BUDGET-AWARE MULTI-VECTOR INDEX COMPRESSION FOR VISUAL DOCUMENT RETRIEVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual Document Retrieval (VDR) has become an essential capability for document intelligence systems, yet the growing storage cost of multi-vector indexes has become a critical barrier to large-scale deployment. Compressing VDR indexes introduces two fundamental challenges: 1) how to allocate a limited storage budget between preserving document-side matching opportunities and reducing representation cost; 2) how to maintain reliable late-interaction scoring when aggressive compression distorts the geometry of compressed representations. In this work, we propose **BaMIC** (**B**udget-**a**ware **M**ulti-Vector **I**ndex **C**ompression), a coverage-first compression method. We identify a structural asymmetry between compression dimensions: reducing vector cardinality directly removes potential matching opportunities, whereas numerical compression preserves matching coverage but may introduce geometric distortions in the extreme compression regime. Based on this observation, BaMIC prioritizes document-side vector coverage over per-vector precision and adapts representation rates to the available storage budget, and applies geometry-calibrated scoring, which requires no additional training or storage, to mitigate compression-induced radial bias. Extensive experiments across multiple VDR benchmarks, vision-language backbones, and compression budgets demonstrate that BaMIC achieves favorable storage-quality trade-offs, retaining 94.1% of full-precision Hit@1 on average with ColQwen3.5 at 64× compression and improving Hit@1 by up to 34.5 points over the strongest vector-count baseline under equal storage budgets.

1 INTRODUCTION

Visual Document Retrieval (VDR) is a key component of modern document intelligence systems, enabling retrieval from visually rich documents whose meaning is jointly conveyed by text, layout, tables, figures, and other visual elements (Qiao et al., 2025; Yu et al., 2025; Tanaka et al., 2025). Recent advances in vision-language models (Luo et al., 2024; Wang et al., 2024) have substantially strengthened this capability, making direct retrieval over document images increasingly effective. As document collections grow, however, the multi-vector representations used by modern VDR systems incur large index-storage costs, which have become a major obstacle to large-scale deployment. Storage-efficient visual document retrieval has therefore emerged as an important research problem.

Existing approaches to compressing multi-vector indexes follow two complementary directions. The first, which has dominated VDR, reduces vector cardinality: recent studies explore token pruning, patch selection, embedding merging, and combinations of pruning with hierarchical merging to retain fewer document vectors (Ma et al., 2025; Yan et al., 2025; 2026a). The second reduces the representation cost of each vector and has been studied more extensively in general vector search, with representative techniques including residual compression for multi-vector late interaction (Santhanam et al., 2022), Product Quantization (PQ) and Optimized Product Quantization (OPQ) (Jégou et al., 2011; Ge et al., 2013), and score-aware anisotropic quantization for maximum inner-product search (Guo et al., 2020). HPC-ColPali (Bach & Minh, 2025) proposes combining the two, pairing attention-guided pruning with vector quantization. Together, these works provide multiple ways to reduce index storage, but leave open how a VDR index should allocate a fixed byte budget across these compression dimensions. We therefore ask: as storage becomes increasingly constrained, what should a multi-vector VDR index preserve to maintain retrieval fidelity?

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

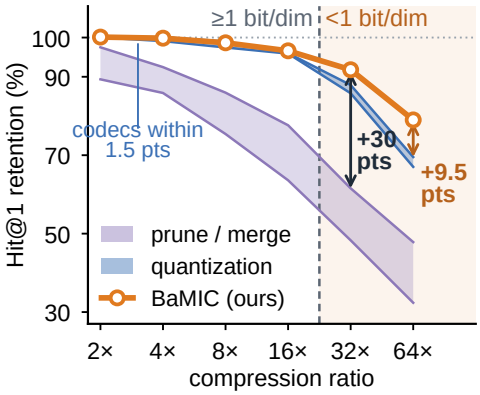


Figure 1: Where should the bytes go? Mean Hit@1 retention over the eight settings of Table 1 at equal storage.

The difficulty of this problem stems from two coupled challenges. *1) Preserving matching coverage.* Under a fixed storage budget, reducing the number of document vectors and reducing their numerical precision are not equivalent choices. Fewer vectors directly shrink the set of candidates available to future queries, whereas lower precision keeps every candidate but represents each of them more coarsely. As Figure 1 shows, at equal bytes pruning and merging lose retrieval quality far faster than quantization, so how the budget is split between vector cardinality and per-vector rate is a first-order compression decision. *2) Preserving scoring geometry.* Even when all candidates are retained, sufficiently aggressive numerical compression can distort the geometry of document vectors and change how they compete under late-interaction scoring. This becomes critical below one bit per dimension, where codebook representations introduce distortions qualitatively different from ordinary low-bit perturbations, and where uncalibrated quantizers fall visibly behind (Figure 1). Together, these challenges suggest that effective VDR index compression must be both budget-aware and regime-aware, preserving the right properties of the multi-vector representation as the available storage decreases.

To address these challenges, we propose **BaMIC**, a **B**udget-**a**ware **M**ulti-Vector **I**ndex **C**ompression method for VDR. Guided by this asymmetry, BaMIC follows a coverage-first policy: by default it keeps every patch vector and spends the storage budget on the per-vector representation rate; a joint allocation study confirms that this default dominates cardinality reduction at every budget. Above one bit per dimension, it applies training-free scalar quantization; below that rate, where scalar quantizers cannot operate, it switches to PQ or OPQ codebooks. Because sub-bit codebooks introduce token-dependent radial variation into decoded vectors, BaMIC scores every decoded vector on the unit sphere, removing this bias while leaving decoded directions unchanged. For PQ and OPQ, this sphere calibration is computed directly from the codes through normalized asymmetric distance computation (nADC), requiring neither retraining nor additional per-vector storage.

Our main contributions are:

- We uncover a structural asymmetry in multi-vector index compression: reducing vector cardinality can remove the winning match, whereas reduced precision keeps the MaxSim error bounded until sub-bit codebooks introduce a distinct radial score bias.
- We propose **BaMIC**, a coverage-first compression method that preserves retrieval fidelity across compression regimes by prioritizing coverage over precision, selecting the quantizer by rate, and removing the radial bias through nADC, a training-free and storage-free calibration.
- Experiments across multiple corpora, backbones and quantization families demonstrate clear advantages from preserving matching coverage under equal storage budgets and consistent gains from geometry-calibrated sub-bit scoring, in line with our analysis.

2 Related Work

2.1 Storage-Efficient Multi-Vector VDR

Multi-vector VDR models such as ColPali represent each page with patch-level embeddings and use late interaction to preserve fine-grained textual, visual, and layout information (Faysse et al., 2025; Cho et al., 2024), but these representations also lead to large document indexes. Recent work has therefore focused primarily on reducing document-side vector cardinality. Light-ColPali studies token pruning and embedding merging (Ma et al., 2025), DocPruner introduces adaptive patch-level pruning (Yan et al., 2025), MarginMerge performs coverage-aware merging (Mahdizadeh et al., 2026), and Prune-then-Merge combines pruning with hierarchical aggregation (Yan et al., 2026a). Other approaches construct compact representations more directly, such as layout-informed region

embeddings in ColParse (Yan et al., 2026b), while HPC-ColPali proposes combining attention-guided pruning with vector quantization (Bach & Minh, 2025). These methods demonstrate effective ways to reduce VDR index size, yet provide limited understanding of how a fixed storage budget should be divided between retaining more document-side matching opportunities and representing fewer vectors more accurately.

2.2 Numerical and Retrieval-Aware Vector Compression

A complementary line of work reduces the representation cost of individual vectors. Classical approaches such as Product Quantization, Optimized Product Quantization, and additive or composite quantization use compact codebooks to approximate high-dimensional vectors (Jégou et al., 2011; Ge et al., 2013; Babenko & Lempitsky, 2014; Zhang et al., 2014), while ColBERTv2 applies centroid-based residual compression to multi-vector late-interaction retrieval (Santhanam et al., 2022). Beyond reconstruction-oriented coding, retrieval-aware methods explicitly optimize quantities more closely related to downstream similarity search: Norm-Explicit Quantization encodes norms and directions with separate codebooks (Dai et al., 2020), ScaNN introduces anisotropic quantization for maximum inner-product search (Guo et al., 2020), and query-aware and recent low-bit methods further account for query distributions or inner-product preservation (Zhang et al., 2023; Zandieh et al., 2026). However, these techniques do not directly resolve how numerical compression should interact with matching coverage or preserve the scoring geometry of normalized multi-vector VDR under aggressive sub-bit compression, where decoded norms carry no relevance information and need not be encoded.

3 Problem Formulation and Structural Analysis

3.1 Budgeted Multi-Vector Compression

A multi-vector VDR index can be compressed by changing two properties of the document representation: the number of stored vectors and the representation cost of each vector. Given a document $V = \{\mathbf{v}_i\}_{i=1}^{N}$ and a query $Q = \{\mathbf{q}_t\}_{t=1}^{T}$, following normalized late-interaction retrievers, we assume $\|\mathbf{v}_i\|_2 = 1$ and $\|\mathbf{q}_t\|_2 \le 1$, and define the single-query-token response as

$$h_V(\mathbf{q}) = \max_i \mathbf{q}^\top \mathbf{v}_i, \qquad S(Q,V) = \sum_{t=1}^{T} h_V(\mathbf{q}_t). \tag{1}$$

For a fixed dimension D, the storage cost is $B \propto N\rho$, where N is the number of stored vectors and ρ is the rate in bits per dimension. A fixed budget can be allocated between preserving more matching opportunities and representing each retained vector more accurately.

To characterize these two choices, we write a compressed representation as $\widehat{V} = \{r_i \mathbf{u}_i\}_{i\in I}$, where I denotes retained vectors, $\mathbf{u}_i$ denotes the reconstructed direction with $\|\mathbf{u}_i\|_2 = 1$, and r_i denotes the reconstructed magnitude. This formulation separates three effects of compression: candidate removal through I, directional distortion through $\mathbf{u}_i$, and radial score distortion through r_i. We call an index *full-support* if no patch vector is removed at encoding time, i.e., $I = \{1, \dots, N\}$. Its *effective support*, the fraction of vectors per page that remain distinct after decoding, can still fall below one when tokens share identical codes.

For a fixed compression path $V \to V_I \to U_I \to \widehat{V}$, the MaxSim error can be decomposed as

$$h_V(\mathbf{q}) - h_{\widehat{V}}(\mathbf{q}) = \underbrace{h_V(\mathbf{q}) - h_{V_I}(\mathbf{q})}_{\text{matching opportunity loss}} + \underbrace{h_{V_I}(\mathbf{q}) - h_{U_I}(\mathbf{q})}_{\text{directional distortion}} + \underbrace{h_{U_I}(\mathbf{q}) - h_{\widehat{V}}(\mathbf{q})}_{\text{radial score distortion}}. \tag{2}$$

3.2 Structural Effects of Compression

The two compression axes introduce different risks. When all candidates are retained, bounded vector perturbation leads to bounded MaxSim error. Specifically, if $\max_i \|\mathbf{v}_i - \widehat{\mathbf{v}}_i\|_2 \le \epsilon$, then

$$|h_V(\mathbf{q}) - h_{\widehat{V}}(\mathbf{q})| \le \epsilon, \qquad \|\mathbf{q}\|_2 \le 1. \tag{3}$$

Therefore, reducing representation precision while preserving candidates introduces a controllable perturbation. In contrast, reducing vector cardinality removes potential future matches, and the

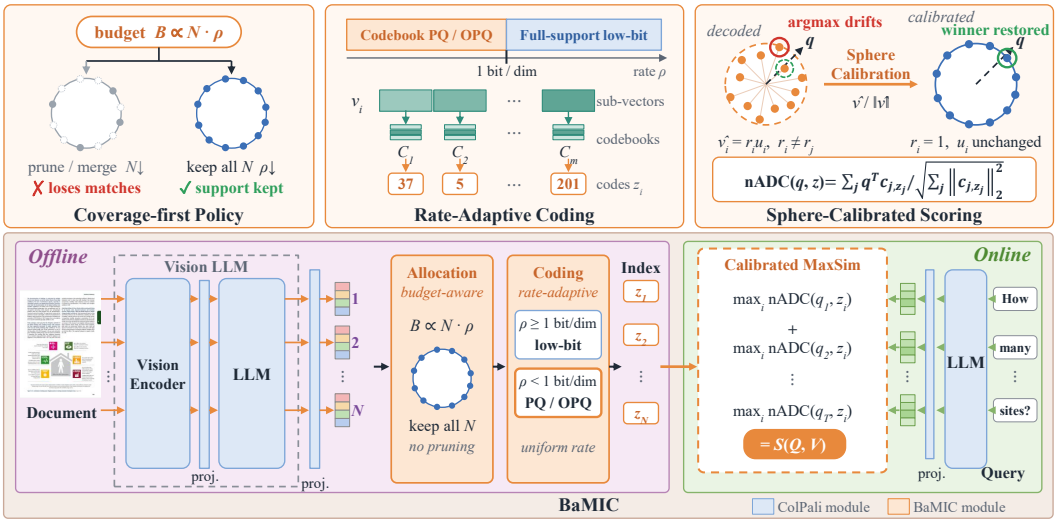


Figure 2: BaMIC: Budget-Aware Multi-Vector Index Compression.

retention ratio alone cannot guarantee retrieval fidelity because deleting a single winner may cause a large MaxSim change.

Preserving candidates is also insufficient when compression changes their relative scoring geometry. A reconstructed vector $r_i \mathbf{u}_i$ contributes $r_i \mathbf{q}^\top \mathbf{u}_i$ to MaxSim. A common scale $r_i = c$ leaves the winner unchanged, whereas candidate-dependent scales modify the decision boundary from $\mathbf{q}^\top(\mathbf{u}_i - \mathbf{u}_j) = 0$ to $\mathbf{q}^\top(r_i \mathbf{u}_i - r_j \mathbf{u}_j) = 0$. Thus, heterogeneous radial variation introduced by aggressive compression can bias MaxSim competition even when matching candidates are preserved.

4 BaMIC: Budget-Aware Multi-Vector Index Compression

4.1 Budget-Aware Compression Policy

BaMIC treats multi-vector compression as allocating the budget $B \propto N\rho$ (Section 3) between vector cardinality N and representation rate ρ. Different from directly removing vectors, BaMIC keeps the index full-support and reduces only the representation rate.

Summed over query tokens, Eq. 2 decomposes the score change of $\widehat{V}$ as

$$\Delta S = \Delta_{\mathrm{sup}} + \Delta_{\mathrm{dir}} + \Delta_{\mathrm{rad}}, \tag{4}$$

where Δ_{sup}, Δ_{dir} and Δ_{rad} collect the matching-opportunity loss, directional distortion and radial score distortion terms, respectively.

Based on this decomposition, BaMIC follows two principles: (1) avoid unnecessary support loss by maintaining vector coverage; and (2) correct radial score distortion when aggressive compression changes the retrieval geometry. We adopt full support as a budget-agnostic default; Appendix C examines interior allocations that trade coverage for precision.

4.2 Rate-Adaptive Representation Coding

BaMIC adopts different representation schemes according to the compression regime, separated by the rate ρ_0 of one bit per dimension. When $\rho \geq \rho_0$, full-support low-bit quantization is preferred because it reduces ρ while keeping N unchanged. Each coordinate of $\mathbf{v}_i$ is quantized with a symmetric uniform quantizer and a per-token scale, taken as the maximum absolute value at 8 and 4 bits and as the RMS value at 2 bits, where heavy-tailed coordinates would otherwise collapse to zero; at 1 bit, only the sign of each coordinate is stored (Appendix B.2).

When $\rho < \rho_0$, scalar quantization is no longer applicable and BaMIC switches to codebook-based representations such as PQ/OPQ. The compressed vector is represented as

$$\widehat{\mathbf{v}}_i = [\mathbf{c}_{1,z_{i,1}}; \mathbf{c}_{2,z_{i,2}}; \cdots; \mathbf{c}_{m,z_{i,m}}], \tag{5}$$

where $\mathbf{z}_i = (z_{i,1}, \ldots, z_{i,m})$ is the code of $\mathbf{v}_i$ and $\mathbf{c}_{j,z_{i,j}}$ is the selected codeword from the j-th sub-codebook; with 256 codewords per sub-codebook, each vector costs m bytes, i.e., $\rho = 8m/D$.

The representation strategy can therefore be summarized as a rate-dependent policy:

$$\mathcal{R}(\rho) = \begin{cases} \text{full-support low-bit quantization}, & \rho \geq \rho_0, \\ \text{PQ/OPQ coding}, & \rho < \rho_0. \end{cases} \tag{6}$$

Both branches build on standard quantizers. BaMIC determines which mechanism suits each storage regime revealed by the structural analysis, and in both regimes scores decoded vectors with the calibration described next.

4.3 Geometry-Calibrated Retrieval

As in Section 3, a decoded vector is written as $\widehat{\mathbf{v}}_i = r_i \mathbf{u}_i$, whose radial component r_i rescales its MaxSim contribution. To remove this unintended scaling factor, BaMIC performs sphere calibration:

$$\widetilde{\mathbf{v}}_i = \frac{\widehat{\mathbf{v}}_i}{\|\widehat{\mathbf{v}}_i\|_2}. \tag{7}$$

In the low-bit regime, this amounts to renormalizing decoded vectors. For PQ/OPQ representations, the normalization can be performed directly in the compressed domain. Dropping the token index, we write the code of a vector as $\mathbf{z} = (z_1, \ldots, z_m)$. Since the sub-codewords occupy disjoint blocks, $\|\widehat{\mathbf{v}}\|_2^2 = \sum_{j=1}^m \|\mathbf{c}_{j,z_j}\|_2^2$, and the calibrated asymmetric distance computation becomes

$$\mathrm{nADC}(\mathbf{q}, \mathbf{z}) = \frac{\sum_{j=1}^m \mathbf{q}_j^\top \mathbf{c}_{j,z_j}}{\sqrt{\sum_{j=1}^m \|\mathbf{c}_{j,z_j}\|_2^2}}, \tag{8}$$

where $\mathbf{q}_j$ is the j-th block of $\mathbf{q}$; for OPQ, $\mathbf{q}$ is first rotated by the learned orthogonal matrix. This formulation introduces no additional per-vector storage and requires only a table of squared codeword norms, built once per index.

5 Experiments

5.1 Experimental Setup

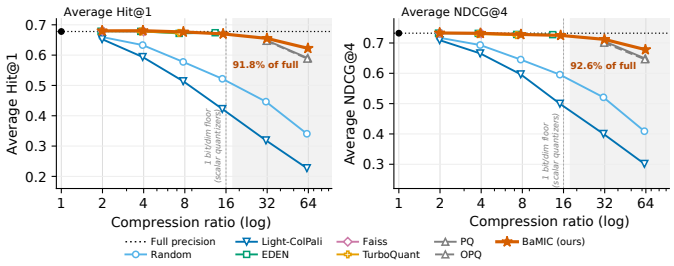
Datasets. We evaluate BaMIC on three benchmarks chosen to vary the properties that govern index compression: how many candidates a query competes over, and how concentrated the answer is within a page. **MMDocIR** (Dong et al., 2025) spans heterogeneous document types and layouts, giving the widest spread of per-page patch counts and embedding distributions, and hence the most varied codebook behaviour. We use it as the primary corpus. **MMLongBench-Doc** (Ma et al., 2024) averages 47.5 pages per document and requires aggregating evidence across pages, so a query can depend on several evidence pages, the regime in which discarding a potential winner is most costly. **MMESGBench** (Zhang et al., 2025) is the hardest case for aggressive compression: built from 157-page ESG reports whose evidence is sparse and localized, often confined to a single figure or table cell, retrieval there hinges on one or two patches surviving quantization rather than on a diffuse page-level signal. To study larger candidate pools, we additionally build two larger corpora: MMDocIR-20K, the full MMDocIR collection, and MMLong-34K, which adds distractor pages from MMDocIR-20K and MMESGBench to MMLongBench-Doc (Appendix B.1). These let us evaluate compression when each query competes against a substantially larger candidate pool.

Baselines. We evaluate on three backbones, ColPali, ColQwen2, and ColQwen3.5 (Faysse et al., 2025), and organize baselines into three families. Pruning methods discard patch embeddings judged unimportant: DocPruner (Yan et al., 2025), together with uniform random pruning as a query-agnostic reference. Merging methods replace groups of patches with fewer representatives: Light-ColPali (Ma et al., 2025) and MarginMerge (Mahdizadeh et al., 2026). Quantization methods retain

Table 1: Main results at matched CR bands. CR is the compression ratio over full-precision multi-vector storage. DocIR, ESG and Long denote MMDocIR, MMESGBench and MMLongBench-Doc. **Bold** marks the best value per column within each CR band.

CR	Method	ColQwen3.5 (D=320)						ColQwen2 (D=128)						ColPali (D=128)			
		DocIR		ESG		Long		Long		DocIR		ESG		DocIR(20K)		ESG	
		H@1	N@4	H@1	N@4	H@1	N@4	H@1	N@4	H@1	N@4	H@1	N@4	H@1	N@4	H@1	N@4
1×	Full	0.662	0.729	0.754	0.814	0.618	0.653	0.518	0.564	0.611	0.672	0.682	0.758	0.494	0.569	0.620	0.673
2×	Random	0.646	0.715	0.740	0.802	0.593	0.634	0.506	0.552	0.588	0.655	0.669	0.740	0.486	0.563	0.608	0.670
	MarginMerge-p	0.599	0.669	0.673	0.731	0.575	0.622	0.498	0.547	0.568	0.639	0.636	0.700	0.469	0.541	0.559	0.636
	MarginMerge-m	0.612	0.680	0.698	0.757	0.594	0.634	0.512	0.550	0.577	0.643	0.650	0.713	0.477	0.547	0.571	0.645
	Light-ColPali	0.627	0.701	0.740	0.796	0.591	0.631	0.508	0.553	0.588	0.654	0.671	0.731	0.485	0.559	0.600	0.661
	DocPruner	0.592	0.672	0.702	0.766	0.545	0.595	0.460	0.512	0.546	0.620	0.600	0.685	0.455	0.533	0.532	0.616
	HPC-ColPali	0.663	**0.730**	**0.757**	**0.815**	**0.619**	0.652	**0.521**	**0.565**	**0.613**	**0.672**	**0.685**	**0.759**	0.492	0.567	0.618	0.671
	EDEN	**0.665**	**0.730**	0.755	**0.815**	0.616	0.653	0.519	0.564	0.611	0.671	0.684	**0.759**	**0.493**	0.568	0.619	0.673
	Faiss SQ	**0.665**	**0.730**	0.755	**0.815**	**0.619**	**0.654**	0.518	0.564	0.611	**0.672**	0.683	**0.759**	0.491	0.568	**0.620**	**0.674**
	BaMIC	**0.665**	**0.730**	**0.757**	**0.815**	**0.619**	**0.654**	0.519	0.564	0.611	0.671	0.684	**0.759**	**0.493**	**0.569**	0.618	0.672
4×	Random	0.615	0.687	0.712	0.774	0.572	0.619	0.496	0.543	0.534	0.620	0.624	0.700	0.459	0.542	0.572	0.649
	MarginMerge-p	0.554	0.629	0.594	0.676	0.548	0.592	0.475	0.524	0.534	0.610	0.582	0.664	0.425	0.501	0.529	0.607
	MarginMerge-m	0.585	0.656	0.643	0.720	0.564	0.608	0.488	0.534	0.551	0.620	0.611	0.684	0.447	0.520	0.562	0.633
	Light-ColPali	0.549	0.639	0.669	0.745	0.564	0.613	0.464	0.522	0.540	0.621	0.613	0.684	0.450	0.538	0.577	0.646
	DocPruner	0.575	0.648	0.670	0.728	0.547	0.585	0.468	0.510	0.491	0.572	0.563	0.631	0.436	0.517	0.525	0.598
	HPC-ColPali	0.665	**0.731**	0.755	0.815	0.613	0.649	0.512	0.561	0.605	0.669	0.675	0.754	0.489	0.567	0.609	0.668
	EDEN	0.665	0.729	0.752	0.815	**0.619**	**0.651**	0.519	**0.564**	**0.614**	**0.672**	**0.688**	**0.759**	0.490	0.566	**0.617**	**0.673**
	TurboQuant	0.660	0.728	**0.764**	**0.816**	0.609	0.644	0.516	0.560	0.605	0.669	0.679	0.753	0.486	0.563	0.599	0.666
	Faiss SQ	0.661	0.729	0.755	0.814	0.618	**0.651**	**0.520**	0.563	0.603	0.671	0.687	0.755	0.486	0.565	0.609	0.668
	BaMIC	**0.667**	0.730	0.755	0.815	**0.619**	**0.651**	**0.520**	**0.564**	0.609	0.668	0.678	0.757	**0.494**	**0.569**	0.609	0.667
8×	Random	0.552	0.636	0.645	0.727	0.536	0.574	0.451	0.493	0.520	0.591	0.557	0.635	0.416	0.503	0.530	0.601
	MarginMerge-p	0.515	0.605	0.550	0.643	0.540	0.582	0.456	0.511	0.518	0.593	0.557	0.639	0.399	0.472	0.502	0.570
	MarginMerge-m	0.546	0.629	0.601	0.692	0.545	0.592	0.472	0.520	0.532	0.606	0.594	0.665	0.424	0.495	0.534	0.605
	Light-ColPali	0.478	0.569	0.558	0.654	0.506	0.566	0.445	0.490	0.463	0.545	0.518	0.603	0.424	0.502	0.520	0.606
	DocPruner	0.485	0.578	0.585	0.660	0.479	0.535	0.413	0.460	0.425	0.510	0.465	0.546	0.396	0.470	0.479	0.545
	HPC-ColPali	0.658	0.725	0.755	**0.815**	**0.611**	**0.649**	**0.518**	0.557	0.601	0.665	**0.683**	**0.752**	**0.485**	0.560	**0.601**	**0.667**
	EDEN	**0.661**	**0.728**	0.750	0.812	0.604	0.644	0.516	**0.562**	**0.605**	**0.670**	0.674	0.751	0.484	**0.562**	0.592	0.664
	TurboQuant	0.657	0.724	0.753	0.809	0.609	0.645	0.508	0.551	0.599	0.665	0.666	0.737	0.461	0.548	0.589	0.655
	BaMIC	0.658	0.726	**0.762**	0.814	0.609	0.645	0.513	0.555	**0.605**	0.667	0.679	0.750	0.477	0.552	0.596	0.664
16×	Random	0.492	0.585	0.582	0.659	0.492	0.543	0.420	0.462	0.433	0.517	0.471	0.559	0.387	0.463	0.470	0.533
	MarginMerge-p	0.445	0.533	0.479	0.565	0.487	0.548	0.425	0.484	0.455	0.539	0.476	0.557	0.350	0.422	0.411	0.498
	MarginMerge-m	0.485	0.568	0.549	0.623	0.520	0.575	0.446	0.498	0.475	0.553	0.516	0.596	0.386	0.452	0.454	0.545
	Light-ColPali	0.385	0.473	0.445	0.531	0.438	0.495	0.380	0.430	0.367	0.461	0.409	0.500	0.363	0.450	0.454	0.532
	DocPruner	0.398	0.469	0.501	0.565	0.420	0.468	0.356	0.409	0.359	0.433	0.391	0.455	0.321	0.388	0.400	0.476
	HPC-ColPali	**0.659**	0.721	0.749	0.807	**0.614**	**0.646**	0.492	0.539	0.576	0.646	0.666	0.730	0.455	0.531	0.558	0.638
	EDEN	**0.659**	**0.726**	**0.757**	**0.812**	0.606	0.645	0.493	**0.545**	**0.596**	**0.660**	0.656	0.733	0.442	0.532	0.568	**0.648**
	BaMIC	0.655	0.723	0.754	0.809	0.601	0.643	**0.496**	**0.545**	0.591	0.658	**0.668**	**0.735**	**0.460**	**0.542**	**0.578**	**0.648**
32×	Random	0.432	0.518	0.469	0.550	0.439	0.494	0.349	0.393	0.335	0.418	0.339	0.424	0.303	0.369	0.346	0.405
	MarginMerge-p	0.335	0.422	0.320	0.394	0.392	0.451	0.365	0.414	0.346	0.423	0.353	0.428	0.264	0.320	0.284	0.354
	MarginMerge-m	0.384	0.474	0.389	0.463	0.441	0.502	0.395	0.445	0.374	0.452	0.389	0.467	0.303	0.356	0.345	0.420
	Light-ColPali	0.295	0.382	0.321	0.408	0.340	0.411	0.301	0.355	0.279	0.356	0.276	0.354	0.292	0.354	0.292	0.369
	DocPruner	0.317	0.366	0.364	0.420	0.353	0.387	0.291	0.333	0.271	0.319	0.271	0.324	0.214	0.254	0.307	0.365
	ScaNN	0.615	0.683	0.734	0.788	0.564	0.602	0.360	0.397	0.396	0.487	0.450	0.521	0.247	0.318	0.341	0.413
	PQ	0.620	0.698	0.736	0.792	0.588	0.625	0.447	0.505	0.525	0.604	0.590	0.673	0.395	0.475	0.484	0.571
	OPQ	0.637	0.698	0.739	0.793	0.565	0.611	0.438	0.483	0.530	0.603	0.594	0.669	0.350	0.443	0.443	0.537
	BaMIC-PQ	0.626	0.702	0.741	0.799	0.598	0.635	**0.474**	**0.519**	**0.540**	0.614	0.606	0.691	**0.417**	**0.491**	**0.531**	0.606
	BaMIC-OPQ	**0.645**	**0.713**	**0.744**	**0.803**	**0.611**	**0.636**	0.466	0.506	0.536	**0.616**	**0.643**	**0.703**	0.407	**0.491**	0.527	**0.610**
64×	Random	0.303	0.392	0.389	0.448	0.331	0.388	0.287	0.331	0.268	0.334	0.278	0.342	0.251	0.307	0.254	0.310
	MarginMerge-p	0.225	0.287	0.200	0.261	0.275	0.320	0.279	0.331	0.275	0.340	0.241	0.290	0.184	0.223	0.139	0.201
	MarginMerge-m	0.284	0.358	0.279	0.356	0.322	0.385	0.319	0.369	0.314	0.381	0.278	0.339	0.215	0.258	0.204	0.264
	Light-ColPali	0.204	0.287	0.201	0.279	0.276	0.338	0.224	0.269	0.185	0.241	0.149	0.213	0.185	0.243	0.169	0.229
	DocPruner	0.214	0.249	0.269	0.295	0.244	0.272	0.206	0.233	0.172	0.205	0.191	0.223	0.120	0.150	0.194	0.234
	PQ	0.558	0.624	0.679	0.746	0.533	0.575	0.336	0.385	0.369	0.452	0.445	0.530	0.239	0.310	0.352	0.420
	OPQ	0.552	0.620	0.679	0.743	0.534	0.576	0.315	0.366	0.358	0.454	0.394	0.478	0.227	0.293	0.330	0.409
	BaMIC-PQ	0.585	0.665	0.715	0.775	0.568	0.595	0.376	0.424	0.431	0.507	**0.517**	**0.596**	0.286	0.358	**0.410**	0.494
	BaMIC-OPQ	**0.612**	**0.683**	**0.722**	**0.783**	**0.582**	**0.618**	**0.391**	**0.432**	**0.442**	**0.527**	0.516	0.591	**0.302**	**0.370**	0.403	**0.501**

every patch and reduce the bits spent on each. Above one bit per dimension we compare against the faiss scalar quantizer (Douze et al., 2026), EDEN (Vargaftik et al., 2022), and TurboQuant (Zandieh et al., 2026); below that rate, where scalar and rotation-based quantizers cannot operate, we compare



CR	Card.	Quant.	BaMIC	C–Q	B–C
	ΔH@1			Cohen's d	
4×	-7.9	+0.0	+0.3	0.26	0.27
8×	-11.8	-0.5	-0.2	0.34	0.33
16×	-17.5	-0.4	-0.8	0.42	0.42
32×	-27.6	-3.1	-1.8	0.59	0.61
64×	-38.7	-9.1	-4.9	0.68	0.75

Figure 3: Compression-axis analysis under equal storage budgets. Left: storage–retrieval frontier averaged over the three ColQwen3.5 corpora, with BaMIC shown as BaMIC-PQ in the sub-bit regime. Right: absolute Hit@1 loss (pp) of each family, pooled over the same corpora, with Cohen's d between families. Cardinality reduction averages Random, MarginMerge-p and MarginMerge-m; quantization and BaMIC average their PQ and OPQ variants in the sub-bit regime.

against product quantization (Jégou et al., 2011), optimized product quantization (Ge et al., 2013), and the anisotropic quantizer of ScaNN (Guo et al., 2020). HPC-ColPali (Bach & Minh, 2025) is evaluated through its released index, which stores every patch with PQ.

Evaluation Metrics. For retrieval effectiveness, we report Hit@1 and NDCG@4. Hit@1 measures whether the highest-ranked page is relevant, and is our primary metric because MaxSim compression failures manifest first as a change of winner. NDCG@4 further evaluates top-rank ordering quality by accounting for the positions of relevant pages within the top-4 results. To characterize the quality–efficiency trade-off, we report both storage-side and query-side cost metrics. On the storage side, we measure B/tok, the serialized bytes per visual token, and CR, the compression ratio over the fp16 index; both are read from the serialized index, so codebooks, rotation matrices and sidecars are included and amortized over the corpus. On the query side, we report p50, the median per-query scoring latency excluding query encoding, and Res., the peak GPU memory held during scoring.

5.2 Main Results

Figure 3 summarizes the compression frontier and Table 1 reports per-setting results. At every budget, reducing vector cardinality degrades retrieval far more than reducing per-vector precision: pruning and merging drift away from the full-precision frontier as compression increases, whereas quantization stays close to it over a much wider range. BaMIC follows the quantization frontier and extends it into the sub-bit regime, where a BaMIC variant achieves the best Hit@1 and NDCG@4 in all eight settings at both $32\times$ and $64\times$. Averaged over the eight settings, BaMIC-OPQ retains 91.8% of full-precision Hit@1 at $32\times$ and 79.0% at $64\times$, versus the best cardinality-reduction method in each setting, i.e., less than half of its loss. With ColQwen3.5, BaMIC-OPQ retains 94.1% at $64\times$ (92.4–95.8% across corpora); ColQwen2 and ColPali account for most of the remaining gap.

Figure 3 right reports the absolute Hit@1 loss of each family, pooled over the 2,735 ColQwen3.5 queries, hence smaller magnitudes than the retention above. Cardinality reduction loses 7.9 points at $4\times$ and 38.7 at $64\times$, whereas quantization loses at most 0.5 up to $16\times$ and 9.1 at $64\times$. BaMIC stays within 0.8 points of full precision up to $16\times$ and roughly halves the $64\times$ loss of quantization to 4.9 points. The effect size between cardinality reduction and quantization grows monotonically from 0.26 to 0.68 (0.27 to 0.75 for BaMIC), indicating that the gap is tied to the compression axis rather than to individual algorithms. A joint allocation grid at matched bytes (Appendix C) confirms that this gap is not an artifact of storing retained vectors in fp16: every interior allocation beats fp16 cardinality reduction and full support is best or tied above one bit per dimension, although at $63.9\times$ halving the vector count to double codebook capacity can be competitive.

The preferred strategy also depends on the storage regime. From $2\times$ to $8\times$, full-support quantizers such as EDEN and Faiss SQ stay within 2.8 Hit@1 points of full precision in every setting, while the strongest cardinality-reduction method loses 4.6–11.0 points at $8\times$; BaMIC performs on par with these codecs and achieves the best Hit@1 in four of the eight settings at $16\times$. In the sub-bit regime, codebooks become necessary but suffer from compression-induced scoring distortion. BaMIC improves Hit@1 over the corresponding uncalibrated codebook in all 32 comparisons,

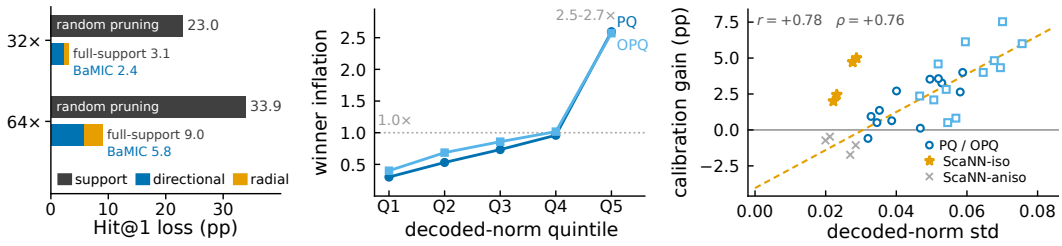


Figure 4: Compression mechanism analysis. (a) Hit@1 loss at equal bytes, pooled over the three ColQwen3.5 corpora: removing vectors costs far more than the full-support path, whose radial part calibration recovers. (b) Larger decoded norms disproportionately win MaxSim. (c) Norm dispersion predicts sphere-calibration gain, with ScaNN variants held out from the fit.

with gains growing from 0.5–4.7 (PQ) and 0.5–8.4 points (OPQ) at 32× to 2.7–7.2 and 4.3–12.2 points at 64×. For example, at 64× on ColQwen3.5, BaMIC-OPQ raises Hit@1 over OPQ from 0.552 to 0.612 on MMDocIR, from 0.679 to 0.722 on MMESGBench, and from 0.534 to 0.582 on MMLongBench-Doc at identical storage. Effective compression thus requires adapting to the regime: preserving matching opportunities through full-support quantization at low and moderate compression, and correcting scoring distortion in the sub-bit regime.

5.3 Ablation and Mechanistic Analysis

To understand why BaMIC remains effective under aggressive compression, we perform controlled interventions with ColQwen3.5 on MMDocIR-20K (Table 2). Eff. supp. is the fraction of patch vectors per page that remain distinct after decoding, Ang. $\sin\theta$ the mean angular error of decoded directions, and Norm std the absolute standard deviation of decoded norms before renormalization; indented rows change only how decoded norms enter scoring. Rescaling all decoded PQ vectors by a common factor of 2 doubles the norm std but leaves retention unchanged at 95.1%, confirming that only the token-dependent variation of norms affects MaxSim ranking. Removing this variation with nADC, while support and directions stay fixed, raises retention from 95.1% to 97.4% at 32× and from 86.2% to 91.5% at 64× without extra storage, whereas 1-bit scalar quantization, with smaller angular error and norm dispersion, retains 98.8%. Radial dispersion is thus specific to sub-bit codebooks.

Table 2: Controlled interventions on decoded geometry.

Interv.	Eff. supp.	Ang. $\sin\theta$	Norm std	H@1 ret.
Full prec.	1.000	0.000	0.000	100.0%
1-bit SQ	1.000	0.579	0.023	98.8%
PQ 32×	0.997	0.675	0.035	95.1%
+ scale×2	0.997	0.675	0.070	95.1%
+ nADC	0.997	0.675	0.000	97.4%
PQ 64×	0.946	0.759	0.053	86.2%
+ nADC	0.946	0.759	0.000	91.5%

Figure 4(a) decomposes the Hit@1 loss at equal bytes, pooling the 2,735 queries of the three ColQwen3.5 corpora. Removing vectors by random pruning costs 23.0 and 33.9 points at 32× and 64×, all of it matching-support loss. The full-support path loses only 3.1 and 9.0 points, split between directional distortion, which also absorbs code collisions, and a radial component of 0.7 and 3.2 points that calibration removes, leaving BaMIC-PQ 2.4 and 5.8 points below full precision. Preserving matching opportunities therefore dominates, while radial distortion is a smaller but recoverable part of the remaining loss.

Figure 4(b–c) shows how this distortion affects retrieval and when correcting it helps. Larger-norm decoded tokens are overrepresented among MaxSim winners, with the highest norm quintile reaching 2.5–2.7× the expected winner frequency, and decoded-norm dispersion correlates with calibration gain across PQ/OPQ interventions ($r = 0.78$, $\rho = 0.76$): ScaNN-iso gains, whereas ScaNN-aniso, which leaves little dispersion, does not. These results connect the mechanism to BaMIC's design: preserve matching opportunities first, then correct the radial bias that aggressive compression introduces.

Table 3: Storage, quality and query cost on MMLong-34K. Quality is retention (%) of full precision; Index and Res. in GiB; p50 is the median scoring latency (ms), excluding query encoding.

Method	Storage			Quality (%)		Query cost	
	CR↑	B/tok↓	Index↓	H@1↑	N@4↑	p50↓	Res.↓
Full precision	1.0	640.00	11.879	100.0	100.0	39.72	11.88
MarginMerge	61.8	10.36	0.192	65.5	65.3	0.49	0.19
Light-ColPali	61.8	10.36	0.192	50.4	53.4	0.50	0.19
Faiss SQ-4b	4.0	160.00	2.970	103.1	100.8	39.77	3.62
BaMIC-4b	4.0	162.00	3.007	99.7	99.8	39.91	3.66
EDEN-2b	7.3	88.00	1.633	102.3	100.8	39.81	2.17
TurboQuant-2b	7.3	88.00	1.633	99.7	98.9	39.72	2.18
BaMIC-2b	7.8	82.00	1.522	101.8	100.3	39.82	2.18
EDEN-1b	13.3	48.00	0.891	101.6	100.3	39.71	1.48
BaMIC-1b	15.2	42.00	0.780	100.5	98.8	39.84	1.50
PQ	63.9	10.02	0.186	88.0	89.6	39.41	0.74
OPQ	63.8	10.04	0.186	86.4	88.3	39.86	0.74
BaMIC-PQ	63.9	10.02	0.186	88.2	92.0	39.14	0.74
BaMIC-OPQ	63.8	10.04	0.186	100.0	97.0	39.68	0.74

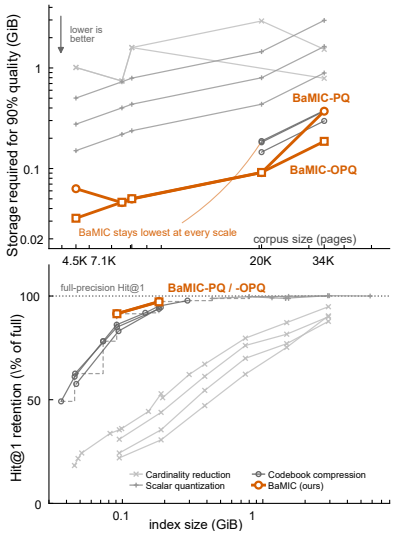


Figure 5: Top: storage required for 90% quality on each ColQwen3.5 corpus. Bottom: storage–quality frontier on MMDocIR-20K.

5.4 Storage and Efficiency Analysis

Table 3 compares compression strategies on MMLong-34K. At about 62–64×, cardinality reduction retains only 65.5% (MarginMerge) and 50.4% (Light-ColPali) of full-precision Hit@1, whereas full-support codebooks retain 86.4–100.0%. Above one bit per dimension, all scalar quantizers, including BaMIC, retain at least 99.7%; retention slightly above 100% corresponds to differences of about a dozen of the 848 queries. BaMIC-1b shrinks the index from 11.879 GiB to 0.780 GiB at 100.5% Hit@1. Resident memory follows the index size, falling from 11.88 GiB to 0.74–3.66 GiB. Latency, in contrast, depends on the number of scored vectors rather than index bytes: cardinality reduction is fastest (about 0.5 ms) at a large quality cost, while full-support methods stay close to full precision (39.1–39.9 ms), and calibration adds no measurable latency or memory.

Figure 5 extends this comparison to all five ColQwen3.5 corpora. On each, BaMIC requires the least storage to reach 90% of full-precision Hit@1 among the measured budgets, 0.03–0.19 GiB, versus 0.15–0.89 GiB for the best scalar quantizer and 0.74–2.9 GiB for cardinality reduction. The frontier on MMDocIR-20K shows the same ordering: at 64×, BaMIC retains 91.5% of full-precision Hit@1, compared with 83.0–86.1% for uncalibrated codebooks and below 37% for cardinality reduction.

6 Conclusion

In this work, we study multi-vector index compression for large-scale visual document retrieval under constrained storage budgets. We show that the two compression axes are not interchangeable: reducing vector cardinality removes matching opportunities that later queries may need, whereas reducing representation precision preserves matching coverage and keeps the MaxSim error bounded until sub-bit codebooks introduce a radial score bias. Building on these findings, we propose BaMIC, a coverage-first compression method that by default keeps every patch vector, selects the quantizer by rate, and removes the radial bias through nADC at no additional storage. Extensive experiments across diverse VDR benchmarks, vision-language backbones, and compression regimes show that BaMIC achieves favorable storage-quality trade-offs and remains effective under aggressive compression. Our study focuses on late-interaction multi-vector retrieval and primarily addresses compression-induced radial distortion; extending these findings to broader retrieval architectures and other forms of representation degradation provides promising directions for future work.

AI USE STATEMENT

In this work, we used generative AI tools for assisting in the writing of proofs. We have not used generative AI tools for developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, implementing methods, assisting with translation, cleaning and reformatting datasets, or interpreting results, and generating synthetic data sets and supporting qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for creating or modifying scientific figures or images, drafting parts of a research paper, summarizing or analyzing existing literature, and editing a research paper to improve readability. We have reviewed all AI-assisted work. All mathematical claims and proof strategies were formulated by the authors, every AI-assisted step in the written proofs was checked line by line by the authors, and all AI-assisted figures and text were checked against the reported results. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The source code for BaMIC, including the nADC implementation and all evaluation and statistical-testing scripts, is provided in the supplementary material. Proofs and assumptions for the theoretical results are given in Appendix A. Dataset construction and processing steps are described in Appendix B.1, and baseline implementation details in Appendix B.3.

REFERENCES

Artem Babenko and Victor S. Lempitsky. Additive quantization for extreme vector compression. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pp. 931–938. IEEE Computer Society, 2014.

Duong Bach and Pham Nhat Minh. Hierarchical patch compression for colpali: Efficient multi-vector document retrieval with dynamic pruning and quantization. In *Proceedings of the 17th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, IC3K 2025, Volume 1: KDIR, Marbella, Spain, October 22-24, 2025*, pp. 98–109. SCITEPRESS, 2025.

Jaemin Cho, Debanjan Mahata, Ozan Irsoy, Yujie He, and Mohit Bansal. M3docrag: Multi-modal retrieval is what you need for multi-page multi-document understanding. *CoRR*, abs/2411.04952, 2024.

Xinyan Dai, Xiao Yan, Kelvin Kai Wing Ng, Jiu Liu, and James Cheng. Norm-explicit quantization: Improving vector quantization for maximum inner product search. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 51–58. AAAI Press, 2020.

Kuicai Dong, Yujing Chang, Derrick-Goh-Xin Deik, Dexun Li, Ruiming Tang, and Yong Liu. Mmdocir: Benchmarking multimodal retrieval for long documents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 30971–31005. Association for Computational Linguistics, 2025.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. *IEEE Trans. Big Data*, 12(2):346–361, 2026.

Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. Colpali: Efficient document retrieval with vision language models. In *ICLR, Singapore, April 24-28*, 2025.

Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. Optimized product quantization for approximate nearest neighbor search. In *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pp. 2946–2953. IEEE Computer Society, 2013.

Ruiqi Guo, Philip Sun, Erik Lindgren, Quan Geng, David Simcha, Felix Chern, and Sanjiv Kumar. Accelerating large-scale inference with anisotropic vector quantization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3887–3896. PMLR, 2020.

Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(1):117–128, 2011.

Chuwei Luo, Yufan Shen, Zhaoqing Zhu, Qi Zheng, Zhi Yu, and Cong Yao. Layoutllm: Layout instruction tuning with large language models for document understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 15630–15640. IEEE, 2024.

Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, Pan Zhang, Liangming Pan, Yu-Gang Jiang, Jiaqi Wang, Yixin Cao, and Aixin Sun. MMLONGBENCH-DOC: benchmarking long-context document understanding with visualizations. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

Yubo Ma, Jinsong Li, Yuhang Zang, Xiaobao Wu, Xiaoyi Dong, Pan Zhang, Yuhang Cao, Haodong Duan, Jiaqi Wang, Yixin Cao, and Aixin Sun. Towards storage-efficient visual document retrieval: An empirical study on reducing patch-level embeddings. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, Findings of ACL, pp. 19568–19580. Association for Computational Linguistics, 2025.

Ailar Mahdizadeh, Aria Salari, Sohail Rajabi, Shahriar Mirabbasi, Panos Nasiopoulos, and Alireza Morsali. Coverage matters: Marginmerge for compressing multi-vector visual document retrievers. *CoRR*, abs/2608.02969, 2026.

Jingfen Qiao, Jia-Huei Ju, Xinyu Ma, Evangelos Kanoulas, and Andrew Yates. Reproducibility, replicability, and insights into visual document retrieval with late interaction. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2025, Padua, Italy, July 13-18, 2025*, pp. 3335–3345. ACM, 2025.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. Colbertv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pp. 3715–3734. Association for Computational Linguistics, 2022.

Ryota Tanaka, Taichi Iki, Taku Hasegawa, Kyosuke Nishida, Kuniko Saito, and Jun Suzuki. Vdocrag: Retrieval-augmented generation over visually-rich documents. In *CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 24827–24837, 2025.

Shay Vargaftik, Ran Ben Basat, Amit Portnoy, Gal Mendelson, Yaniv Ben-Itzhak, and Michael Mitzenmacher. EDEN: communication-efficient and robust distributed mean estimation for federated learning. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 21984–22014. PMLR, 2022.

Dongsheng Wang, Natraj Raman, Mathieu Sibue, Zhiqiang Ma, Petr Babkin, Simerjot Kaur, Yulong Pei, Armineh Nourbakhsh, and Xiaomo Liu. Docllm: A layout-aware generative language model for multimodal document understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 8529–8548. Association for Computational Linguistics, 2024.

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

Yibo Yan, Mingdong Ou, Yi Cao, Xin Zou, Jiahao Huo, Shuliang Liu, James T. Kwok, and Xuming Hu. Sculpting the vector space: Towards efficient multi-vector visual document retrieval via prune-then-merge framework. In *Findings of the Association for Computational Linguistics, ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 24883–24925. Association for Computational Linguistics, 2026a.

Yibo Yan, Mingdong Ou, Yi Cao, Xin Zou, Shuliang Liu, Jiahao Huo, Yu Huang, James T. Kwok, and Xuming Hu. Beyond the grid: Layout-informed multi-vector retrieval with parsed visual document representations. *CoRR*, abs/2603.01666, 2026b.

Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Junhao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, and Maosong Sun. Visrag: Vision-based retrieval-augmented generation on multi-modality documents. In *ICLR 2025, Singapore, April 24-28, 2025*, 2025.

Amir Zandieh, Majid Daliri, Majid Hadian, and Vahab Mirrokni. Turboquant: Online vector quantization with near-optimal distortion rate. In *International Conference on Learning Representations*, volume 2026, pp. 56418–56439, 2026.

Jin Zhang, Defu Lian, Haodi Zhang, Baoyun Wang, and Enhong Chen. Query-aware quantization for maximum inner product search. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pp. 4875–4883. AAAI Press, 2023.

Lei Zhang, Xin Zhou, Chaoyue He, Di Wang, Yi Wu, Hong Xu, Wei Liu, and Chunyan Miao. Mmesgbench: Pioneering multimodal understanding and complex reasoning benchmark for ESG tasks. In *Proceedings of the 33rd ACM International Conference on Multimedia, MM 2025, Dublin, Ireland, October 27-31, 2025*, pp. 12829–12836. ACM, 2025.

Ting Zhang, Chao Du, and Jingdong Wang. Composite quantization for approximate nearest neighbor search. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pp. 838–846. JMLR.org, 2014.

A Theoretical Analysis of Multi-Vector Compression

This appendix formalizes the three effects in Eq. 2. Throughout, a page is $V = \{\mathbf{v}_i\}_{i=1}^{N}$ with $\|\mathbf{v}_i\|_2 = 1$, a query is $Q = \{\mathbf{q}_t\}_{t=1}^{T}$ with $\|\mathbf{q}_t\|_2 \le 1$, $h_V(\mathbf{q}) = \max_i \mathbf{q}^\top \mathbf{v}_i$ is the support function of $\mathrm{conv}(V)$, and $S(Q,V) = \sum_t h_V(\mathbf{q}_t)$. A compressed page is $\widehat{V} = \{r_i \mathbf{u}_i\}_{i\in I}$ with $\|\mathbf{u}_i\|_2 = 1$ and $r_i > 0$. For a query with gold page V^+, the ranking margin is $\gamma(Q) = S(Q,V^+) - \max_{V^- \neq V^+} S(Q,V^-)$.

A.1 Vector Removal versus Full-Support Perturbation

Proposition A.1 (Asymmetry). *Let $\delta(I) = \max_{\mathbf{v}\in V} \min_{\mathbf{u}\in V_I} \|\mathbf{v}-\mathbf{u}\|_2$ be the coverage radius of a retained subset $V_I \subset V$. For any $\|\mathbf{q}\|_2 \le 1$:*

(i) $0 \le h_V(\mathbf{q}) - h_{V_I}(\mathbf{q}) \le \delta(I)$, and for every $|I| \le N-1$ there exist a page V of unit vectors and a unit query $\mathbf{q}$ such that $h_V(\mathbf{q}) - h_{V_I}(\mathbf{q}) = 2$;

(ii) if all vectors are retained and $\max_i \|\mathbf{v}_i - \widehat{\mathbf{v}}_i\|_2 \le \epsilon$, then $|h_V(\mathbf{q}) - h_{\widehat{V}}(\mathbf{q})| \le \epsilon$.

Proof. (i) The lower bound holds since $V_I \subset V$. Let $\mathbf{v}^\star$ attain $h_V(\mathbf{q})$ and $\mathbf{u}^\star \in V_I$ be its nearest retained vector; then $h_V(\mathbf{q}) - h_{V_I}(\mathbf{q}) \le \mathbf{q}^\top(\mathbf{v}^\star - \mathbf{u}^\star) \le \|\mathbf{v}^\star - \mathbf{u}^\star\|_2 \le \delta(I)$. For tightness, set $\mathbf{v}_k = \mathbf{q}$ for a removed index k and $\mathbf{v}_i = -\mathbf{q}$ otherwise. (ii) Since $|\max_i a_i - \max_i b_i| \le \max_i |a_i - b_i|$, we have $|h_V(\mathbf{q}) - h_{\widehat{V}}(\mathbf{q})| \le \max_i |\mathbf{q}^\top(\mathbf{v}_i - \widehat{\mathbf{v}}_i)| \le \epsilon$ by Cauchy–Schwarz.

Removal is thus controlled by coverage rather than by the number of retained vectors, and a single removed winner can reverse the score, whereas full-support perturbation is uniformly bounded. The bound propagates to the ranking.

Corollary A.2 (Ranking preservation). *If every page is compressed with* $\max_i \|\mathbf{v}_i - \widehat{\mathbf{v}}_i\|_2 \le \epsilon$*, the gold page remains ranked first whenever* $\gamma(Q) > 2T\epsilon$*.*

Proof. By Proposition A.1(ii), each page score changes by at most $T\epsilon$, so the gold score decreases and every other score increases by at most $T\epsilon$.

Only low-margin queries can therefore be affected by full-support compression, consistent with the failure analysis in Appendix G.

A.2 Radial Bias of Codebook Decoding

Lemma A.3 (Norm shrinkage of PQ decoding). *Let unit training vectors be split into disjoint blocks* $\mathbf{v}_n = [\mathbf{v}_n^{(1)}; \dots; \mathbf{v}_n^{(m)}]$ *with codes* z_{nj}*, and let each codeword* $\mathbf{c}_{j,k}$ *be the mean of the sub-vectors assigned to it. With* $E_{j,k}$ *and* $D_{j,k}$ *the mean of* $\|\mathbf{v}^{(j)}\|_2^2$ *and of* $\|\mathbf{v}^{(j)} - \mathbf{c}_{j,k}\|_2^2$ *over this cell,*

$$\|\mathbf{c}_{j,k}\|_2^2 = E_{j,k} - D_{j,k}, \qquad \frac{1}{M}\sum_n \|\widehat{\mathbf{v}}_n\|_2^2 = 1 - \frac{1}{M}\sum_n \|\mathbf{v}_n - \widehat{\mathbf{v}}_n\|_2^2. \tag{9}$$

The same holds for OPQ in the rotated space.

Proof. For any set with mean $\bar{\mathbf{x}}$, the mean of $\|\mathbf{x}\|_2^2$ equals $\|\bar{\mathbf{x}}\|_2^2$ plus the mean of $\|\mathbf{x}-\bar{\mathbf{x}}\|_2^2$. Summing over cells and blocks, using that disjoint blocks add in squared norm and that $\sum_j \|\mathbf{v}_n^{(j)}\|_2^2 = 1$, gives the second identity.

Decoded norms thus shrink on average by exactly the reconstruction error, and most for tokens in high-distortion cells, so the radii r_i reflect compression difficulty rather than relevance. A common radius leaves every ranking unchanged, since $\max_i c\,\mathbf{q}^\top\mathbf{u}_i = c\max_i \mathbf{q}^\top\mathbf{u}_i$ for $c > 0$; heterogeneous radii do not. For a query token, let $a_i = \mathbf{q}^\top\mathbf{u}_i$, $i^\star = \arg\max_i a_i$ and $\Delta = a_{i^\star} - \max_{j\neq i^\star} a_j > 0$, and measure heterogeneity by the relative dispersion $\eta = \max_i |r_i - \bar{r}|/\bar{r} < 1$ around a reference radius $\bar{r} > 0$.

Proposition A.4 (Winner stability). *If* $\Delta > 2\eta$*, then* $\arg\max_i r_i a_i = i^\star$ *for all radii with relative dispersion at most* η*. Conversely, if* $a_{i^\star} - a_j < \eta(|a_{i^\star}| + |a_j|)$ *for some* $j \neq i^\star$*, such radii exist with* $r_j a_j > r_{i^\star} a_{i^\star}$*.*

Proof. Over the admissible radii, the smallest value of $r_{i^\star}a_{i^\star}$ is $\bar{r}(a_{i^\star} - \eta|a_{i^\star}|)$ and the largest value of $r_j a_j$ is $\bar{r}(a_j + \eta|a_j|)$, so their difference is $\bar{r}[(a_{i^\star} - a_j) - \eta(|a_{i^\star}| + |a_j|)]$. Since $|a_i| \le 1$, it is at least $\bar{r}(\Delta - 2\eta) > 0$ in the first case and negative in the second.

The winner is therefore safe only when its directional margin exceeds the relative radial dispersion, which explains why larger decoded norms win MaxSim disproportionately (Figure 4b). Reconstruction-optimal radii do not remove this effect: for a fixed $\mathbf{u}_i$, $\|\mathbf{v}_i - r\mathbf{u}_i\|_2^2$ is minimized at $r = \mathbf{v}_i^\top\mathbf{u}_i = \cos\theta_i$, whose dispersion vanishes only if all tokens share the same angular error. This is why storing reconstruction-optimal norms degrades retrieval (Table 8).

A.3 Sphere Calibration

Proposition A.5 (Sphere calibration). *Let* $\Pi_{\mathbb{S}}(\mathbf{x}) = \mathbf{x}/\|\mathbf{x}\|_2$*. (i) For unit* $\mathbf{v}$ *and any* $\mathbf{x} \neq \mathbf{0}$*,* $\Pi_{\mathbb{S}}(\mathbf{x})$ *is the closest unit vector to* $\mathbf{x}$ *and* $\|\mathbf{v} - \Pi_{\mathbb{S}}(\mathbf{x})\|_2 \le 2\|\mathbf{v} - \mathbf{x}\|_2$*. (ii) For PQ with* $\widehat{\mathbf{v}} = [\mathbf{c}_{1,z_1}; \dots; \mathbf{c}_{m,z_m}]$*,*

$$\mathbf{q}^\top \Pi_{\mathbb{S}}(\widehat{\mathbf{v}}) = \frac{\sum_j \mathbf{q}_j^\top \mathbf{c}_{j,z_j}}{\sqrt{\sum_j \|\mathbf{c}_{j,z_j}\|_2^2}}, \tag{10}$$

where $\mathbf{q}_j$ *is the* j*-th block of* $\mathbf{q}$*, and the same holds for OPQ with* $\mathbf{q}$ *replaced by* $\mathbf{Rq}$ *for its orthogonal rotation* $\mathbf{R}$*.*

Table 4: Corpus statistics. Vec./page is the mean number of patch vectors per page.

Corpus	Encoder (D)	Pages	Patch vectors	Vec./page	Queries
Main benchmarks (Table 1)					
MMDocIR	ColQwen3.5 (320)	4,487	3,367,756	751	1,100
MMESGBench	ColQwen3.5 (320)	7,077	5,313,203	751	785
MMLongBench-Doc	ColQwen3.5 (320)	6,529	4,869,346	746	850
MMDocIR	ColQwen2 (128)	4,487	2,719,130	606	1,100
MMESGBench	ColQwen2 (128)	7,077	5,313,203[†]	751	785
MMLongBench-Doc	ColQwen2 (128)	6,529	4,889,714	749	850
MMDocIR-20K	ColPali (128)	20,395	21,027,245	1031	1,100
MMESGBench	ColPali (128)	7,077	7,296,387	1031	785
Large-scale corpora (scale, ablation and efficiency studies)					
MMDocIR-20K	ColQwen3.5 (320)	20,395	9,746,888	478	1,100
MMLong-34K	ColQwen3.5 (320)	34,001	19,929,437	586	848

[†]Identical to ColQwen3.5 by construction: every MMESGBench page exceeds the pixel budget of both processors, which yields the same patch grid (Appendix B.1). The embeddings themselves differ.

Proof. (i) For unit $\mathbf{y}$, $\|\mathbf{x}-\mathbf{y}\|_2^2 = \|\mathbf{x}\|_2^2 + 1 - 2\mathbf{x}^\top\mathbf{y}$ is minimized at $\mathbf{y} = \mathbf{x}/\|\mathbf{x}\|_2$, and $\|\mathbf{v} - \Pi_{\mathbb{S}}(\mathbf{x})\|_2 \le \|\mathbf{v}-\mathbf{x}\|_2 + |\|\mathbf{x}\|_2 - 1| \le 2\|\mathbf{v}-\mathbf{x}\|_2$. (ii) Disjoint blocks have zero cross inner products, so both the squared norm and the inner product decompose over blocks; an orthogonal rotation preserves both.

Calibration therefore sets $\eta = 0$, so the winner is decided by decoded directions alone, while retaining the guarantee of Corollary A.2 with 2ϵ in place of ϵ. For PQ and OPQ, it requires only a table of squared centroid norms built once per index. For additive quantizers with full-dimensional codewords, $\|\widehat{\mathbf{v}}\|_2^2$ contains cross terms $2\sum_{j<l}\mathbf{c}_{j,z_j}^\top\mathbf{c}_{l,z_l}$ that depend on the code combination, so the norm cannot be read from per-codeword tables.

B Implementation Details

B.1 Datasets and Corpus Construction

Table 4 summarizes all corpora. Each corpus consists of page-level patch embeddings produced by a frozen encoder and L2-normalized as released (norm standard deviation 4×10^{-7}).

Queries. Queries and gold page labels follow the original benchmarks. We discard questions without a valid evidence page: 232 of 1,082 in MMLongBench-Doc (228 without evidence, 4 with an invalid page index) and 148 of 933 in MMESGBench. A question is evaluated on a corpus only if at least one of its evidence pages is included.

Main benchmarks. For the ColQwen encoders, MMDocIR uses a 4,487-page subset comprising the 208 documents referenced by its 1,100 queries, each truncated to its first 50 pages (median 20). For ColPali, we use the full collection of 313 documents and 20,395 pages. ColPali encodes pages at a fixed resolution, yielding 1,031 vectors per page. The ColQwen models use dynamic resolution, so the count is determined by the image processor rather than by D. Both processors cap a page at 768 tokens (ColQwen2: 602,112 pixels, patch factor 14×2; ColQwen3.5: 786,432 pixels, patch factor 16×2). All MMESGBench pages exceed both caps and are resized to identical patch grids, so the two encoders produce the same vector count with different embeddings. MMDocIR pages fall below both caps and follow each encoder's native patch factor, while MMLongBench-Doc contains both cases.

Large-scale corpora. We construct two larger corpora with ColQwen3.5. MMDocIR-20K is the full MMDocIR collection; its lower average of 478 vectors per page, compared with 751 on the subset, stems from a different page rendering rather than page composition (474 vs. 481 vectors for pages within and beyond the first 50). MMLong-34K augments the 6,529 gold MMLongBench-Doc pages with 27,472 distractor pages from MMDocIR-20K and MMESGBench, checked for overlap with the gold documents, enlarging the candidate pool about fivefold. Documents are encoded up to

their first 200 pages, which excludes two questions and leaves 848. Within each corpus, all methods share the same embeddings.

B.2 BaMIC Implementation

Backbones and representations. We use three frozen late-interaction retrievers, ColQwen3.5 (D=320), ColQwen2 and ColPali (both D=128), without any fine-tuning. Patch embeddings are L2-normalized as released, and the full-precision reference stores them in fp16, i.e. 640 and 256 bytes per token for D=320 and D=128, respectively. All pages are scored exhaustively with MaxSim; no approximate nearest-neighbor structure is used.

Low-bit regime. For compression ratios of 2×, 4×, 8× and 16×, BaMIC retains every patch vector and applies a symmetric uniform scalar quantizer with 8, 4, 2 and 1 bits per dimension, respectively. Each token stores a single fp16 scale (2 bytes per token). At 8 and 4 bits, the scale is the per-token maximum absolute value, with 255 and 15 quantization levels. At 2 bits, we instead use the per-token RMS value $\|\mathbf{v}\|_2/\sqrt{D}$ with three levels $\{-s, 0, +s\}$: patch embeddings have a median crest factor (ratio of the maximum to the RMS magnitude) of 3.07, so a maximum-based scale would map most dimensions to zero. At 1 bit, we store the sign of each dimension with the scale $\overline{|v|}\sqrt{\pi/2}$. Decoded vectors are renormalized to unit norm, the same sphere calibration as in Section 4, matching the original embeddings; the per-token scale is kept to support decoding without renormalization and does not affect the ranking.

Sub-bit regime. For 32× and 64×, BaMIC uses PQ or OPQ from Faiss PQ with 8-bit sub-codebooks, so that m sub-quantizers cost m bytes per token. We set m=20 and m=10 for D=320, and m=8 and m=4 for D=128. Codebooks are trained without supervision on 200k patch vectors sampled from the indexed corpus with a fixed seed (13). Following the Faiss defaults, each sub-quantizer is trained with 25 k-means iterations on at most 65,536 points. OPQ learns a $D \times D$ rotation on a 65,536-vector subsample with 50 alternating iterations, initialized by a PQ fit with 40 k-means iterations and using 4 k-means iterations per subsequent round.

Hardware and measurement. All experiments run on a single NVIDIA RTX 6000D GPU (84 GB) with an Intel Xeon w7-3445 CPU, using PyTorch 2.10, Faiss 1.15.0 and Python 3.14; ScaNN runs in a separate Python 3.12 environment. Index size is measured from the serialized index, including codebooks, rotation matrices and side information. Latency is measured in a fresh process per configuration, after 20 warm-up queries, over five timed rounds with a single worker. We report a configuration only if its GPU clock drift stays within 10% and its round-to-round standard deviation within 5% of the mean.

B.3 Baselines

All methods are compared at the same serialized index size and operate on the same cached embeddings of a frozen encoder, without retraining; this training-free setting is shared by BaMIC. All pages are scored exhaustively with MaxSim. Vector-count baselines store the retained or merged vectors in fp16, with the retention ratio set so that the index matches the target budget; random selection keeps a uniformly sampled subset of patch vectors per page with a fixed seed.

For MarginMerge we run the official code and report its coverage-aware anchor selection (MarginMerge-p) and representative synthesis (MarginMerge-m) separately; since the learned synthesis module is not released, MarginMerge-m uses the training-free synthesis path of the official code, and query prototype banks are built from query sets disjoint from the evaluated corpus. Light-ColPali and DocPruner have no released code, so we reimplement them under the same protocol: Light-ColPali merges patches by agglomerative clustering (average linkage, cosine distance), and DocPruner applies per-document adaptive thresholding.

Faiss SQ, EDEN and TurboQuant are taken from Faiss 1.15.0 at the bit widths they support. For HPC-ColPali, we follow the official configuration of the released code, which stores all document patches with Faiss PQ and applies attention-guided pruning only to queries. We evaluate this index-side operator, vary only the number of sub-quantizers m to match each budget, and bypass its HNSW graph. Since the resulting index at 32× and 64× is identical to our PQ baseline with the same m, so

Table 5: Iso-byte allocation between vector count and precision with ColQwen3.5 on MMLong-34K (848 queries) and MMDocIR-20K (1,100 queries). Pruning keeps a uniform fraction of patch vectors per page; Kept × B/tok is constant within each block. Cells: Hit@1 retention (%) and, in parentheses, the Hit@1 difference (pp) to the underlined row; *: $p<0.05$.

Method	Kept (%)	B/tok	MMLong-34K	MMDocIR-20K
15.2×				
Pruning	6.6	640.00	83.0 (−7.9)	67.2 (−19.5)
Pruning + BaMIC-8b	13.1	322.00	88.2 (−5.5*)	79.1 (−12.1*)
Pruning + BaMIC-4b	26.0	162.00	95.6 (−2.2)	89.7 (−5.6*)
Pruning + BaMIC-2b	51.3	82.00	106.5 (+2.7)	95.1 (−2.2*)
BaMIC-1b	100.0	42.00	100.5	98.8
63.9×				
Pruning	1.6	640.00	57.2 (−14.0)	35.5 (−34.5)
Pruning + BaMIC-2b	12.3	82.00	93.5 (+2.4)	75.4 (−9.9)
Pruning + BaMIC-1b	24.0	42.00	97.4 (+4.2)	82.6 (−5.5)
Pruning + PQ	50.0	20.02	100.0 (+5.3)	92.9 (+0.9)
Pruning + BaMIC-PQ	50.0	20.02	102.6 (+6.5)	94.0 (+1.5)
PQ	100.0	10.02	88.0 (−0.2)	86.2 (−3.2)
BaMIC-PQ	100.0	10.02	88.2	91.5

Table 7: Statistical significance of geometry calibration across compression regimes. Δ denotes Hit@1 improvement after calibration. CI represents 95% confidence intervals over paired queries.

Compression	ΔHit@1 (pp)	95% CI	Significant pairs	Cohen's d
4×	+0.40	[+0.17, +0.63]	1/8	+0.035
8×	+0.16	[-0.10, +0.43]	0/8	+0.009
16×	+0.43	[+0.11, +0.74]	0/8	+0.019
32×	+1.86	[+0.91, +2.82]	5/8	+0.071
64×	+4.72	[+3.62, +5.82]	8/8	+0.145

we report it under PQ. For ScaNN we use `scann` 1.4.2 with anisotropic quantization (η=0.2) and obtain the isotropic variant by disabling the anisotropic loss; its lookup layout stores each 4-bit code in one byte, so at equal serialized size it carries half as much code information as PQ.

C Joint Allocation Between Vector Count and Precision

The comparisons in Section 5 place each method on one edge of the allocation space: full support at reduced precision, or reduced vector count at fp16. To test whether our conclusion depends on this design, Table 5 evaluates interior allocations that prune a fraction of patch vectors per page and quantize the survivors at the same serialized bytes, with PQ codebooks for the retained vectors using m=20 and trained as in App. B.2. We use uniform pruning, the strongest vector-count baseline in five of the eight settings at 64× (Table 1).

Spending the budget on vector count at fp16 is the worst allocation at both budgets and on both corpora, trailing the full-support reference by 7.9–19.5 pp at 15.2× and by 14.0–34.5 pp at 63.9×, and every interior allocation outperforms it. The deficit of vector-count baselines in Section 5 is therefore not an artifact of storing retained vectors in fp16. Above one bit per dimension, full support is best or tied: on MMDocIR-20K, BaMIC-1b exceeds every interior point, while on MMLong-34K the best interior point (51% of vectors at 2 bits) is 2.7 pp higher but not significant (p=0.06).

At 63.9×, halving the vector count to double the codebook budget becomes competitive: Pruning + BaMIC-PQ exceeds BaMIC-PQ by 6.5 pp on MMLong-34K and by 1.5 pp on MMDocIR-20K. At m=10, decoded directions are coarse and codes begin to collide, whereas pruning half of the vectors removes little coverage (0.8–2.5 pp at 2× in Table 1). Calibration remains effective on the retained subset (+2.6 and +1.1 retention points), so allocation and radial correction are complementary. Full support is thus a robust default that is never beaten by vector-count reduction, while choosing the kept fraction adaptively at extreme sub-bit budgets is a natural extension of BaMIC.

Table 6: Equal-byte comparison against vector-count compression. Gray rows: Hit@1 of the best vector-count; others: Hit@1 gain (pp) over it. McNemar $p < 0.05^{*}$, 0.01^{**}, 0.001^{***}.

CR	Method	DocIR-20K	Long-34K	Mean d
4×	*Best vec.-count*	0.538	0.446	
	BaMIC-4b	+8.1***	+0.5	+0.13
32×	*Best vec.-count*	0.326	0.379	
	PQ	+26.1***	+8.7***	+0.34
	OPQ	+25.3***	+5.3**	+0.29
	ScaNN-aniso	+23.5***	+6.0**	+0.29
	ScaNN-iso	+22.2***	+4.0*	+0.26
	BaMIC-PQ	+27.5***	+8.1***	+0.35
	BaMIC-OPQ	+27.4***	+7.7***	+0.34
64×	*Best vec.-count*	0.219	0.296	
	PQ	+31.3***	+10.1***	+0.38
	OPQ	+30.5***	+9.4***	+0.36
	ScaNN-aniso	+26.0***	+11.3***	+0.35
	ScaNN-iso	+21.9***	+7.9***	+0.28
	BaMIC-PQ	+34.5***	+10.3***	+0.41
	BaMIC-OPQ	+34.5***	+15.6***	+0.45

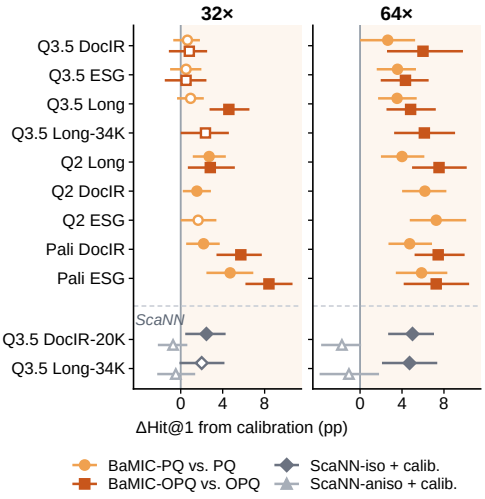


Figure 6: Per-corpus calibration gain with 95% CI; filled markers: $p < 0.05$.

D Statistical Significance Analysis

All comparisons are paired on identical queries. For each comparison we report the Hit@1 difference Δ in percentage points (pp), a 95% confidence interval from 5,000 bootstrap resamples with queries clustered by document, and an exact McNemar test on discordant queries. Cohen's d is also reported; since per-query outcomes are binary, d is small by construction and is best interpreted together with Δ and the test results.

D.1 Equal-Byte Comparison with Vector-Count Compression

Table 6 compares full-support methods with the strongest vector-count baseline under equal serialized bytes. At 32× and 64×, every full-support method outperforms this baseline on both corpora, with gains of 21.9–34.5 pp on MMDocIR-20K and 4.0–15.6 pp on MMLong-34K. All 24 comparisons are significant, 23 of them at $p < 0.01$. The margin on MMDocIR-20K grows with compression, as the Hit@1 of the best vector-count baseline drops from 0.326 at 32× to 0.219 at 64×. BaMIC yields the largest gain in three of the four settings, up to +15.6 pp for BaMIC-OPQ on MMLong-34K at 64× ($d = 0.45$). At 4×, the gain is significant on MMDocIR-20K (+8.1 pp) but not on MMLong-34K (+0.5 pp), where both methods stay within 1.5% of full precision.

D.2 Calibration Gain Across Compression Ratios

Table 7 reports the effect of calibration across compression ratios, averaged over eight settings. Up to 16×, the mean gain remains below 0.5 pp and at most one corpus is individually significant. The gain becomes substantial only in the sub-bit regime, reaching +1.86 pp at 32× (5/8 significant) and +4.72 pp at 64× (8/8 significant, 95% CI [3.62, 5.82]). This pattern agrees with the analysis in Section 3: above one bit per dimension, reconstructed norms remain nearly uniform, whereas codebook reconstruction at lower rates introduces the radial dispersion that calibration removes.

D.3 Per-Corpus and Cross-Quantizer Consistency

Figure 6 breaks down the sub-bit results by corpus, where Q3.5, Q2 and Pali denote ColQwen3.5, ColQwen2 and ColPali. At 64×, all 15 PQ and OPQ comparisons are significant at $p < 0.01$, and 14 remain significant after Bonferroni correction over the 38 tests in the figure. The gain for OPQ matches or exceeds that for PQ in 11 of the 12 corpus–budget pairs where both are available, in line with the larger norm dispersion that OPQ leaves after decoding. Gains are also largest where uncalibrated quality is lowest: at 32×, calibration improves ColPali on MMESGBench by +4.7 pp

Table 8: Controlled analysis of geometry calibration at 64× compression with ColQwen3.5. All variants use the same full-support compressed index and differ only in their treatment of decoded token norms. Values are retention (%) relative to full precision.

			MMDocIR-20K				MMLong-34K			
			PQ		OPQ		PQ		OPQ	
Variant	Norm eq.	+B/tok	H@1	N@4	H@1	N@4	H@1	N@4	H@1	N@4
Full precision			100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Codebook, w/o calibration	×	0	86.2	88.5	85.0	86.7	88.0	89.6	86.4	88.3
+ global rescaling (×2)	×	0	86.2	88.5	85.0	86.7	88.0	89.6	86.4	88.3
+ MSE-optimal norm (stored)	×	0.5	54.3	58.7	78.9	81.5	42.8	53.4	72.3	78.4
+ per-token calibration (**BaMIC**)	✓	0	**91.5**	**92.0**	**91.5**	**94.4**	**88.2**	**92.0**	**100.0**	**97.0**
Gain from calibration			+5.3	+3.5	+6.5	+7.7	+0.2	+2.4	+13.6	+8.7
Decoded-norm std			0.053		0.046		0.047		0.059	

with PQ and +8.4 pp with OPQ, compared with less than 1 pp for ColQwen3.5 on MMDocIR and MMESGBench.

Applying the same calibration to ScaNN provides a mechanism check. It improves the isotropic variant by 2.0–5.0 pp, significantly in three of the four settings, but has no significant effect on the anisotropic variant (-0.5 to -1.7 pp), whose training objective already suppresses norm dispersion. Calibration thus helps when the quantizer leaves radial dispersion in place and has no significant effect otherwise.

E Controlled Analysis of Geometry Calibration

Table 8 isolates the effect of geometry calibration at 64× compression. All variants use the same full-support compressed index, fixing the retained tokens and decoded directions while changing only how decoded norms enter scoring. We compare the uncalibrated codebook with three controlled interventions: global norm rescaling, reconstruction-oriented per-token norms, and norm equalization as used by BaMIC.

Global rescaling leaves retrieval unchanged, confirming that a shared change in norm scale does not affect ranking. In contrast, storing MSE-optimal per-token norms does not recover retrieval quality despite adding 0.5 bytes per token; on MMDocIR-20K, Hit@1 retention drops from 86.2% to 54.3% for PQ and from 85.0% to 78.9% for OPQ. These controls indicate that the relevant distortion is not global scale or inaccurate norm reconstruction, but token-dependent radial variation that changes competition under MaxSim.

Removing this variation with BaMIC improves Hit@1 retention by up to 13.6 points and NDCG@4 by up to 8.7 points. The largest gain occurs for OPQ on MMLong-34K, which also has the largest decoded-norm dispersion among the four settings, while the smaller PQ Hit@1 gain on the same corpus is accompanied by a 2.4-point NDCG@4 improvement. Finally, nADC realizes the same calibrated scoring directly from PQ/OPQ codes and a per-index table of squared codeword norms, without additional per-token storage.

F Retrieval Quality Beyond the Top Rank

To verify that the gains of BaMIC are not limited to the top-ranked page, we evaluate retrieval quality at deeper cutoffs (Table 9). In the sub-bit regime, calibration improves every metric: at 64×, BaMIC-OPQ raises MRR from 89.4% to 95.3% of full precision and still improves Hit@10 by 3.8 points over OPQ. It also lowers the mean rank of the first relevant page by about one third, indicating that calibration moves the whole ranked list forward rather than only reordering near-tied top candidates. Above one bit per dimension, BaMIC performs on par with the corresponding low-bit codecs, differing by less than one point at every cutoff with no significant difference on any corpus.

Table 9: Retrieval quality at deeper cutoffs (retention %, mean over the three ColQwen3.5 main corpora). Gold rank: mean rank of the first relevant page, lower is better. Bold: best at each sub-bit ratio.

CR	Method	H@1	H@4	H@10	N@4	MRR	Gold rank↓
1×	Full precision	100.0	100.0	100.0	100.0	100.0	100.0
4×	Faiss SQ-4b	100.0	99.8	99.7	99.9	100.0	98.7
	BaMIC-4b	100.4	99.7	99.8	99.9	100.2	100.4
16×	EDEN-1b	99.3	100.0	99.9	99.3	99.5	108.7
	BaMIC-1b	98.7	99.7	99.4	99.0	99.2	108.3
32×	PQ	95.5	97.5	98.5	96.2	96.5	143.0
	OPQ	95.2	96.6	97.9	95.6	96.3	135.6
	ScaNN-aniso	93.8	95.5	97.4	94.2	95.0	155.7
	BaMIC-PQ	96.6	**98.3**	99.0	97.2	97.3	119.5
	BaMIC-OPQ	**98.3**	98.1	**99.3**	**97.9**	**98.5**	**109.2**
64×	PQ	86.9	91.4	94.0	88.4	89.7	204.6
	OPQ	86.6	91.1	94.1	88.1	89.4	182.0
	ScaNN-aniso	82.4	88.8	92.9	85.2	86.4	234.1
	BaMIC-PQ	91.7	94.6	96.8	92.4	93.5	145.4
	BaMIC-OPQ	**94.1**	**96.0**	**97.9**	**94.8**	**95.3**	**121.9**

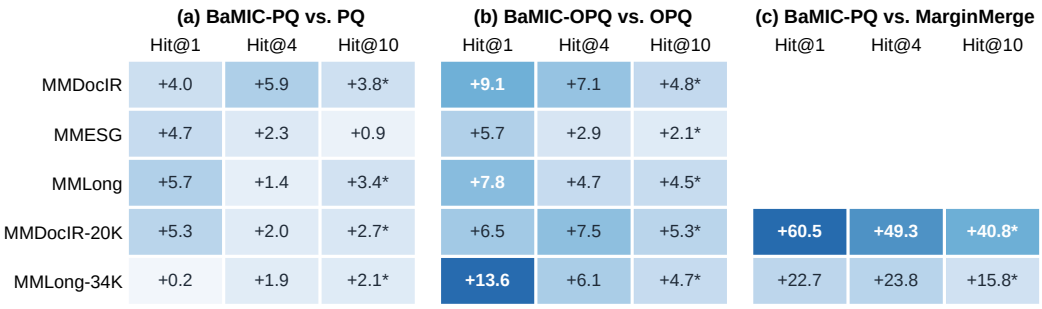


Figure 7: Per-corpus gains of BaMIC at 64× (retention points). *: $p < 0.05$ at Hit@10.

Figure 7 shows that these gains hold consistently across corpora: calibration improves all 30 corpus–codebook–cutoff combinations, and the Hit@10 gain is significant in 9 of the 10 corpus–codebook pairs. The advantage over vector-count compression is an order of magnitude larger and persists at every cutoff; at 64×, BaMIC-PQ exceeds MarginMerge in Hit@10 retention by 40.8 points on MMDocIR-20K and 15.8 points on MMLong-34K.

G Failure Analysis

We analyze how calibration changes individual queries at 64×, restricting to queries answered correctly at full precision. Calibration recovers 6.3–19.8% of these queries and breaks only 1.3–5.7%, so recoveries outnumber regressions in every setting, by a factor of 1.7 to 6.5 (Figure 8a).

Errors concentrate on queries whose gold page wins by a small margin at full precision (Figure 8b). Calibration yields its largest gains on the two lowest margin quartiles, raising Hit@1 by more than 12 points for OPQ, whereas queries with the largest margins are almost always preserved. The queries that remain wrong after calibration do not suffer larger distortion: the directional and radial errors of their winning tokens match those of preserved queries, but their full-precision margin is about three times smaller. These failures therefore reflect fragile rankings that any perturbation at this rate can flip, rather than a residual scoring bias, and most of them are near misses ranked second to fourth. MMLong-34K is the hardest setting in absolute terms: its larger candidate pool lowers full-precision Hit@1 from 0.618 to 0.452, as distractor pages take the top rank from queries with small margins. The queries that remain correct at full precision therefore have larger margins, which is why retention relative to full precision stays high on this corpus (Tables 3 and 8).

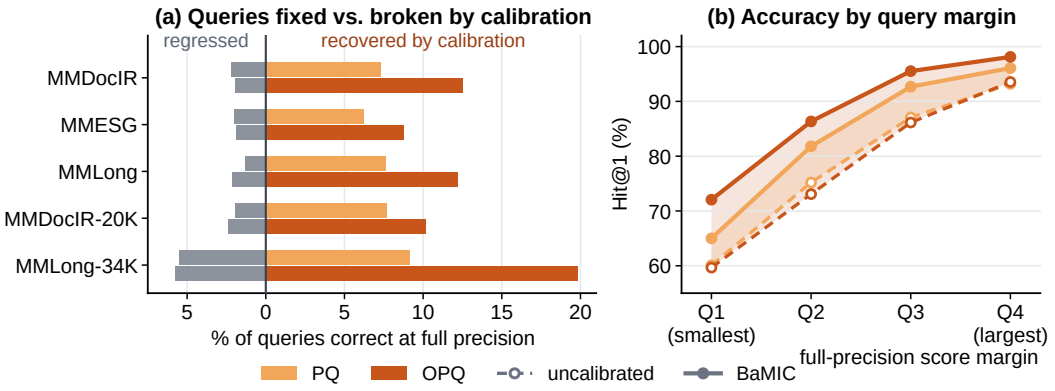


Figure 8: Where calibration helps and where it fails (64×, queries answered correctly at full precision). (a) Share of queries recovered or broken by calibration. (b) Hit@1 by quartile of the full-precision score margin, averaged over corpora.

Table 10: Storage needed to match BaMIC at equal quality. Each cell is the smallest measured index size at which a method reaches the Hit@1 of the BaMIC target, as a multiple of the target's size; ">" denotes that the target is not reached within the measured range.

	BaMIC target at 64×				BaMIC target at 32×			
	MMDocIR-20K		MMLong-34K		MMDocIR-20K		MMLong-34K	
Method	-PQ	-OPQ	-PQ	-OPQ	-PQ	-OPQ	-PQ	-OPQ
Random	32.2×	32.0×	8.2×	32.1×	>16×	>16×	16.1×	16.1×
MarginMerge-p	>32×	>32×	4.2×	>32×	>16×	>16×	>16×	>16×
MarginMerge-m	>32×	>32×	4.2×	>32×	>16×	>16×	>16×	>16×
Light-ColPali	>32×	>32×	8.2×	16.2×	>16×	>16×	8.1×	8.1×
DocPruner	>32×	>32×	>32×	>32×	>16×	>16×	>16×	>16×
PQ	1.6×	1.6×	2.0×	2.0×	1.6×	1.6×	1.0×	1.0×
OPQ	2.0×	2.0×	2.0×	>2×	>1×	>1×	>1×	>1×

H Storage at Equal Quality

Table 10 reverses the equal-byte comparison: for a BaMIC configuration at 32× or 64×, it reports how much larger an index each baseline needs to reach the same Hit@1, using only measured operating points without interpolation. To match BaMIC-OPQ at 64×, the strongest vector-count method requires 32× more storage on MMDocIR-20K and 16× more on MMLong-34K. Most vector-count methods fail to reach BaMIC at all: on MMDocIR-20K, both merging operators and DocPruner remain below every target even at 2× compression, the largest budget we measure, and DocPruner reaches no target on either corpus. The most favorable case for vector-count methods is BaMIC-PQ on MMLong-34K, where the merging operators match it with about 4× more storage, consistent with the small calibration gain of PQ on this corpus.

Uncalibrated codebooks need more storage to match BaMIC at 64×: 1.6–2× on MMDocIR-20K, and 2× for PQ and more than 2× for OPQ on MMLong-34K. Because only measured operating points are used, these multiples are upper bounds set by the factor-of-two spacing of the measured budgets; for BaMIC-PQ on MMLong-34K, for instance, the Hit@1 gap to PQ is only 0.2 points. At 32×, PQ matches both BaMIC variants on MMLong-34K, whereas OPQ does not reach them within its measured range. Scalar quantizers are omitted, since their lowest rate of one bit per dimension already exceeds these budgets.