	0.81	0.82	94.09
TAT-DQA	EOS-Adaptive	0.71	0.85	0.55	77.11	0.79	0.46	64.66	0.73	0.37	52.34
	Random		0.89	0.62	86.58	0.81	0.54	75.70	0.73	0.46	64.04
	Cluster		0.86	0.67	93.35	0.76	0.58	81.85	0.64	0.45	63.24
	SAP (Ours)		**0.94**	**0.67**	**93.91**	**0.88**	**0.61**	**85.73**	**0.79**	**0.51**	**71.41**

Table 6: **Detailed Performance: ColPali on ViDoRe v1.** Per-dataset NDCG@5 retention with SAP using the SR-guided window . Best per column in **bold**.

Dataset	Method	Upper Bound	$\gamma = 0.20$			$\gamma = 0.10$			$\gamma = 0.05$		
		Full NDCG@5	S.Ret	NDCG@5	%	S.Ret	NDCG@5	%	S.Ret	NDCG@5	%
ArxivQA	EOS-Adaptive	0.86	0.85	0.76	88.38	0.76	0.71	82.59	0.66	0.63	73.50
	Random		0.91	0.82	95.74	0.84	0.79	92.22	0.76	0.75	86.82
	Cluster		**0.87**	**0.84**	**97.16**	**0.79**	**0.82**	**95.65**	**0.69**	**0.79**	**92.03**
	SAP (Ours)		0.93	0.82	95.48	0.87	0.79	91.62	0.78	0.74	85.42
DocVQA	EOS-Adaptive	0.59	0.87	0.49	83.27	0.80	0.40	69.00	0.71	0.30	51.04
	Random		0.86	0.52	89.04	0.77	0.46	77.75	0.66	0.37	62.83
	Cluster		0.82	0.56	94.74	0.71	0.51	87.41	0.59	0.44	75.37
	SAP (Ours)		**0.93**	**0.56**	**95.62**	**0.88**	**0.52**	**88.45**	**0.81**	**0.47**	**79.90**
InfoVQA	EOS-Adaptive	0.91	0.83	0.82	90.90	0.74	0.76	84.26	0.65	0.72	78.91
	Random		0.86	0.85	93.53	0.78	0.81	89.14	0.70	0.75	82.52
	Cluster		0.81	0.87	95.89	0.69	0.81	89.70	0.56	0.72	79.25
	SAP (Ours)		**0.91**	**0.88**	**96.74**	**0.85**	**0.84**	**92.67**	**0.77**	**0.78**	**85.71**
Shift Project	EOS-Adaptive	0.86	0.84	0.68	79.63	0.76	0.61	71.49	0.67	0.52	60.65
	Random		0.85	0.77	90.03	0.76	0.66	77.74	0.66	0.53	61.49
	Cluster		0.79	0.77	90.04	0.66	0.63	73.88	0.54	0.54	63.56
	SAP (Ours)		**0.91**	**0.83**	**97.59**	**0.85**	**0.81**	**95.04**	**0.76**	**0.69**	**80.36**
SynDocQA-AI	EOS-Adaptive	0.98	0.82	0.86	87.29	0.74	0.83	84.51	0.64	0.79	80.91
	Random		0.87	0.97	98.45	0.78	0.94	95.78	0.68	0.86	87.94
	Cluster		**0.81**	**0.98**	**99.69**	**0.70**	**0.96**	**97.27**	**0.58**	**0.88**	**89.57**
	SAP (Ours)		0.92	0.96	97.34	0.85	0.93	95.11	0.76	0.87	88.24
SynDocQA-Energy	EOS-Adaptive	0.96	0.82	0.84	87.61	0.72	0.70	72.79	0.62	0.66	69.48
	Random		0.86	0.90	94.01	0.77	0.84	88.03	0.68	0.77	80.90
	Cluster		**0.82**	**0.92**	**96.26**	0.70	0.88	92.40	0.58	0.83	86.58
	SAP (Ours)		0.92	0.92	95.72	**0.86**	**0.90**	**93.85**	**0.79**	**0.86**	**89.90**
SynDocQA-Gov	EOS-Adaptive	0.94	0.84	0.91	96.20	0.74	0.84	88.93	0.64	0.76	80.86
	Random		0.87	0.94	99.28	0.78	0.90	94.97	0.69	0.82	87.18
	Cluster		0.80	0.91	96.55	0.69	0.88	93.37	0.58	0.85	89.74
	SAP (Ours)		**0.93**	**0.94**	**99.33**	**0.86**	**0.91**	**96.53**	**0.79**	**0.87**	**92.29**
SynDocQA-Health	EOS-Adaptive	0.98	0.83	0.93	95.16	0.76	0.89	91.40	0.67	0.81	82.82
	Random		0.86	0.96	97.88	0.78	0.93	95.21	0.70	0.88	89.69
	Cluster		**0.81**	**0.97**	**99.49**	0.69	0.91	93.22	0.56	0.85	86.92
	SAP (Ours)		0.93	0.97	98.81	**0.87**	**0.97**	**98.81**	**0.79**	**0.91**	**93.15**
TabFQuAD	EOS-Adaptive	0.88	0.87	0.81	92.38	0.78	0.74	84.01	0.69	0.68	77.87
	Random		0.87	0.85	97.19	0.79	0.82	93.39	0.70	0.76	86.98
	Cluster		0.85	0.86	98.41	**0.75**	**0.85**	**96.65**	**0.64**	**0.83**	**94.09**
	SAP (Ours)		**0.94**	**0.88**	**100.23**	0.88	0.84	95.57	0.80	0.80	91.42
TAT-DQA	EOS-Adaptive	0.81	0.83	0.63	78.12	0.72	0.54	66.39	0.61	0.45	55.73
	Random		0.82	0.70	85.89	0.72	0.60	74.11	0.61	0.49	60.12
	Cluster		0.79	0.74	91.46	0.67	0.66	81.81	**0.55**	**0.56**	**69.49**
	SAP (Ours)		**0.92**	**0.76**	**93.95**	**0.84**	**0.69**	**84.71**	0.72	0.56	69.19

Table 7: **Detailed Performance: ColQwen2 on ViDoRe v1.** Per-dataset NDCG@5 retention with SAP using the SR-guided window . Best per column in **bold**.

Dataset	Method	Upper Bound	$\gamma = 0.20$			$\gamma = 0.10$			$\gamma = 0.05$		
		Full NDCG@5	S.Ret	NDCG@5	%	S.Ret	NDCG@5	%	S.Ret	NDCG@5	%
ArxivQA	EOS-Adaptive	0.88	0.90	0.84	94.93	0.82	0.77	86.81	0.73	0.64	72.19
	Random		0.92	0.86	96.85	0.86	0.82	93.14	0.79	0.77	87.33
	Cluster		**0.88**	**0.87**	**98.80**	**0.81**	**0.85**	**95.83**	**0.72**	**0.82**	**92.86**
	SAP (Ours)		0.94	0.86	97.72	0.89	0.83	94.40	0.82	0.77	86.91
DocVQA	EOS-Adaptive	0.61	0.88	0.56	90.94	0.81	0.48	78.87	0.71	0.34	56.27
	Random		0.87	0.53	86.65	0.79	0.48	77.50	0.70	0.40	65.72
	Cluster		0.84	0.56	91.80	0.73	0.49	80.08	0.60	0.43	69.62
	SAP (Ours)		**0.94**	**0.60**	**98.36**	**0.89**	**0.59**	**96.76**	**0.82**	**0.54**	**87.69**
InfoVQA	EOS-Adaptive	0.92	0.84	0.86	93.08	0.75	0.77	83.86	0.64	0.60	65.54
	Random		0.87	0.88	95.70	0.80	0.85	92.27	0.72	0.78	84.40
	Cluster		0.83	0.89	96.39	0.71	0.83	90.20	0.59	0.74	80.42
	SAP (Ours)		**0.92**	**0.90**	**97.94**	**0.86**	**0.86**	**93.93**	**0.78**	**0.79**	**86.20**
Shift Project	EOS-Adaptive	0.89	0.85	0.87	97.81	0.79	0.81	90.16	0.72	0.70	77.83
	Random		0.88	0.84	94.09	0.81	0.78	87.01	0.73	0.65	72.93
	Cluster		0.81	0.82	91.32	0.70	0.72	80.96	0.59	0.58	65.31
	SAP (Ours)		**0.90**	**0.90**	**100.93**	**0.84**	**0.86**	**96.05**	**0.78**	**0.83**	**92.44**
SynDocQA-AI	EOS-Adaptive	0.99	0.84	0.98	98.39	0.77	0.90	90.67	0.69	0.78	78.97
	Random		0.89	0.98	98.77	0.82	0.95	96.24	0.74	0.89	89.56
	Cluster		0.81	0.97	97.90	0.71	0.92	93.11	0.60	0.90	90.43
	SAP (Ours)		**0.87**	**0.98**	**99.00**	**0.81**	**0.97**	**98.00**	**0.75**	**0.94**	**94.63**
SynDocQA-Energy	EOS-Adaptive	0.96	0.85	0.91	94.74	0.78	0.86	89.37	0.70	0.72	74.87
	Random		0.89	0.94	98.18	0.82	0.90	93.52	0.73	0.82	85.64
	Cluster		**0.81**	**0.95**	**98.59**	**0.71**	**0.91**	**94.28**	0.60	0.84	87.15
	SAP (Ours)		0.89	0.94	97.76	0.83	0.90	93.56	**0.76**	**0.87**	**91.00**
SynDocQA-Gov	EOS-Adaptive	0.97	0.85	0.94	97.18	0.78	0.91	93.97	0.71	0.79	81.78
	Random		0.89	0.94	96.98	0.82	0.91	93.99	0.74	0.86	88.76
	Cluster		0.82	0.95	98.09	0.71	0.92	94.41	0.60	0.87	89.15
	SAP (Ours)		**0.87**	**0.96**	**98.51**	**0.82**	**0.94**	**97.33**	**0.75**	**0.89**	**92.06**
SynDocQA-Health	EOS-Adaptive	0.98	0.85	0.97	99.12	0.78	0.93	94.61	0.71	0.82	83.81
	Random		0.89	0.97	98.68	0.82	0.95	96.89	0.74	0.90	91.56
	Cluster		0.81	0.97	98.64	0.70	0.90	92.10	0.60	0.87	88.77
	SAP (Ours)		**0.88**	**0.98**	**99.62**	**0.82**	**0.97**	**99.43**	**0.75**	**0.93**	**94.90**
TabFQuAD	EOS-Adaptive	0.96	0.89	0.92	96.78	0.82	0.88	92.55	0.74	0.83	86.60
	Random		0.90	0.94	98.46	0.84	0.92	96.08	0.76	0.89	92.70
	Cluster		0.85	0.94	98.16	**0.77**	**0.93**	**97.64**	**0.67**	**0.90**	**94.70**
	SAP (Ours)		**0.93**	**0.94**	**98.75**	0.88	0.92	96.74	0.79	0.87	91.29
TAT-DQA	EOS-Adaptive	0.79	0.90	0.69	87.76	0.82	0.59	75.25	0.72	0.44	55.76
	Random		0.88	0.69	87.71	0.81	0.61	77.54	0.72	0.52	65.44
	Cluster		0.81	0.71	90.05	0.71	0.67	85.09	0.61	0.58	73.87
	SAP (Ours)		**0.94**	**0.75**	**95.57**	**0.89**	**0.71**	**89.51**	**0.82**	**0.62**	**78.33**

Table 8: **Detailed Performance: Jina Embeddings v4 on ViDoRe v1.** Per-dataset NDCG@5 retention with SAP using the SR-guided window . Best per column in **bold**.

Dataset	Method	Upper Bound	$\gamma = 0.20$			$\gamma = 0.10$			$\gamma = 0.05$		
		Full NDCG@5	S.Ret	NDCG@5	%	S.Ret	NDCG@5	%	S.Ret	NDCG@5	%
Biomed Lec.	EOS-Adaptive	0.57	0.88	0.50	87.16	0.82	0.44	77.73	0.76	0.39	68.66
	Random		0.93	0.54	94.77	0.89	0.52	90.71	**0.82**	**0.47**	**82.12**
	Cluster		0.92	0.55	95.89	0.86	0.51	88.92	0.78	0.46	80.45
	SAP (Ours)		**0.94**	**0.55**	**96.20**	**0.89**	**0.53**	**93.70**	0.82	0.46	80.57
Econ. Reports	EOS-Adaptive	0.51	0.84	0.44	86.34	0.78	0.42	83.42	**0.71**	**0.41**	**80.55**
	Random		0.88	0.45	88.39	0.81	0.42	82.07	0.74	0.39	76.68
	Cluster		0.87	0.43	84.69	0.77	0.36	70.69	0.64	0.27	52.20
	SAP (Ours)		**0.93**	**0.47**	**92.83**	**0.86**	**0.46**	**90.61**	0.77	0.40	77.77
ESG (Multi)	EOS-Adaptive	0.54	0.87	0.43	78.57	0.80	0.39	71.21	0.71	0.25	46.42
	Random		0.90	0.48	87.97	0.83	0.41	75.80	0.75	0.34	62.66
	Cluster		0.86	0.40	74.32	0.76	0.29	52.42	0.63	0.23	42.97
	SAP (Ours)		**0.95**	**0.50**	**92.18**	**0.90**	**0.48**	**88.31**	**0.83**	**0.42**	**77.29**
ESG (Human)	EOS-Adaptive	0.60	0.87	0.43	71.66	0.80	0.33	55.06	0.71	0.25	42.46
	Random		0.90	0.48	79.76	0.84	0.39	65.56	0.76	0.34	56.22
	Cluster		0.86	0.47	78.70	0.76	0.38	63.95	0.64	0.25	41.85
	SAP (Ours)		**0.95**	**0.55**	**92.53**	**0.91**	**0.53**	**88.88**	**0.84**	**0.42**	**69.93**

Table 9: **Detailed Performance: ColPali on ViDoRe v2.** Per-dataset NDCG@5 retention with SAP using the SR-guided window . Best per column in **bold**.

Dataset	Method	Upper Bound	$\gamma = 0.20$			$\gamma = 0.10$			$\gamma = 0.05$		
		Full NDCG@5	S.Ret	NDCG@5	%	S.Ret	NDCG@5	%	S.Ret	NDCG@5	%
Biomed Lec.	EOS-Adaptive	0.54	0.84	0.46	84.58	0.77	0.43	78.49	0.67	0.38	69.43
	Random		0.90	0.52	95.09	0.83	0.49	89.89	0.76	0.45	82.05
	Cluster		0.85	0.51	93.92	0.75	0.48	87.75	0.64	0.43	79.19
	SAP (Ours)		**0.93**	**0.53**	**96.94**	**0.87**	**0.51**	**93.24**	**0.81**	**0.48**	**88.11**
Econ. Reports	EOS-Adaptive	0.48	0.80	0.43	89.49	0.72	0.41	85.09	0.63	0.36	75.09
	Random		0.85	0.41	85.54	0.77	0.40	82.83	0.68	0.35	73.44
	Cluster		0.79	0.40	83.09	0.67	0.35	72.04	0.55	0.29	59.33
	SAP (Ours)		**0.89**	**0.45**	**93.57**	**0.82**	**0.42**	**86.93**	**0.74**	**0.38**	**79.79**
ESG (Multi)	EOS-Adaptive	0.57	**0.89**	**0.56**	**98.69**	0.81	0.46	81.51	0.71	0.38	67.19
	Random		0.86	0.47	82.04	0.77	0.37	65.14	0.67	0.30	52.02
	Cluster		0.79	0.43	76.40	0.69	0.38	67.30	0.58	0.30	53.04
	SAP (Ours)		0.91	0.51	88.83	**0.84**	**0.49**	**86.43**	**0.77**	**0.41**	**71.80**
ESG (Human)	EOS-Adaptive	0.57	0.85	0.48	84.30	0.76	0.43	74.57	0.65	0.26	45.62
	Random		0.85	0.46	81.48	0.76	0.38	66.63	0.66	0.30	53.09
	Cluster		0.79	0.46	80.81	0.69	0.44	77.91	0.58	0.31	55.18
	SAP (Ours)		**0.92**	**0.50**	**88.29**	**0.85**	**0.49**	**85.68**	**0.76**	**0.43**	**76.13**

Table 10: **Detailed Performance: ColQwen2 on ViDoRe v2.** Per-dataset NDCG@5 retention with SAP using the SR-guided window . Best per column in **bold**.

Dataset	Method	Upper Bound	$\gamma = 0.20$			$\gamma = 0.10$			$\gamma = 0.05$		
		Full NDCG@5	S.Ret	NDCG@5	%	S.Ret	NDCG@5	%	S.Ret	NDCG@5	%
Biomed Lec.	EOS-Adaptive	0.61	0.90	0.57	93.15	0.81	0.52	84.64	0.71	0.43	70.94
	Random		0.91	0.58	95.78	**0.86**	**0.56**	**92.08**	0.79	0.52	85.09
	Cluster		0.86	0.57	94.07	0.79	0.55	89.88	0.70	0.48	78.95
	SAP (Ours)		**0.92**	**0.60**	**97.94**	0.87	0.55	90.96	**0.82**	**0.52**	**85.72**
Econ. Reports	EOS-Adaptive	0.55	0.87	0.52	94.91	0.79	0.48	86.98	**0.71**	**0.43**	**77.51**
	Random		0.88	0.49	89.38	0.81	0.44	80.53	0.74	0.40	72.68
	Cluster		0.81	0.43	77.64	0.70	0.35	62.91	0.60	0.30	54.13
	SAP (Ours)		**0.91**	**0.55**	**99.45**	**0.84**	**0.48**	**88.27**	0.77	0.42	76.22
ESG (Multi)	EOS-Adaptive	0.53	0.91	0.45	84.56	0.84	0.43	81.14	0.74	0.34	64.37
	Random		0.89	0.45	84.20	0.82	0.40	76.15	0.73	0.30	57.03
	Cluster		0.82	0.41	78.38	0.72	0.34	63.37	0.63	0.27	51.91
	SAP (Ours)		**0.93**	**0.50**	**94.54**	**0.88**	**0.46**	**87.79**	**0.82**	**0.41**	**77.93**
ESG (Human)	EOS-Adaptive	0.64	0.91	0.58	90.68	0.85	0.52	82.17	0.75	0.31	48.41
	Random		0.89	0.52	81.75	0.83	0.46	73.03	0.74	0.35	55.24
	Cluster		0.81	0.46	72.86	0.70	0.35	55.78	0.60	0.28	43.57
	SAP (Ours)		**0.93**	**0.58**	**91.18**	**0.88**	**0.55**	**87.07**	**0.82**	**0.42**	**66.15**

Table 11: **Detailed Performance: Jina Embeddings v4 on ViDoRe v2.** Per-dataset NDCG@5 retention with SAP using the SR-guided window . Best per column in **bold**.

H Full NDCG Cutoff Results

Tables 12 and 13 report NDCG@k retention for $k \in \{1, 5, 10, 50, 100\}$ at the SR-guided window for SAP, on ViDoRe v1 and v2. The spread across cutoffs is below 2.5% at $\gamma = 0.20$ and below 8% at $\gamma = 0.05$, confirming SAP exhibits no cutoff-sensitive degradation.

I Width Robustness of the SR-Guided Window

Our default fixes the window width at $k = \lceil 0.2\, L_{total} \rceil$ layers. To assess sensitivity to this choice, we hold the window position fixed by the drop-alignment rule and vary only the width k from 1 to L_{total}, measuring NDCG@5 retention at $\gamma = 0.10$ via the nearest 9-window from the brute-force ablation. Figure 6 shows the resulting curves for all three backbones. Each backbone exhibits a broad plateau spanning the majority of valid k values, and our default falls comfortably within it (ColPali: $k = 4$, ColQwen2: $k = 6$, Jina v4: $k = 8$). A single-layer window ($k = 1$) shows variance due to per-layer centrality noise; overly wide windows dilute the structural signal by mixing aggregation and alignment layers. The fixed default therefore balances both extremes and is not a sensitive hyperparameter.

J Calibration Size Robustness

The SR-guided procedure has two hyperparameters: the relative window width ρ (analyzed in Appendix I) and the calibration set size N. To assess sensitivity to N and to the choice of random sample, we sweep $N \in \{50, 100, 200, 500, 1000\}$ and run the procedure with 5 different random seeds for each N (samples drawn uniformly from the ColPali training corpus). For each (N, seed) we record the snap window assigned by the drop-alignment rule.

Table 14 summarizes the outcome: ColPali and Jina v4 converge to a unique window at $N \geq 50$ and $N \geq 100$ respectively; ColQwen2 — the most variance-sensitive backbone — converges to the canonical 60–80% window at $N \geq 500$ (one seed drifts to the adjacent 70–90% window, which differs by only 1.1% NDCG@5; see Table 1). Our default $N = 500$ is therefore sufficient for all three backbones, and doubling to $N = 1000$ yields unanimous seed agreement.

K Full Window Ablation (additional retention ratios)

Tables 15 and 16 report the full 9-window brute-force ablation at $\gamma = 0.20$ and $\gamma = 0.05$. The plateau shift with backbone depth (ColPali peaking around 50–80%, ColQwen2 at 60–80%, Jina v4 at 70–90%) is consistent across all three retention ratios.

L 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.

L.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:

$$\frac{C_{SAP}}{C_{Attn}} \approx \frac{|\mathcal{L}^*| \cdot H \cdot N^2}{L \cdot H \cdot N^2 \cdot d} = \frac{|\mathcal{L}^*|}{L \cdot d} \qquad (14)$$

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.

Table 12: **Full NDCG cutoffs on ViDoRe v1.** NDCG@k retention (%) for SAP at the SR-guided window. **S.Ret** is Score Retention.

Model	Method	γ	S.Ret	@1	@5	@10	@50	@100
ColPali	SAP	0.2	0.94	95.5	97.5	97.5	97.8	97.8
	SAP	0.1	0.89	91.1	93.9	94.5	95.1	95.1
	SAP	0.05	0.81	80.8	86.3	87.3	88.5	88.7
ColQwen2	SAP	0.2	0.92	95.8	97.1	97.1	97.5	97.5
	SAP	0.1	0.86	91.4	93.2	93.7	94.2	94.3
	SAP	0.05	0.78	81.6	85.6	86.5	87.5	87.8
Jina Embeddings v4	SAP	0.2	0.91	98.2	98.4	98.6	98.7	98.7
	SAP	0.1	0.85	93.7	95.6	95.9	96.3	96.3
	SAP	0.05	0.78	85.9	89.5	90.2	91.0	91.2

Table 13: **Full NDCG cutoffs on ViDoRe v2.** NDCG@k retention (%) for SAP at the SR-guided window. **S.Ret** is Score Retention.

Model	Method	γ	S.Ret	@1	@5	@10	@50	@100
ColPali	SAP	0.2	0.94	93.3	93.4	94.7	95.7	96.0
	SAP	0.1	0.89	91.0	90.4	91.3	92.4	93.2
	SAP	0.05	0.81	72.2	76.4	79.2	80.9	82.3
ColQwen2	SAP	0.2	0.91	90.1	91.9	90.8	93.2	93.6
	SAP	0.1	0.85	85.7	88.1	86.7	88.6	89.2
	SAP	0.05	0.77	76.7	79.0	77.8	81.0	81.8
Jina Embeddings v4	SAP	0.2	0.92	94.2	95.8	95.2	96.0	96.2
	SAP	0.1	0.87	88.5	88.5	87.6	90.3	90.6
	SAP	0.05	0.81	72.9	76.5	77.8	81.7	82.8

Table 14: **Calibration robustness.** SR-guided window selected by the procedure across calibration sizes $N \in \{50, 100, 200, 500, 1000\}$ and 5 random seeds (samples drawn from the ColPali training corpus). Each cell reports the most frequent snap window across 5 seeds, with agreement count in parentheses.

Backbone	N=50	N=100	N=200	N=500	N=1000
ColPali (18L)	60–80% (5/5)	60–80% (5/5)	60–80% (5/5)	60–80% (5/5)	60–80% (5/5)
ColQwen2 (28L)	60–80% (3/5)	80–100% (2/5)	70–90% (3/5)	60–80% (4/5)	60–80% (5/5)
Jina v4 (36L)	70–90% (3/5)	70–90% (5/5)	70–90% (5/5)	70–90% (5/5)	70–90% (5/5)

L.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 17.

Results Analysis. As shown in Table 17, the Full Forward Pass requires approximately 206ms per page.

- **Negligible Overhead:** SAP requires only 0.06ms per page, corresponding to $\sim \mathbf{0.03}\%$ overhead 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.

L.3 End-to-End Storage and Retrieval Benchmark

To complement the per-page pruning-latency analysis above, we measure the deployment-time index footprint and retrieval throughput on a real corpus. Setup: ColPali on the ViDoRe v2 union (3,006 documents, 1,152 queries). For each compression ratio

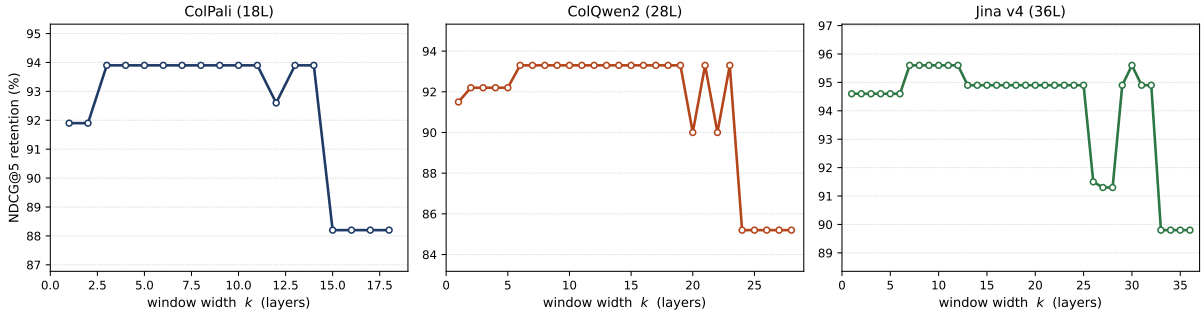

Figure 6: **Width robustness of the SR-guided window.** NDCG@5 retention at $\gamma = 0.10$ (via the nearest 9-window approximation) as the window width k is varied with position fixed by the drop-alignment rule. All three backbones exhibit a broad plateau containing our default $k = \lceil 0.2\, L_{total} \rceil$.

Table 15: **Full window ablation at $\gamma = 0.20$ (NDCG@5 retention %) for SAP.** Brute-force best window per row in **bold**.

Model	Bench	0–20	10–30	20–40	30–50	40–60	50–70	60–80	70–90	80–100
ColPali	v1	94.1	96.3	97.2	97.6	**97.6**	97.5	97.6	97.1	95.4
	v2	89.9	93.4	94.5	93.2	**94.7**	92.9	93.9	93.4	89.6
ColQwen2	v1	93.6	95.0	95.8	95.9	96.4	96.8	97.1	**97.2**	96.8
	v2	89.7	90.1	91.1	91.4	90.6	91.7	92.6	**92.8**	88.9
Jina v4	v1	95.0	96.0	96.5	96.4	96.9	97.6	98.2	**98.4**	98.2
	v2	88.7	89.5	93.0	93.5	95.6	94.8	93.8	**95.8**	94.3

γ, we (1) apply SAP top-$\lceil \gamma N \rceil$ pruning per document, (2) stack the pruned embeddings into a single $[\text{docs}, k, d]$ on-disk index, (3) reload the index into GPU memory, and (4) time end-to-end brute-force MaxSim retrieval (similarity + per-doc max aggregation + top-5 ranking) over all queries on a single V100-32GB GPU. Five warmup queries precede each measurement run; the table below excludes them.

At $\gamma = 0.10$, SAP compresses storage by $10\times$ and accelerates retrieval by $7.9\times$; at $\gamma = 0.05$, by $20\times$ and $13\times$ respectively. The per-query latency scales roughly as $O(\gamma \cdot N_{\text{docs}})$ because the dominant cost is the $\text{query} \times \text{index}$ matmul, whose cost is linear in the total patch count. These measured speedups extrapolate directly to million-scale Visual RAG deployments under the same hardware.

M Comparison with Trained Method (Full Table)

We benchmark SAP against **Light-ColPali** (Ma et al., 2025), a state-of-the-art method that requires supervised training to merge visual tokens. Table 19 reports NDCG@5 and retention % relative to the full-model upper bound. SAP uses our SR-guided window; the compression factor maps to our retention ratio as $\gamma \approx 1/\text{factor}$, and we report SAP at the closest available γ. Note that full-model upper bounds differ slightly between this work and Light-ColPali due to evaluation environments.

N SR–NDCG Correlation: Scatter Plot

Figure 7 reports the SR–NDCG@5 retention scatter across all (model, dataset, method, γ) configurations. SAP (solid markers) consistently occupies the high-fidelity, high-NDCG region; baselines (hollow markers) exhibit substantially more scatter. The Pearson correlation of $r \approx 0.60$ confirms a meaningful but imperfect relationship between SR and NDCG, consistent with their distinct definitions (Section 3.2).

Table 16: **Full window ablation at $\gamma = 0.05$ (NDCG@5 retention %) for SAP.** Brute-force best window per row in **bold**.

Model	Bench	0–20	10–30	20–40	30–50	40–60	50–70	60–80	70–90	80–100
ColPali	v1	81.5	83.5	85.4	86.0	87.3	**88.1**	86.4	85.6	81.9
	v2	74.3	72.2	75.0	75.1	75.5	**80.4**	76.8	73.0	72.5
ColQwen2	v1	68.6	76.4	80.4	82.0	83.2	83.6	85.5	**86.0**	83.8
	v2	68.3	71.8	76.4	76.3	76.2	77.5	**79.6**	79.3	75.2
Jina v4	v1	76.2	80.8	81.9	82.4	84.3	85.4	88.2	**89.5**	87.1
	v2	66.9	73.4	74.8	76.3	77.1	77.3	**79.4**	76.5	78.0

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

Table 17: **Efficiency Benchmark on ViDoRe v2.** Latency is 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 introduces negligible overhead ($< 0.03\%$), whereas clustering-based methods incur a significant penalty ($\approx 6\%$).

Table 18: **End-to-end storage and retrieval benchmark.** ColPali on the ViDoRe v2 union (3,006 documents, 1,152 queries). Index size measured on disk; query latency is the mean over all queries (brute-force MaxSim, V100-32GB). NDCG@5 retention from Table 5.

γ	Size (MB)	Load (ms)	Lat (ms)	QPS	NDCG@5
1.00	751.5	537	6.12	163	100%
0.20	150.4	90	1.37	731	93%
0.10	74.9	51	0.78	1290	90%
0.05	37.4	28	0.47	2114	76%

Table 19: **Training-Free vs. Trained Compression.** SAP vs. Light-ColPali (Ma et al., 2025) (trained token merging) on ColPali and ColQwen2. Subscripts denote % retention of the corresponding full model. We exclude a Jina v4 + Light-Merging baseline because Light-ColPali releases trained weights only for the ColPali and ColQwen2 backbones.

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 (Ours)	**No**	$\mathbf{0.83_{98.2}}$	$0.80_{93.9}$	$0.74_{86.3}$
ColQwen2	Light-ColQwen2†	Yes	$\mathbf{0.82_{99.7}}$	$\mathbf{0.81_{98.8}}$	$\mathbf{0.80_{97.5}}$
	SAP (Ours)	**No**	$0.86_{97.7}$	$0.82_{93.2}$	$0.75_{85.6}$

† Results cited from Light-ColPali (Ma et al., 2025).

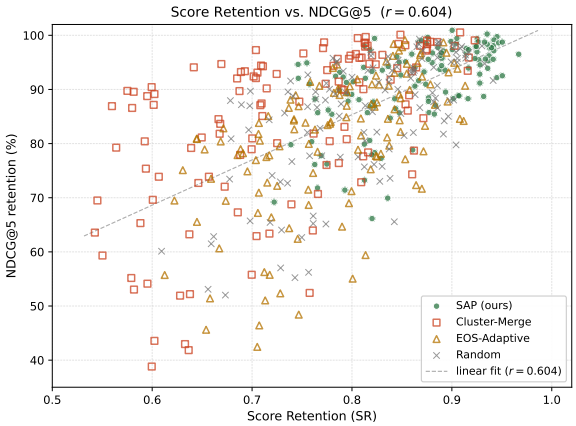

Figure 7: **Score Retention vs. NDCG@5 retention.** Each point is one (model, dataset, method, γ) configuration. SAP (solid markers) consistently occupies the high-fidelity region; baselines (hollow markers) exhibit substantially more scatter. The moderate correlation ($r \approx 0.60$) reflects that SR measures per-pair fidelity while NDCG measures corpus-level ranking; the two are complementary, not interchangeable.