أكاديمية كاوست
KAUST ACADEMY

جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

Generative Late-Interaction Embeddings For Visual Document Retrieval

Talal Aloushan, Rose Khairoalsendi, Jana Shatta,
Mohammed Alhassan, Leen Alrehaili, Mohamed Eltahir,
Tanveer Hussain

tlal.aloushan@kaust.edu.sa, rouz.alsindi@kaust.edu.sa, Jana.shata@kaust.edu.sa,
mohammed.alhassan.1@kaust.edu.sa, leen.alrehaili@kaust.edu.sa, mohamed.hamid@kaust.edu.sa,
hussaint@edgehill.ac.uk

Abstract

Late-interaction retrieval is the state of the art for visual document search, but it pays for its accuracy in storage: ColPali represents one page as 1,030 patch vectors, about 258 KB per page, two orders of magnitude more than a single-vector index. Existing post-hoc remedies pool, prune, or quantize the stored vectors, and none operates below roughly one hundred vectors per page; methods that reach smaller budgets retrain the encoder. We first explain this shared limit. Page token embeddings concentrate near a low-dimensional manifold on the unit sphere, and a simple lemma bounds the MaxSim scoring error of any code by its covering radius, predicting that sampling-based codes are near-optimal above the covering demand of the manifold and collapse below it, at roughly sixteen vectors per page for the measured dimension. We then introduce *generative late-interaction embeddings* (GLIE): a spherically anchored code that starts exactly at normalized clustering, is refined by a network that reads the page, and regenerates fine patch vectors on demand for reranking a shortlist, all post hoc on a frozen encoder. On the ten ViDoRe v1 subsets, GLIE retains 80% of uncompressed nDCG@5 while storing 258× less (4 vectors, 1.0 KB per page), 91% at 64×, and 98% at 16×; its margin over the strongest training-free baseline, chosen per subset and budget, concentrates below the predicted transition (+0.069 mean nDCG@5 at $k \leq 8$ versus +0.020 above, a 3.5× step), and the same transition reappears when the codec is transferred zero-shot to unseen corpora. Fitting the code for all ten corpora and six budgets takes 56 GPU-minutes total with 415K trainable parameters, against roughly 72 GPU-hours per budget for a fine-tuned competitor, which GLIE also outperforms at five of six budgets under matched training source and budget.

Date: August 24, 2026
Code: https://github.com/mohammad2012191/GLIE

Contents

1 Introduction **2**

2 Related Work **3**

3 The Geometry of Page Token Clouds **4**

3.1 Setup . . . 4

3.2 Measurements: a low-dimensional manifold on the unit sphere . . . 4

3.3 Theory: a covering bound, and a prediction . . . 5

4 Generative Late-Interaction Embeddings **6**

4.1 Spherical anchoring . . . 6

4.2 Zero-initialized refinement that reads the page . . . 6

4.3 Anchored generative read-out . . . 6

4.4 Training . . . 7

4.5 Cost, elasticity, and transfer . . . 8

5 Experiments **8**

5.1 Setup . . . 8

5.2 Main results: the predicted transition . . . 9

5.3 Zero-shot transfer across corpora . . . 10

5.4 Comparison to a fine-tuned competitor . . . 11

5.5 Efficiency and elasticity . . . 11

5.6 Ablations . . . 12

6 Discussion **12**

7 Limitations and Future Work **13**

8 Conclusion **13**

References **14**

1 Introduction

Retrieval over visual documents has converged on late interaction. Instead of collapsing a page into one vector, models such as ColPali [1] embed every image patch separately and score a query against a page with the MaxSim operator, the sum over query tokens of the maximum inner product with any page vector. This design is what makes visual retrieval work: it preserves local evidence such as a table cell, a plot axis, or a phrase in a figure caption, and multi-vector representations are provably more expressive than any single vector of comparable size [2].

The price is storage. ColPali stores 1,030 patch vectors of dimension 128 per page, about 258 KB in bfloat16; one million pages cost a quarter terabyte of embeddings before any index structure is added. The cost is paid at rest, in memory during search, and in transfer at query time, and the community now names index footprint among the paradigm's central open problems [3, 4]. The dominant remedies compress the stored set directly: pooling merges similar vectors [5], pruning keeps a salient subset [6], merging studies combine both [7, 8], and engines quantize each vector [9]. These methods share a premise, that the compressed representation is a subset or local average of the encoder's vectors, and they share a limit: none reports operating below roughly one hundred vectors per page, and methods that do reach small budgets retrain the encoder [10–12].

This paper starts from a different question: *what is the stored object, geometrically?* Measuring ColPali page embeddings, we find the 1,030 vectors of a page concentrate near a curved manifold of intrinsic dimension around six, living on the unit sphere of the 128-dimensional embedding space (Section 3). From this we prove an elementary lemma bounding the MaxSim error of any k-vector code by its covering radius. Combined with classical quantization theory [13, 14], the lemma makes a testable prediction, stated in Section 3 before any result: clustering-based codes are near-optimal while the budget exceeds the covering demand of the manifold and collapse below it, at roughly sixteen vectors per page for the measured dimension. The regime where storage savings are largest is exactly the regime where every sampling-based method must fail.

The same geometry yields the method. If a page is a low-dimensional surface on the sphere, the honest compressed object describes the surface rather than sampling it. *Generative late-interaction embeddings* (GLIE) build such a code post hoc from a frozen encoder: per-page cluster centroids are projected back onto the unit sphere, where the encoder actually places its outputs (a free correction worth up to +0.10 nDCG@5 on its own); a zero-initialized network refines the anchors while reading the full token set, so the code starts exactly at normalized clustering and can only improve; and a shared decoder regenerates approximate fine vectors for the top candidates of a first stage, so a query is answered by cheap MaxSim over k vectors per page followed by exact-style reranking of a shortlist. The decoder emits each stored vector unchanged among its outputs, so regeneration can add evidence but structurally cannot destroy it.

We evaluate on all ten ViDoRe v1 subsets against the strongest training-free baseline chosen *per subset and budget* among normalized clustering, semantic merging, and token pooling, plus a fine-tuned Light-ColPali baseline reproduced at matched training budget. Our contributions:

- **Geometry and a verified prediction.** We measure the geometry of page token clouds, prove a covering-radius bound on MaxSim error, and verify its central prediction: the achievable margin over sampling-based codes concentrates below $k \approx 16$ (+0.069 mean nDCG@5 at $k \leq 8$ versus +0.020 above, a 3.5× step), a transition that replicates across two refiner architectures, ten corpora, and a zero-shot transfer setting where it is sharpest (Sections 3, 5).
- **A frozen-encoder generative codec.** Spherical anchoring, zero-initialized page-reading refinement, and an anchored generative decoder with count-proportional structure, trained by operator-matched distillation with leak-free negative mining (Section 4).

- **State of the art below the transition, at negligible cost.** GLIE retains 80% of uncompressed nDCG@5 at 258× compression, 91% at 64×, and 98% at 16×; fitting all ten corpora at six budgets takes 56 GPU-minutes total with 415K trainable parameters (0.014% of the backbone), and the codec transfers zero-shot to unseen corpora, beating a LoRA-fine-tuned Light-ColPali at five of six budgets under matched source and budget (Section 5).

Everything in this paper operates on a frozen encoder. We view this constraint as the practically important one: it is the setting of every team that builds on a public checkpoint, and the setting in which the strongest published baselines are training-free.

2 Related Work

Late-interaction retrieval.. ColBERT introduced token-level document representations scored with MaxSim [15]; ColBERTv2 compressed the index with centroid-plus-residual storage [16]; ColPali transferred the paradigm to visual documents and introduced the ViDoRe benchmark, on which late interaction dominates single-vector alternatives [1]; Video-ColBERT extends it to video and names storage as its principal drawback [3]. Jayaram proves multi-vector embeddings strictly more expressive than single vectors of comparable dimension [2] — the structure is worth preserving, which cautions against collapsing pages into one vector. The gap in this line: expressiveness is bought with storage, and no account exists of *when* that storage can be compressed safely.

Post-hoc reduction of the stored set.. Token pooling merges document vectors by hierarchical clustering, retaining about 97% of quality at pool factor 4 [5]. For visual documents, Light-ColPali finds that merging post-projector embeddings retains 94.6% at roughly 36× reduction, its most aggressive published point being about sixteen vectors per page, and that pruning is ill-suited to the task [7]; anchor pruning [6] and prune-then-merge [8] extend the near-lossless range. All store a subset or local average of the encoder's vectors, none operates below its published floor, and none explains why they are strong at moderate budgets or why they fail together below them. Our analysis supplies the explanation, and our method targets precisely the budgets below the shared floor.

Small budgets via retraining.. ConstBERT learns a projection to a constant number of document vectors [10]; CRISP trains clusterability into the representation [11]; MetaEmbed trains nested meta-tokens serving budgets 1–64 at test time [12]; two-dimensional Matryoshka training makes visual document embeddings elastic [17]. Each requires training or finetuning the encoder, and each fixes its budgets at training time; changing the budget of a deployed index forces a re-encode of the corpus. Our setting is complementary: a frozen public checkpoint, cached embeddings, and a per-budget fit measured in GPU-minutes.

Orthogonal axes.. PLAID [9], EMVB [18], and HPC-ColPali [19] shrink bits per vector while keeping every vector; Matryoshka learning shrinks the dimension axis [20, 21]; MUVERA sketches multi-vector scoring for candidate generation but keeps full vectors at rest [22]. These compose multiplicatively with vector-count compression and are not competitors on the storage-at-rest axis.

Positioning.. The gaps above are one gap seen from four sides: token sets are treated as unstructured collections, with no theory of when compressing them must fail. We characterize the stored object geometrically, prove when sampling it suffices, verify the predicted failure point, and operate past it with a generative read-out no selection-based method offer. Table 1 summarizes.

Table 1 Positioning. "Effective < 16 vec/page" means published results at that budget without severe degradation. Adaptation cost per budget: Light-ColPali as reported by its authors; ours measured (Section 5.5).

Method	Frozen backbone	Effective < 16 vec	Budget-elastic (no re-encode)	Generative read-out	Adapt cost per budget
Token pooling [5]	✓	✗	✓	✗	0
Cluster merging [7]	✓	✗	✓	✗	0
Light-ColPali (trained) [7]	✗	✗	✗	✗	~72 GPU-h
MetaEmbed	✗	partial	✗	✗	full retrain
GLIE (ours)	✓	✓	✓	✓	~1 GPU-min

3 The Geometry of Page Token Clouds

We first establish what a late-interaction page representation is as a geometric object, then derive from it both an explanation of the existing compression landscape and a prediction that Section 5 tests.

3.1 Setup

A frozen encoder maps a page to token vectors $X = \{x_1, \ldots, x_N\} \subset \mathbb{R}^D$, with $N \approx 1{,}030$ and $D = 128$ for ColPali [1], and a query to $Q = \{q_1, \ldots, q_m\}$. The late-interaction score is

$$\mathrm{MaxSim}(Q, X) = \sum_{j=1}^{m} \max_i \langle q_j, x_i \rangle. \tag{1}$$

A compressed representation at budget k replaces X by a set $C = \{c_1, \ldots, c_k\}$, scored by the same operator. We write $r(X, C) = \max_i \min_j \|x_i - c_j\|$ for the covering radius of C with respect to X.

3.2 Measurements: a low-dimensional manifold on the unit sphere

Three measurements characterize per-page token clouds:

- **Intrinsic dimension.** The TwoNN estimator [23] gives a median of 6.4 against an ambient dimension of 128.
- **Effective linear dimension.** The participation ratio of the covariance spectrum [24] is 13.8, about twice the intrinsic dimension: the manifold is curved, bending through more linear directions than its dimensionality.
- **Quantization dimension.** Per-page k-means inertia decays smoothly as $k^{-\alpha}$ with $\alpha \approx 0.44$ and no elbow at any k: a continuum, not a cluster structure. Reading the slope as $2/d$ [13, 14, 25] gives an effective dimension of ≈ 4.5, consistent with TwoNN given pre-asymptotic k.

A fourth property is exact rather than estimated: ColPali L2-normalizes every output token, so $\|x_i\| = 1$ and the entire cloud lies on the unit sphere $\mathbb{S}^{D-1}$. The manifold is a *submanifold of the sphere*, and a faithful code inherits both constraints, low-dimensional in arrangement, and on the sphere. Clustering satisfies only the first: a k-means centroid is a Euclidean mean of unit vectors and lies strictly inside the ball, and under the inner products of Equation (1) its deficient norm attenuates the centroid against every query direction. The following identity quantifies the deficit exactly.

Proposition 1 (The centroid norm records its cluster's spread). *Let $x_1, \ldots, x_n$ be unit vectors with mean c. Then*

$$\frac{1}{n}\sum_{i=1}^{n}\|x_i - c\|^2 \;=\; 1 - \|c\|^2,$$

and consequently the k-means objective on the sphere equals $\sum_j n_j\,(1 - \|c_j\|^2)$ over clusters j with sizes n_j.

Proof. Expand $\|x_i - c\|^2 = 1 - 2\langle x_i, c\rangle + \|c\|^2$ and average; $\frac{1}{n}\sum_i \langle x_i, c\rangle = \|c\|^2$. □

Proposition 1 explains two facts we verify in Section 5: re-projecting centroids onto the sphere is a large *free* improvement (the attenuation is pure distortion under MaxSim), and the improvement shrinks as k grows (more centroids mean tighter clusters, whose means already sit near the sphere). It does *not* imply that the norm must be stored: we test that separately and report a null result (Section 5.6).

3.3 Theory: a covering bound, and a prediction

For a point set X, the map $u \mapsto \max_i \langle u, x_i\rangle$ is the support function of $\mathrm{conv}(X)$ [26, 27]; MaxSim evaluates it at the query tokens. This yields an elementary bound we have not found stated for late-interaction codes.

Lemma 1 (Covering radius bounds MaxSim error). *Let $Q = \{q_j\}_{j=1}^{m}$ with $\|q_j\| \le 1$ and let C be any k-point code for X. Then*

$$\big|\mathrm{MaxSim}(Q, X) - \mathrm{MaxSim}(Q, C)\big| \;\le\; m \cdot d_H(\mathrm{conv}(X), \mathrm{conv}(C)) \;\le\; m \cdot \max\{r(X, C),\, r(C, X)\},$$

where d_H is the Hausdorff distance. If every $c_j \in \mathrm{conv}(X)$, as for k-means centroids, the one-sided degradation satisfies $\mathrm{MaxSim}(Q, X) - \mathrm{MaxSim}(Q, C) \le m \cdot r(X, C)$.

Proof. Per query token the score difference is a difference of support functions, and the support map is an isometry from convex bodies with the Hausdorff metric into $C(\mathbb{S}^{D-1})$ with the supremum norm [27]; bound the Hausdorff distance by the covering radii via Cauchy–Schwarz and sum over the m tokens. □

The covering number of a compact d-dimensional set scales as $N(\epsilon) \sim \epsilon^{-d}$ [28, 29], and the covering radius of the best k-point code as $k^{-1/d}$. With $d \approx 6$, this has two consequences. Above the covering demand, a modest number of centroids covers the cloud at small radius, which is why training-free clustering and pooling are strong incumbents [5, 7], to our knowledge the first explanation of that strength. Below it, the code transitions from covering the surface to leaving regions unrepresented, and any query token pointing at an uncovered region loses its evidence entirely.

The prediction.. Sampling-based codes should track the uncompressed ceiling while k exceeds the covering demand and fall steeply below roughly $k \approx 16$ for the measured dimension; learned compression can pay only below that transition, because above it there is nothing left to recover. Section 5 tests this on ten corpora, on our own method (whose contribution should vanish above the transition), and in a transfer setting (where a codec with no corpus-specific information should show the transition most cleanly). The full-benchmark sweep confirms the first part directly: normalized clustering trails the ceiling by only 0.021 nDCG@5 at $k = 64$ and 0.045 at $k = 32$, but by 0.111 at $k = 16$, 0.194 at $k = 8$, 0.251 at $k = 4$, and 0.324 at $k = 2$ (Table 2).

Note

Takeaway A page representation is a curved ~6-dimensional manifold on the unit sphere, sampled at a thousand points. Codes must respect the sphere (Proposition 1); sampling the manifold is near-optimal down to its covering demand and provably degrades below it (Lemma 1); compression beyond that point requires regenerating the surface, not sampling it.

4 Generative Late-Interaction Embeddings

Section 3 concluded that below the covering demand the stored object must describe the surface rather than sample it, and that whatever is stored must live on the sphere. GLIE discharges both requirements with three components, each of which provably starts at, and can only improve on, the strongest training-free solution. The encoder is frozen throughout; everything happens post hoc on cached embeddings. Figure 1 shows the resulting query path and Figure 2 the fitting procedure.

4.1 Spherical anchoring

Per page, k-means centroids C_0 of the token set X are computed and projected onto the unit sphere, $u_j = c_j/\|c_j\|$. By Proposition 1 the projection removes a pure MaxSim distortion at zero cost; empirically it is worth +0.073 to +0.102 nDCG@5 at aggressive budgets on the full benchmark, shrinking to +0.031 at $k = 64$ exactly as the proposition predicts (Table 2, first two rows). Normalized clustering, not raw clustering, is therefore the honest baseline for everything that follows, and the practical one-line recommendation for any dot-product late-interaction system that clusters: *normalize your centroids*.

4.2 Zero-initialized refinement that reads the page

A shared cross-attention module refines the anchors against the full token set,

$$C \;=\; \mathrm{normalize}\big(U + \pi_\theta(\mathrm{Attn}(U,\, X,\, X))\big), \tag{2}$$

with the output projection π_θ initialized at zero, so the code begins *exactly* at normalized clustering and training can only move it where the objective improves. The correction reads X directly, so the code can encode scoring-relevant structure that centroids average away; no post-processing of C_0 alone could, since C_0 does not contain it. We also evaluated a more elaborate refiner with tangent-space updates and a dispersion-conditioned gate; across three corpora the two are statistically indistinguishable (mean difference 0.001 nDCG@5), so we adopt the simpler form and report the comparison as an ablation. The module is small (415K parameters against the 3B backbone) and runs once per page at indexing time.

4.3 Anchored generative read-out

A shared decoder g_ψ expands the k stored vectors back into N unit vectors for reranking. Its structure encodes three guarantees. First, *count-proportional slots*: cluster j owns n_j output slots ($\sum_j n_j = N$), so the decoder reproduces the page's actual cluster structure rather than a fixed learned layout. Second, an *exact anchor*: slot 0 of each cluster emits the refined vector verbatim, so the decoded set contains the code set and, because MaxSim is a maximum,

$$\mathrm{MaxSim}(Q, g_\psi(C)) \;\geq\; \mathrm{MaxSim}(Q, C) \quad \text{per query token:}$$

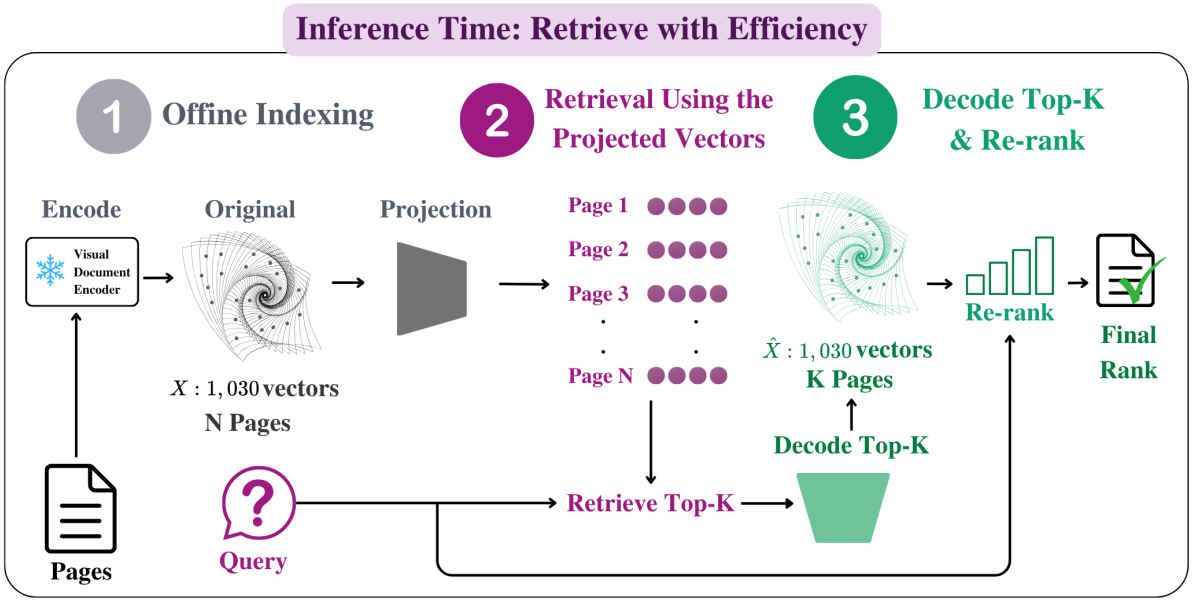


Figure 1 **Store few, search all, regenerate the finalists.** Indexing (once, offline) clusters a page's 1,031 patch embeddings, projects the centroids onto the unit sphere, refines them, and stores s vectors (1 KB per page at $s = 4$, against 258 KB). Search scores every page on those s vectors at pooling cost. Only the top-K shortlist is expanded back to ≈ 1030 vectors by the decoder and rescored exactly. The asymmetry is the method: cheap over everything, expensive over almost nothing.

regeneration can add evidence but structurally cannot destroy it, with no init-temperature tuning. Third, *bounded tangent displacement*: children leave their anchor only along the sphere, by at most a learned $\alpha \leq 0.75$, with a parameter-free Fourier encoding of each child's index within its cluster. At query time the first stage scores all pages on the k-vector codes at pooled cost, and the decoder regenerates vectors only for the top-L candidates ($L = 20$), which are rescored by full MaxSim; non-candidates keep their first-stage order.

4.4 Training

The objective is operator-matched: the code and the regenerated vectors are trained to reproduce the frozen encoder's *per-query-token* MaxSim values,

$$\mathcal{L}_{\text{tok}} = \frac{1}{m} \sum_{j=1}^{m} \Big(\max_{l} \langle q_j, \hat{x}_l \rangle - \max_{i} \langle q_j, x_i \rangle \Big)^2, \tag{3}$$

applied to both read-outs, plus a listwise KL over each query's candidate list, a one-sided penalty on negatives that overshoot the teacher (the failure mode that destroys rerank precision), a within-cluster Chamfer term between regenerated and true vectors, and a support-function matching term on fixed random directions. Reconstruction error is deliberately absent: it optimizes a quantity we found anti-correlated with retrieval.

Training uses only training-side queries (Figure 2): hard negatives are mined once from training queries against training documents under the frozen code, the decoder never conditions on a query, and evaluation queries touch nothing. Because the two read-outs are separate deployments (single-stage retrieval versus a two-stage cascade), each is checkpoint-selected on its own held-out metric; both checkpoints come from the same run and the shared-checkpoint numbers are retained in the released logs.

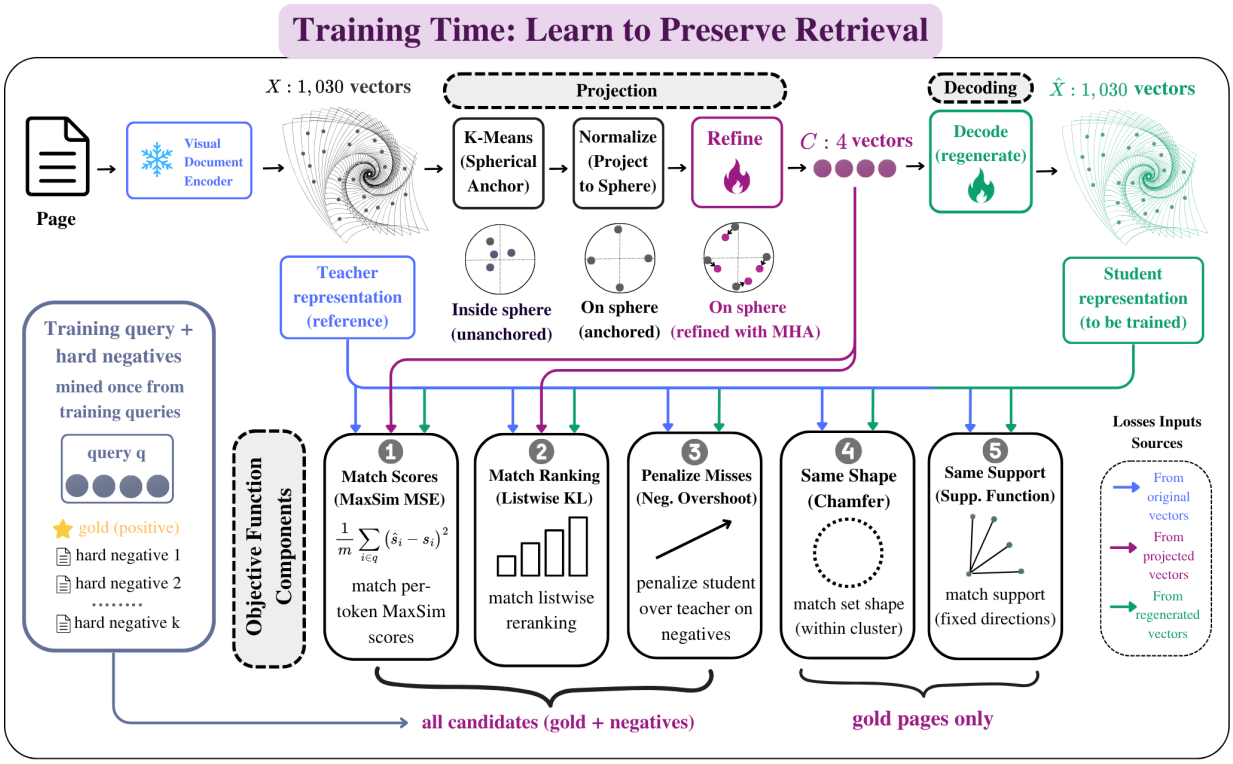


Figure 2 **Fitting the codec.** The encoder is frozen and acts as the teacher; only the refiner (415K parameters) and the decoder are trained. Operator-matched losses compare the student's per-token MaxSim to the teacher's on every candidate, gold and negative; set-geometry losses act on the gold page's regenerated vectors only. Hard negatives are mined once from training queries against training documents, and the decoder is query-free, so no evaluation query influences any stored representation.

4.5 Cost, elasticity, and transfer

At budget k the stored footprint is $k \times 128$ bfloat16 values plus two per-cluster scalars (a norm and a count, a measured 1.56% overhead that we retain for completeness despite the null ablation of Section 5.6), against 258 KB uncompressed: 1.0 KB per page at $k = 4$. The decoder is one shared module amortized over the corpus. Because the backbone is frozen, re-fitting for a new budget touches only cached embeddings — no corpus re-encode — and the refiner and decoder condition only on centroids, token sets, and cluster layouts, never on anything corpus-specific, so a codec fitted on one collection applies unchanged to another: Section 5.3 evaluates this zero-shot transfer directly.

5 Experiments

5.1 Setup

Benchmark and protocol.. We evaluate on all ten ViDoRe v1 subsets [1]. The corpus of each subset is its set of *unique* pages: rows are de-duplicated by image identity, and pages without queries remain in the index as distractors (two subsets repeat pages across many queries; indexing rows instead of pages understates even the uncompressed ceiling, e.g. 0.29 instead of 0.73 on TAT-DQA). Queries are split 80/20 into fit and held-out sets with a fixed seed shared by every method; all learned components use only the fit side, and we report single-relevant nDCG@5 on the held-out side. Held-out query counts are 100 (ArxivQA, DocVQA, InfoVQA), 56 (TabFQuAD),

Table 2 ViDoRe v1 (10 subsets), macro-averaged held-out nDCG@5 by stored vectors per page. All methods post hoc on frozen ColPali v1.3. Margin rows give the gain over the strongest training-free baseline chosen per subset and budget (macro; micro weights subsets by query count). Oracle reranks the same top-20 shortlist with true vectors and is a reference point, not an upper bound (see text).

Method	k=2	k=4	k=8	k=16	k=32	k=64
Raw k-means	0.451	0.494	0.565	0.653	0.746	0.796
Token pooling	0.547	0.571	0.613	0.646	0.695	0.755
Cluster merging	0.462	0.473	0.539	0.611	0.685	0.775
Normalized k-means	0.523	0.597	0.653	0.737	0.803	0.827
GLIE (stage 1)	0.578	0.641	0.693	0.755	0.815	0.834
GLIE (+ generative rerank)	**0.611**	**0.681**	**0.731**	**0.774**	**0.819**	**0.835**
shortlist oracle	0.765	0.786	0.827	0.837	0.844	0.850
% of uncompressed (0.848)	72%	80%	86%	91%	97%	98%
Margin over best baseline (macro)	+.058	+.077	+.073	+.035	+.016	+.008
Margin over best baseline (micro)	+.073	+.073	+.080	+.035	+.030	+.016

333 (TAT-DQA), and 20 each for Shift and the four synthetic subsets; we flag where the small subsets limit resolution.

Encoder and budgets.. ColPali v1.3, frozen, 1,031 patch vectors of dimension 128 per page (257.8 KB bfloat16). Budgets $k \in \{2, 4, 8, 16, 32, 64\}$, i.e. 516× down to 16× fewer stored vectors.

Baselines.. All post hoc on the same frozen encoder at identical stored budget: raw per-page k-means; *normalized* k-means (spherical anchoring alone); semantic cluster merging, the training-free stage of Light-ColPali [7]; and sequential token pooling [5]. Margins are always reported against the *strongest* training-free baseline chosen per subset *and* per budget — this discipline matters: token pooling wins several subsets at $k \leq 8$ and cluster merging wins others at $k \geq 16$, so any single fixed baseline would overstate us. We additionally reproduce Light-ColPali's fine-tuning stage (LoRA on the frozen checkpoint, their merge in the loop) at matched training budget (Section 5.4).

5.2 Main results: the predicted transition

Table 2 and Figures 3–4 carry the paper's central result. GLIE reaches 80% of uncompressed retrieval quality storing 1.0 KB per page and 91% at 4 KB, budgets at which no published post-hoc method reports results. The margin over the strongest training-free baseline is positive at every budget but is not uniform: it forms a plateau of ≈ 0.07 across $k \leq 8$, halves at $k = 16$, and decays to noise by $k = 64$ — the transition predicted, before these runs, at $k \approx 16$ by the covering analysis. The same step appears independently in the single-stage read-out ($0.035 \rightarrow 0.012$), replicates under a second refiner architecture (4.0× vs. 4.3× on the three development subsets), and reappears sharpest in the transfer setting below. Two honesty notes. First, four synthetic subsets have ceilings of 0.96–1.00 and are saturated at $k \geq 32$ (three contribute exactly zero margin there), so the small high-budget averages partly reflect saturation, not merely exhausted headroom. Second, in six of sixty cells the cascade exceeds the full-corpus ceiling; each corresponds to fewer than two queries changing one rank on a small subset, and a cascade whose first stage retains the gold page can legitimately beat full-corpus retrieval because the gold then competes against $L-1$ rather than $N-1$ pages. For the same reason the shortlist oracle is a reference, not a bound, and the decoder overtakes it in a handful of small-subset cells.

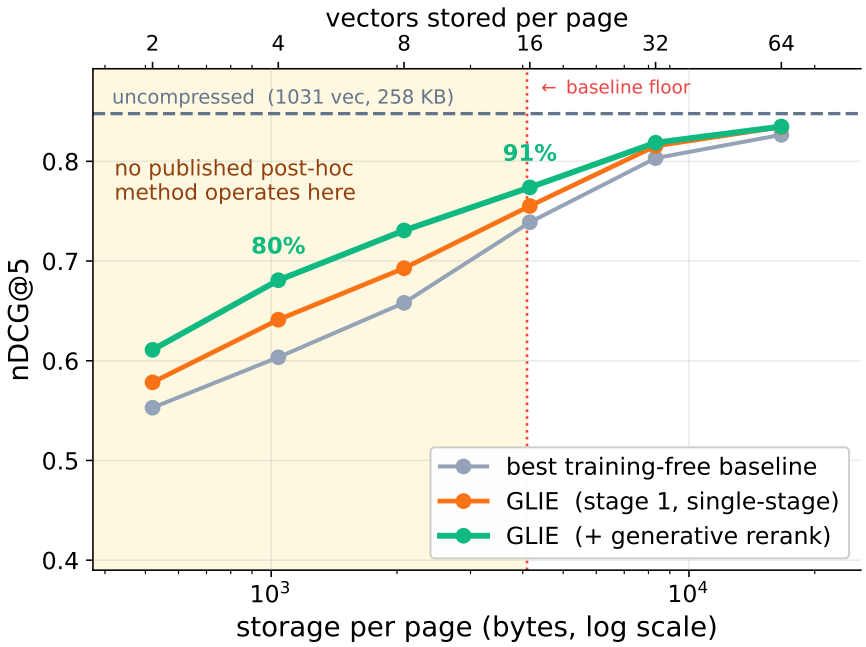


Figure 3 Retrieval quality against storage per page (log scale). GLIE retains 80% of uncompressed nDCG@5 at 258× compression and 91% at 64×. The shaded region marks budgets below the most aggressive published post-hoc operating point (≈ 16 vectors per page, Light-ColPali's merging factor 49 [7]).

Per-subset robustness: the rerank margin is positive on nine of ten subsets at $k = 2$, and is strongest and most consistent on TAT-DQA (+0.096/+0.089/+0.107/+0.052/+0.052/+0.028 across budgets), which at 333 held-out queries is the best-powered subset — the most reliable evidence is also the strongest. Where the decoder falls below stage 1 at all (10 of 60 cells, magnitudes ≤ 0.019), eight of the ten are at $k \geq 16$: above the covering demand the decoder has nothing to add, as predicted.

5.3 Zero-shot transfer across corpora

The codec conditions on nothing corpus-specific, so we evaluate it fitted on one collection and applied *frozen* to others: no target queries, no target fitting, only the target's own k-means and a forward pass.

Leave-one-out.. Fitting on two of the three development subsets and transferring to the third retains 40–73% of the in-corpus margin below the transition (mean decoder margins +0.037/+0.020/+0.034 at $k = 2/4/8$) and essentially nothing above it (−0.008/+0.009/+0.004): a 16× step, against 4.3× in-corpus. A codec with no corpus-specific information helps only where the theory says headroom exists — the cleanest of the transition's five appearances. Going from one source corpus to two improved the transferred margin in twelve of twelve budget–target cells (mean +0.006), supporting diversity, not volume, as the lever. Transfer benefit also tracks available headroom quantitatively: at $k = 8$, ranking the three targets by their baseline's gap to the ceiling ranks the transfer margins identically (DocVQA gap 0.195 → +0.070; ArxivQA 0.140 → +0.019; InfoVQA 0.119 → +0.014).

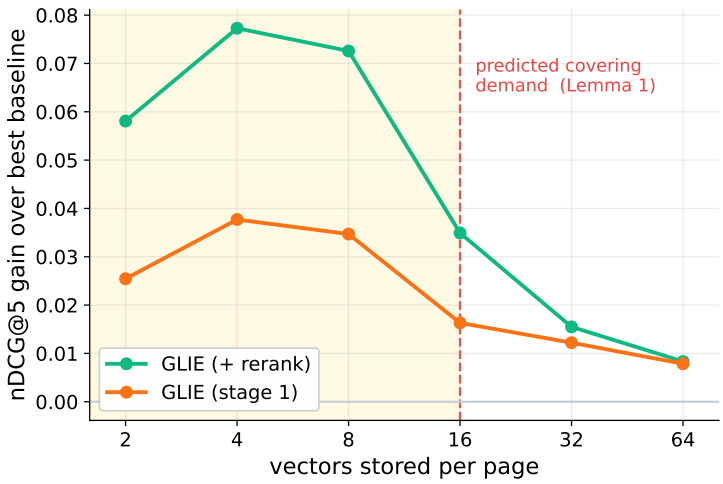


Figure 4 The margin over the strongest training-free baseline concentrates below the covering demand predicted in Section 3: mean +0.069 nDCG@5 at $k \leq 8$ against +0.020 at $k \geq 16$, a 3.5× step located between $k = 8$ and $k = 16$. Both read-outs show the step independently.

Table 3 Matched-source, matched-budget comparison, three development subsets, held-out nDCG@5. Both methods fit on the same 4,000 pages of the ColPali training collection and are applied zero-shot to the evaluation corpora. Light-ColPali is our single-GPU reproduction (batch 8 versus the authors' 256; pairwise ColBERT loss); its authors report ≈ 72 A100-hours per budget at full scale.

Method	k=2	k=4	k=8	k=16	k=32	k=64
Normalized k-means (free)	0.454	0.535	0.563	0.619	**0.666**	**0.698**
Light-ColPali (LoRA fine-tune)	0.488	0.518	0.562	**0.637**	0.650	0.668
GLIE (frozen backbone, transfer)	**0.506**	**0.568**	**0.594**	0.625	0.665	0.694

5.4 Comparison to a fine-tuned competitor

Table 3 makes three points. GLIE beats the fine-tuned baseline at five of six budgets (by up to +0.050) and loses at $k = 16$, while training 415K parameters for about one GPU-minute per budget against a 13.3M-parameter LoRA for about 1.5 GPU-hours. Second, at $k \geq 32$ *free normalized clustering matches or beats both learned methods* — in the zero-shot setting neither fine-tuning nor a learned codec buys anything above the covering demand, which is the theory's prediction landing on a competitor as well as on us. Third, at this matched budget the fine-tune does not consistently beat free clustering at any high budget; this measures the value of the frozen-backbone setting rather than diminishing Light-ColPali, which its authors train on 130K queries for 72 GPU-hours per budget.

5.5 Efficiency and elasticity

Table 4 summarizes measured cost. Fitting GLIE for *all ten corpora at all six budgets* took 56 GPU-minutes in total. Storage at $k = 4$ is 1,040 bytes per page (1,030 for vectors plus a 1.56% per-cluster sidecar) against 257.8 KB uncompressed: one million pages shrink from 258 GB to 1.0 GB. Because the backbone is frozen, changing the storage budget of a deployed index touches only cached embeddings; a fine-tuned method must re-encode the corpus. The shared decoder (≈1 MB) is amortized over the corpus, and query-time cost is pooled-baseline MaxSim plus $L = 20$ decoder forwards per query in the rerank mode.

Table 4 Measured cost. GLIE fitting times are wall-clock over cached embeddings (mean over 60 corpus–budget fits; medians in text); Light-ColPali reproduction at matched budget; published Light-ColPali as reported by its authors [7]. Storage includes our per-cluster sidecar.

Method	Adapt cost / budget	Trainable params	Backbone updated	Corpus re-encode on budget change
Token pooling / clustering	none	0	no	no
GLIE	0.93 GPU-min (median 0.43)	415K	no	no
Light-ColPali (repro, matched)	≈1.5 GPU-h	13.3M	yes (LoRA)	yes
Light-ColPali (published)	≈72 A100-h	13.3M	yes (LoRA)	yes

Table 5 Ablations (three development subsets unless noted). Each row changes one component of the full method.

Variant	Effect on nDCG@5
Raw instead of normalized anchors	−0.031 to −0.102 (Table 2, rows 1 vs. 4)
No refinement (normalized k-means)	−0.008 to −0.055 (Table 2, rows 4 vs. 5)
No generative rerank (stage 1 only)	−0.001 to −0.040 (Table 2, rows 5 vs. 6)
Gated tangent refiner instead of zero-init	±0.006, mean +0.001: statistically tied
Sidecar (norm and count scalars) removed	−0.006 mean at $k \leq 8$: null result, reported as such

5.6 Ablations

Three findings in Table 5 deserve comment. The largest single factor is spherical anchoring, a training-free projection (Proposition 1). The gated refiner's extra machinery — tangent-space updates, a dispersion-conditioned gate — does not pay: across three corpora it ties the zero-initialized form, so we ship the simpler one. And the sidecar ablation is a genuine null: although Proposition 1 shows the centroid norm *records* each cluster's spread, feeding it to the decoder is worth only +0.006, within noise — the decoder's count-proportional structure evidently already carries the usable part of that information. The norm explains why the sphere projection works; it is not why the decoder works.

6 Discussion

A prediction with five appearances.. The covering analysis made one falsifiable claim before the campaign: learned compression pays only below $k \approx 16$. The transition then appeared in the rerank margin (3.5× step on ten corpora), in the single-stage margin, under a second refiner architecture, on a fine-tuned competitor (Light-ColPali gains nothing over free clustering at $k \geq 32$ zero-shot), and, sharpest, under transfer (16× step), where the codec carries no corpus-specific information at all. We consider this stronger evidence than any single margin: the theory predicts where the method works *and* where it is useless, and both halves verify.

What each component is for.. The decomposition in Table 2 assigns the gains cleanly. Spherical anchoring is the largest single step and is free; it should be adopted by any clustering-based late-interaction system regardless of everything else in this paper. Refinement adds most where coverage is scarce (+0.055 at $k = 2$, +0.008 at $k = 64$). The generative rerank is the aggressive-budget specialist: +0.040 at $k = 4$, negligible above the transition, and the shortlist-oracle column bounds what better decoding could still recover (0.786 vs. 0.681 at $k = 4$), splitting the residual error into decode fidelity (recoverable) and shortlist recall (not, without a larger L).

Honest accounting as method.. Two design choices did the most to keep the numbers trustworthy. Comparing against the strongest training-free baseline per subset and budget cost us headline size — token pooling wins several low-budget subsets, cluster merging several high-budget ones — but removes the weak-baseline objection entirely. And the two null or negative results we report (the sidecar ablation; transferred stage-1 dipping below baseline at high budgets) are both predicted by the same analysis that predicts the wins, which is precisely what makes the analysis useful.

Practical recommendation.. For a deployed frozen index: normalize centroids always (free); below ≈16 vectors per page, add GLIE's refinement and rerank (one GPU-minute per corpus–budget, transfers zero-shot to new collections at zero fitting cost); above it, use normalized clustering and spend nothing.

7 Limitations and Future Work

Rate versus location.. Lemma 1 with $d \approx 6$ predicts the transition's *location*, which we verify; the asymptotic decay *rate* $k^{-1/d}$ at our small k is pre-asymptotic and we do not claim a quantitative rate match.

Single encoder.. All results use ColPali v1.3; intrinsic dimension and hence the transition point may differ across encoders.

Beyond pages.. The analysis predicts the largest gains where token sets are largest and most redundant. Video late interaction stores frames-by-patches per clip over a cloud with an additional smooth temporal axis; the same anchoring, refinement, and distillation pipeline applies unchanged, and storage is that setting's named principal drawback [3].

8 Conclusion

Late-interaction retrieval owes its accuracy to storing token-level vectors and its deployment cost to the same decision. We argued the way out is geometric. A page's thousand token vectors concentrate near a curved ~6-dimensional manifold on the unit sphere, and this one fact organizes the field: it explains why training-free clustering has been the strong incumbent, yields a free +0.03–0.10 nDCG@5 correction (normalize the centroids) that any dot-product late-interaction system can adopt today, and predicts through a covering-radius bound exactly where every sampling-based code must collapse — a transition at roughly sixteen vectors per page that we then observe five independent times, including on a fine-tuned competitor and in zero-shot transfer. Generative late-interaction embeddings operate past that transition: spherically anchored, zero-initialized refinement that can only improve on clustering, and an anchored decoder that regenerates the token cloud for reranking without ever being able to destroy stored evidence. On the full ViDoRe v1 benchmark this retains 80% of uncompressed quality at 258× less storage and 91% at 64×, beats the strongest training-free baseline chosen per subset and budget at every operating point, and beats a LoRA-fine-tuned competitor at five of six budgets under matched source and budget — with 415K trainable parameters, 56 GPU-minutes for the entire sweep, and zero-cost transfer to unseen corpora. The storage cost of late interaction is not a fixed tax on its accuracy: the redundancy that makes token sets expensive is structured, measurable, and, below the covering demand, learnable.

References

[1] Manuel Faysse et al. “ColPali: Efficient Document Retrieval with Vision Language Models”. In: *International Conference on Learning Representations (ICLR)*. arXiv:2407.01449. 2025.

[2] Rajesh Jayaram. *Multi-Vector Embeddings are Provably More Expressive than Single Vector Embeddings*. arXiv:2606.23475. 2026.

[3] Arun Reddy et al. “Video-ColBERT: Contextualized Late Interaction for Text-to-Video Retrieval”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. arXiv:2503.19009. 2025.

[4] Benjamin Clavié et al. *LIR: The First Workshop on Late Interaction and Multi-Vector Retrieval*. ECIR 2026 Workshop Proposal, arXiv:2511.00444. 2026.

[5] Benjamin Clavié, Antoine Chaffin, and Griffin Adams. *Reducing the Footprint of Multi-Vector Retrieval with Minimal Performance Impact via Token Pooling*. arXiv:2409.14683. 2024.

[6] Zhuchenyang Liu et al. *Training-Free Multi-Vector Compression for Visual Document Retrieval*. arXiv:2601.20107. 2026.

[7] Yubo Ma et al. “Towards Storage-Efficient Visual Document Retrieval: An Empirical Study on Reducing Patch-Level Embeddings”. In: *Findings of the Association for Computational Linguistics: ACL 2025*. arXiv:2506.04997. 2025.

[8] Yibo Yan et al. *Towards Efficient Multi-Vector Visual Document Retrieval via Prune-then-Merge Framework*. arXiv:2602.19549. 2026.

[9] Keshav Santhanam et al. “PLAID: An Efficient Engine for Late Interaction Retrieval”. In: *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM)*. arXiv:2205.09707. 2022.

[10] Sean MacAvaney, Antonio Mallia, and Nicola Tonellotto. “Efficient Constant-Space Multi-Vector Retrieval”. In: *Proceedings of the 47th European Conference on Information Retrieval (ECIR)*. arXiv:2504.01818. 2025.

[11] João Veneroso et al. *CRISP: Clustering Multi-Vector Representations for Denoising and Pruning*. arXiv:2505.11471. 2025.

[12] Zilin Xiao et al. “MetaEmbed: Scaling Multimodal Retrieval at Test-Time with Flexible Late Interaction”. In: *International Conference on Learning Representations (ICLR)*. arXiv:2509.18095, oral. 2026.

[13] Paul L. Zador. “Asymptotic Quantization Error of Continuous Signals and the Quantization Dimension”. In: *IEEE Transactions on Information Theory* 28.2 (1982), pp. 139–149.

[14] Siegfried Graf and Harald Luschgy. *Foundations of Quantization for Probability Distributions*. Vol. 1730. Lecture Notes in Mathematics. Springer, 2000.

[15] Omar Khattab and Matei Zaharia. “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*. 2020.

[16] Keshav Santhanam et al. “ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction”. In: *Proceedings of NAACL-HLT*. 2022.

[17] Haowen Xiang et al. *Towards Budget-Elastic Visual Document Retrieval via a 2D Multimodal Matryoshka Training Framework*. arXiv:2606.07654. 2026.

[18] Franco Maria Nardini, Cosimo Rulli, and Rossano Venturini. "Efficient Multi-Vector Dense Retrieval with Bit Vectors". In: *Proceedings of the 46th European Conference on Information Retrieval (ECIR)*. arXiv:2404.02805. 2024.

[19] Duong Bach. *Hierarchical Patch Compression for ColPali: Efficient Multi-Vector Document Retrieval with Dynamic Pruning and Quantization*. arXiv:2506.21601. 2025.

[20] Aditya Kusupati et al. "Matryoshka Representation Learning". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2022.

[21] Rohan Jha, Bo Wang, Michael Günther, et al. *Jina-ColBERT-v2: A General-Purpose Multilingual Late Interaction Retriever*. arXiv:2408.16672. 2024.

[22] Laxman Dhulipala et al. "MUVERA: Multi-Vector Retrieval via Fixed Dimensional Encodings". In: *Advances in Neural Information Processing Systems (NeurIPS)*. arXiv:2405.19504. 2024.

[23] Elena Facco et al. "Estimating the Intrinsic Dimension of Datasets by a Minimal Neighborhood Information". In: *Scientific Reports* 7 (2017), p. 12140.

[24] Peiran Gao et al. *A Theory of Multineuronal Dimensionality, Dynamics and Measurement*. bioRxiv 214262. 2017.

[25] Mikaela Iacobelli. "Asymptotic Quantization for Probability Measures on Riemannian Manifolds". In: *ESAIM: Control, Optimisation and Calculus of Variations* 22.3 (2016), pp. 770–785.

[26] R. Tyrrell Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[27] Rolf Schneider. *Convex Bodies: The Brunn-Minkowski Theory*. 2nd. Cambridge University Press, 2014.

[28] Andrey N. Kolmogorov and Vladimir M. Tikhomirov. "ϵ-Entropy and ϵ-Capacity of Sets in Functional Spaces". In: *American Mathematical Society Translations, Series 2* 17 (1961), pp. 277–364.

[29] Partha Niyogi, Stephen Smale, and Shmuel Weinberger. "Finding the Homology of Submanifolds with High Confidence from Random Samples". In: *Discrete & Computational Geometry* 39.1–3 (2008), pp. 419–441.