Multi-Vector Index Compression in Any Modality

Hanxiang Qin*
hqin14@jhu.edu
Johns Hopkins University
Baltimore, MD, USA

Alexander Martin*
amart233@jhu.edu
Johns Hopkins University
Baltimore, MD, USA

Rohan Jha
rjha5@jhu.edu
Johns Hopkins University
Baltimore, MD, USA

Chunsheng Zuo
czuo3@jhu.edu
Johns Hopkins University
Baltimore, MD, USA

Reno Kriz
rkriz1@jhu.edu
Johns Hopkins University
Baltimore, MD, USA

Benjamin Van Durme
vandurme@jhu.edu
Johns Hopkins University
Baltimore, MD, USA

Abstract

We study efficient multi-vector retrieval for late interaction in any modality. Late interaction has emerged as a dominant paradigm for information retrieval in text, images, visual documents, and videos, but its computation and storage costs grow linearly with document length, making it costly for image-, video-, and audio-rich corpora. To address this limitation, we explore query-agnostic methods for compressing multi-vector document representations under a constant vector budget. We introduce four approaches for index compression: sequence resizing, memory tokens, hierarchical pooling, and a novel attention-guided clustering (AGC). AGC uses an attention-guided mechanism to identify the most semantically salient regions of a document as cluster centroids and to weight token aggregation. Evaluating these methods on retrieval tasks spanning text (BEIR), visual-document (ViDoRe), and video (MSR-VTT, MultiVENT 2.0), we show that attention-guided clustering consistently outperforms other parameterized compression methods (sequence resizing and memory tokens), provides greater flexibility in index size than non-parametric hierarchical clustering, and achieves competitive or improved performance compared to a full, uncompressed index.[1]

CCS Concepts

• **Information systems** → **Search index compression**.

Keywords

Multi-vector representations, Index compression, Late interaction, Omni-modal retrieval

ACM Reference Format:
Hanxiang Qin, Alexander Martin, Rohan Jha, Chunsheng Zuo, Reno Kriz, and Benjamin Van Durme. 2026. Multi-Vector Index Compression in Any Modality. In *Preprint*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

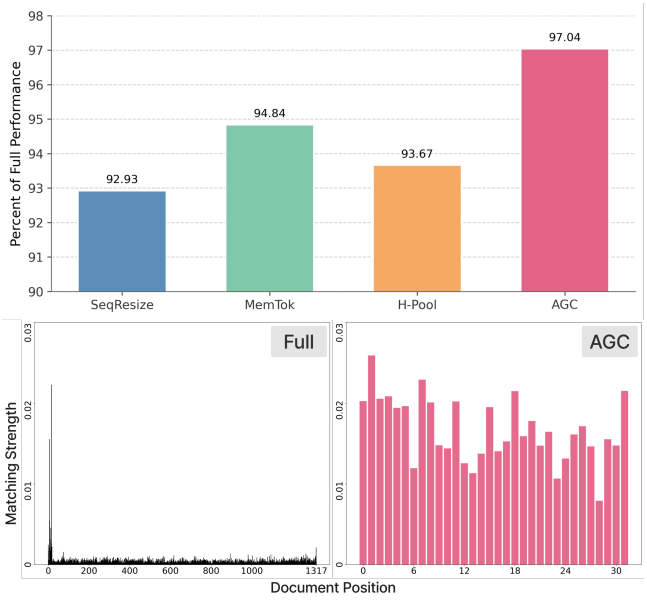

Figure 1: We explore index compression in any modality. We introduce SeqResize, projection-based, MemTok, token-based, H-Pool, heuristic-based, and AGC (Ours), hybrid attention-similarity. AGC better utilizes index tokens while maintaining performance (nDCG@10) at high compression.

1 Introduction

Online information is increasingly multimodal, including videos, articles with figures or images, podcasts, and interactive web content. It is thus essential that information retrieval systems be able to index over large multimodal collections. However, indexing multimodal content at scale requires an incredible amount of storage,[2] limiting the ability of search providers to build truly multimodal indices. While recent advances in multi/omnimodal retrieval [7, 12, 27, 33, 42] have begun to make performance gains, significant progress is still needed to achieve scalable performance in real-world settings.

*Equal Contribution
[1]The source code is available at: github.com/hanxiangqin/omni-col-press

[2]Indexing 1 video takes 10MB under multi-vector setting, but it is estimated that YouTube hosts 14 billion videos [37], estimating an index size of 140 Petabytes.

Preprint '26,

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

In this work, we focus on multi-vector late interaction [21], which has shown promise in multimodal domains [12, 13, 38, 42, 50, 55]. Optimizations like ColBERTv2 [44] have improved efficiency through a two-stage retrieval pipeline that uses document cluster centroids to avoid scoring every document, enabling sub-linear scaling in collection size. However, the computation and storage cost still grow linearly with document length [43, 53]. This linear scaling presents a prohibitive barrier for multimodal corpora, where a single multimodal document could easily reach thousands of tokens. Additionally, these thousand token representations are often underutilized in late-interaction, making the indices built from them largely unnecessary in practice (Figure 1). We find that most multimodal late interaction models use only about 1% of their index during a full evaluation pass. To address this gap, we propose learning compact, query-agnostic, multi-vector, multimodal document representations under a constant vector budget. By bounding the document representation to a constant size, we ensure that both index storage and query-time costs remain manageable and fully customizable to storage or compute constraints, while retaining the benefits of fine-grained late interaction.

We adapt three strong performing multi-vector compression methods from textual domain to multimodal: (1) *Sequence Resizing* (SeqResize) [e.g., 35], where a full multi-vector document is projected down along the sequence dimension by an MLP; (2) *Memory Tokens* (MemTok) [e.g., 32, 53], where learnable vectors are appended to the document context and used as the representation; and (3) *Hierarchical Pooling* (H-Pool) [e.g., 9], which iteratively groups similar vectors and replaces them with their mean. However, these methods are ill-suited for multimodal compression as they struggle to handle redundant and noisy inputs, or suffer from representation collapse. To address these limitations, we introduce a novel attention-guided clustering, where learnable universal query tokens are used to guide the attention to select centroids and weight the aggregation for clustering (AGC).

We evaluate these methods across four tasks and three modalities: BEIR [48], a document retrieval benchmark (text), ViDoRe [36], a visual document retrieval benchmark (vision), MSR-VTT [54], a video-retrieval benchmark (vision), and MultiVENT 2.0 [23], a video-retrieval benchmark (audiovisual). On these benchmarks, we provide an extensive set of experiments and introduce new state-of-the-art results on ViDoRe and MSR-VTT. We find that AGC is the strongest compression technique in any modality, offering the best performance at learned compression rates and better transferability between sizes than non-parametric compression (H-Pool). Additionally, we find that training with a compression objective can improve performance over an uncompressed multi-vector index on ViDoRe and MSR-VTT, highlighting that compression reduces the redundancy and noise of multimodal inputs.

Our contributions are summarized as follows:

(1) We introduce four methods for index compression in any modality: SeqResize, MemTok, H-Pool, and AGC.
(2) AGC presents a novel approach, in which learnable universal query tokens select centroids and weight cluster pooling.
(3) We present a series of experiments demonstrating the strong performance and flexibility of AGC across document, visual document, and video retrieval settings.

2 Related Work

Multimodal Retrieval. Many works have introduced benchmarks for evaluating representations in information retrieval. In the text-only setting, evaluation suites such as MS MARCO [3] and BEIR [48], have become standard for measuring retrieval effectiveness across diverse domains, tasks, and query types. Video retrieval has been extensively studied using benchmarks that use natural language descriptions to retrieve videos, such as, MSR-VTT [54], VATEX [51], DiDeMo [15], and ActivityNet Captions [22]. More recently, MultiVENT 2.0 [23] has provided a large-scale, multilingual benchmark for real world video retrieval. Visual document retrieval has also recently emerged as another challenging multimodal task, requiring strong optical character recognition ability and visual understanding of layout and graphics, e.g., ViDoRe [36] and MMDocIR [10]. Complementary efforts have introduced modality-specific embedding benchmarks, including MTEB for text embeddings [39], MSEB for audio embeddings [14], and MMEB for vision-language embeddings [20]. In this work, we focus on the following settings: text, visual document, video (vision only) and video (audiovisual). We believe this covers the most challenging combinations of modalities and model capabilities.

Multi-Vector Index Compression. Multi-vector embeddings offer a number of distinct axes by which to compress the index. Naturally, being just a collection of vectors, it is amenable to the same quantization [11, 18] and truncation [19, 24] methods as single-vector retrieval. It is also the norm in multi-vector text retrieval to down-project the encoder's large hidden dimension to a more manageable dimension (e.g. 768 $\rightarrow$ 128) [21]. Furthermore, as is the focus of this paper, *multi*-vector indices can be compressed along the sequence dimension as well in order to end up with fewer tokens. This is generally split between methods that prune tokens according to simple corpus-level or attentional heuristics [1, 61], pool them implicitly via special tokens that aggregate meaning or explicitly via heuristic representation merging [9, 16, 25, 41, 49], or simply project the sequence length into a fixed quantity of embeddings [35]. Finally, index methods like PLAID [43, 44] cluster the document token vectors and represent each as its nearest cluster centroid plus a low-bit-quantized version of its residual.

Attention-based Compression. To address the computational burden of long contexts for language models, prior work explores token compression via KV cache eviction. While query-aware methods effectively prune tokens based on prompt attention [8, 17, 28, 30, 47, 58, 59], they are incompatible with retrieval indexing, which requires document representations to be computed before the query is known. On the query-agnostic side, approaches instead leverage self-attention scores or learnable parameters to determine token importance [5, 6, 41, 56, 60]. However, a critical gap remains: these methods optimize for generative tasks by preserving the global "gist," whereas retrieval requires retaining discriminative details needed for distinguishing hard negatives from positive documents.

3 Preliminaries

Retrieval. Given a collection of documents $\mathcal{D}$, where each document $d \in \mathcal{D}$ contains one or more modalities (e.g., text, audio,

visual), and a text query q, we look to provide a ranking of documents in $\mathcal{D}$ based on their relevance to q.

Late Interaction. To calculate the similarity score $\mathcal{S}$ between two multi-vector representations, we adopt ColBERT-style Late Interaction [21].

Given a query representation $\mathbf{Q}$ and a document representation $\mathbf{C}$, we compute the relevance score $s(q,d)$ via the MaxSim operation:

$$s(q,d) = \sum_{i=1}^{n_q} \max_{1 \le j \le m} \langle \mathbf{q}_i, \mathbf{c}_j \rangle$$

where $\langle \cdot, \cdot \rangle$ denotes the dot product. By summing the maximum similarities for each q_i, we get a score for d's relevance to q.

3.1 Problem Formulation

We formulate the compression of multimodal documents as the process of generating optimal representations under constraints for scalable late-interaction retrieval.

Query and Document Representation. We employ a query encoder ϕ that maps a query q to a sequence of token embeddings $\mathbf{Q} \in \mathbb{R}^{n_q \times h}$, where n_q is the sequence length and h is the embedding dimension. We do not constrain the query length.

For documents, we define a mapping π that transforms a document d into a sequence of m vectors:

$$\pi : d \mapsto \mathbf{C} = \left[\mathbf{c}_1, \ldots, \mathbf{c}_m \right] \in \mathbb{R}^{m \times h}$$

Here, m is a fixed budget of vectors independent of the document's original length. The mapping π represents the full representation generation pipeline which may involve direct encoding, parametric or heuristic compression, or a combination of these operations.

A critical constraint in retrieval is that this mapping must be applied during indexing, where the query q remains unknown. Consequently, π must compress the document in a query-agnostic manner while preserving information likely to be relevant for future queries. In this work, we explore both unparameterized and parameterized compression techniques, denoting parameterized formulations as π_θ, where θ denotes learnable weights.

Objective. Our goal is to define π such that a scoring function $\mathcal{S}(\mathbf{Q}, \mathbf{C})$ assigns higher scores to relevant query-document pairs compared to less relevant ones. For parameterized mappings, this becomes an optimization problem where we seek to maximize retrieval accuracy within the fixed storage budget m.

4 Multi-Vector Compression

We introduce three methods for multi-vector compression based on prior work (pictured in Figure 2): sequence resizing (SeqResize), memory tokens (MemTok), and hierarchical pooling (H-Pool).

4.1 Sequence Resizing

SeqResize is a parameterized compression method that projects the output of an encoder along the sequence dimension to a compressed representation with a fixed number of tokens. Prior work first used this method in document compression for text retrieval [35]. SeqResize is an intuitive approach, allowing an encoder to fully contextualize all representations in a document and then parameterize compression separately (but trained jointly) from the encoder.

To perform SeqResize, the tokens of the document $\mathbf{X} \in \mathbb{R}^{n \times h}$ are passed into the bidirectional encoder F_{enc}, which is an L-layer transformer. Let $\mathbf{Z}^{(L)} = F_{\text{enc}}(\mathbf{X}; \theta) \in \mathbb{R}^{n \times h}$ be the last-layer hidden states. Since n varies across documents, we first pad or truncate $\mathbf{Z}^{(L)}$ to a fixed length n_0:

$$\bar{\mathbf{Z}}^{(L)} = \text{PadTrunc}(\mathbf{Z}^{(L)}, n_0) \in \mathbb{R}^{n_0 \times h}.$$

We then resize along the sequence dimension to produce the compressed multi-vector representation $\mathbf{C} \in \mathbb{R}^{m \times h}$ using a 2-layer MLP:

$$\mathbf{C} = \left(\sigma(\bar{\mathbf{Z}}^{(L)\top} \mathbf{W}_1^\top) \mathbf{W}_2^\top \right)^\top, \qquad \mathbf{W}_1 \in \mathbb{R}^{d \times n_0},\ \mathbf{W}_2 \in \mathbb{R}^{m \times d}.$$

Here, h is the hidden dimension, θ are the parameters of the encoder, and σ is a nonlinearity (e.g., ReLU). The transpose in the MLP form indicates that the same MLP maps each hidden channel's length-n_0 sequence to a length-m sequence, i.e., it operates over the sequence dimension. The parameters of $\theta, \mathbf{W}_1, \mathbf{W}_2$ are the learnable parameters of the compression function.

4.2 Memory Tokens

MemTok is a parameterized compression method that appends learnable "memory tokens" to a document context to use as the document representation. Prior works have used this method in document compression for retrieval [53] and generation [32]. MemTok approaches are a straightforward compression method, allowing for document compression to leverage a single encoder, instead of parameterizing compression in a different network (e.g., SeqResize). Following these works, we aim to learn memory tokens for any document representation.

To perform compression with MemTok, we append a set of m memory tokens $\mathbf{M} \in \mathbb{R}^{m \times h}$ to the document tokens $\mathbf{X} \in \mathbb{R}^{n \times h}$, and feed the concatenated sequence into the encoder F_{enc}, which is an L-layer transformer. After bidirectional self-attention, each memory token attends over the entire document, and we discard all non-memory positions. The final states of the m memory tokens form the compressed multi-vector representation of the document.

Let $\mathbf{Z}^{(L)} = [\mathbf{Z}_\mathbf{X}, \mathbf{Z}_\mathbf{M}] \in \mathbb{R}^{(n+m) \times h}$ be the hidden states of the last layer of the encoder, where $\mathbf{Z}_\mathbf{X} \in \mathbb{R}^{n \times h}$ and $\mathbf{Z}_\mathbf{M} \in \mathbb{R}^{m \times h}$ are the hidden states of the document and memory tokens, respectively. Formally, the compressed multi-vector representation of the document $\mathbf{C} \in \mathbb{R}^{m \times h}$ is given by:

$$[\mathbf{Z}_\mathbf{X}^{(L)}, \mathbf{Z}_\mathbf{M}^{(L)}] = F_{\text{enc}}([\mathbf{X}, \mathbf{M}]; \theta),$$
$$\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_m] = \mathbf{Z}_\mathbf{M}^{(L)}.$$

where h is the hidden dimension, θ are the parameters of the encoder, which are the learnable parameters of the compression function. In our experiments, the memory tokens $\mathbf{M}$ are initialized as learnable parameters and updated during training.

4.3 Hierarchical Pooling

H-Pool is a non-parametric compression method that iteratively groups similar vectors and replaces them with their mean. Prior work has introduced this method in document compression in the text domain [9]. Unlike the other approaches, H-Pool does not require the model to be trained for compression and allows for a simple heuristic-driven approach agnostic to modality or model.

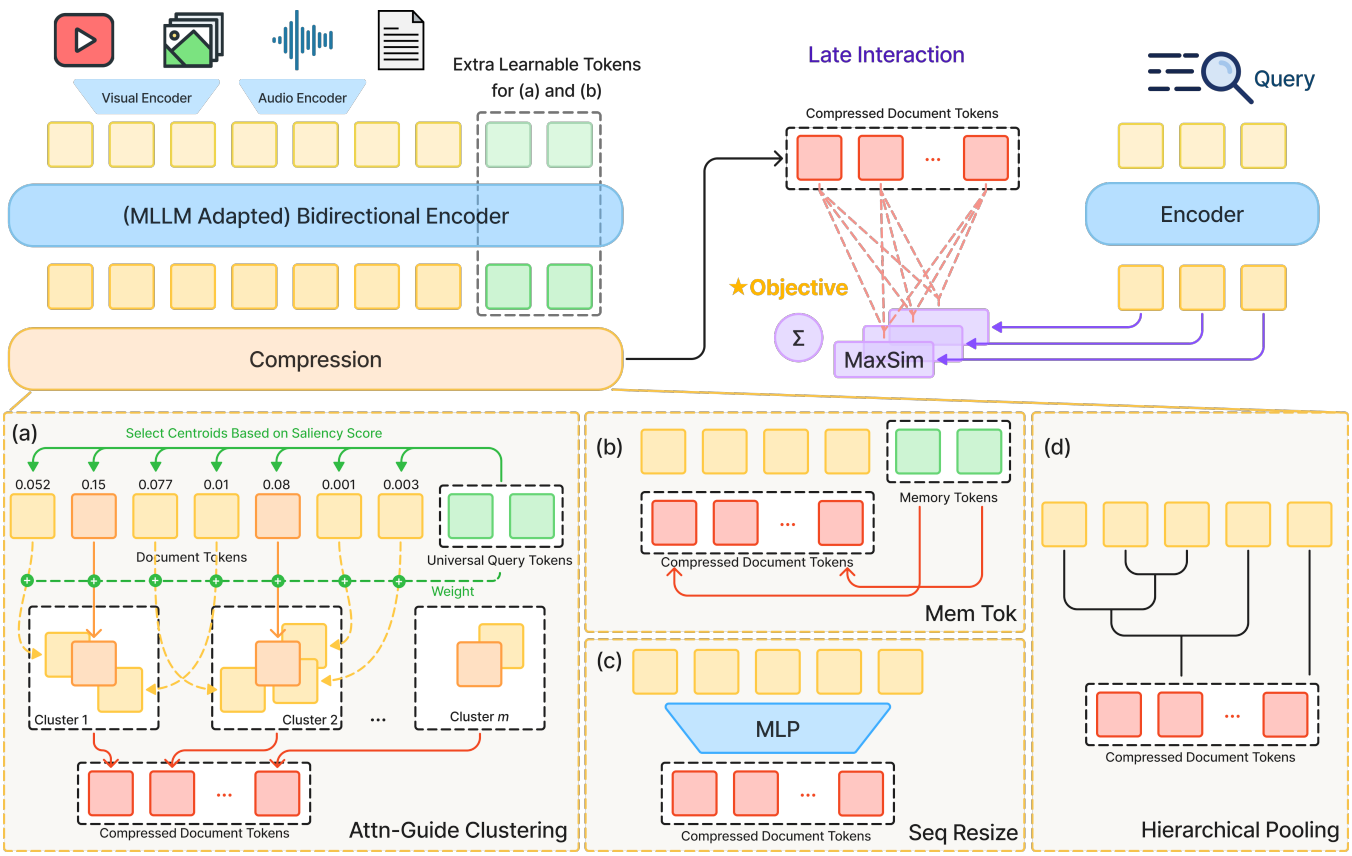

Figure 2: Overview of multi-vector index compression techniques. (a) AGC uses universal query tokens to guide attention-based centroid selection and weight the aggregation of clustering. (b) MemTok appends tokens to the document context to act as the final representation. (c) SeqResize down projects a document representation along the sequence dimension. (d) H-Pool iteratively groups similar vectors and replaces them with their mean.

For each sequence, H-Pool starts from a sequence of token embeddings $\mathbf{X} \in \mathbb{R}^{n \times h}$. We compute a cosine distance matrix $\mathbf{R} \in \mathbb{R}^{n \times n}$ with entries

$$r_{ij} = 1 - \frac{x_i^\top x_j}{\|x_i\|_2 \, \|x_j\|_2},$$

which we use to run agglomerative hierarchical pooling with Ward linkage [52]. At any step of the algorithm, we maintain clusters as index sets $\{\mathcal{A}_1, \ldots, \mathcal{A}_k\}$; each cluster $\mathcal{A}_a$ has centroid

$$\mu_a = \frac{1}{|\mathcal{A}_a|} \sum_{i \in \mathcal{A}_a} x_i.$$

Ward's method iteratively merges the pair of clusters $(\mathcal{A}_a, \mathcal{A}_b)$ that minimizes the increase in within-cluster squared error,

$$\Delta_{a,b} = \frac{|\mathcal{A}_a| \, |\mathcal{A}_b|}{|\mathcal{A}_a| + |\mathcal{A}_b|} \, \|\mu_a - \mu_b\|_2^2,$$

until exactly $m - m'$ clusters remain, yielding a partition $\{\mathcal{A}_1, \ldots, \mathcal{A}_{m-m'}\}$. The pooled token embeddings are then defined as the mean of each cluster:

$$c_j = \frac{1}{|\mathcal{A}_j|} \sum_{i \in \mathcal{A}_j} x_i, \quad j = 1, \ldots, m - m',$$

In implementation, we provide the option to keep m' tokens as protected tokens and concatenate them back to the pooled token embeddings to form the final compressed sequence $\mathbf{C} = [c_1, \ldots, c_m]$.

4.4 Limitations

SeqResize, MemTok, and H-Pool reveal limitations when applied to multimodal data under a fixed token budget constraint. First, parametric methods like SeqResize exhibit a modeling failure in which many tokens remain unused during a single evaluation pass; consequently, it fails to scale effectively with the token budget (see subsection 6.4). MemTok suffers from information collapse: its architecture inherently smooths over distinct features, impeding the effective utilization of multi-vector representations. (See subsection 6.5). Second, neither SeqResize nor MemTok possesses the necessary heuristics to eliminate redundant information. Audio and visual signals are often semantically empty or redundant, such as silent audio segments, static backgrounds, and unchanged temporal sequences [4, 26]. These methods waste their limited token budgets on encoding repetitive and noisy signals rather than capturing key semantic content. Finally, while H-Pool actively removes redundancy, the reliance on greedy iterative merging makes

them vulnerable to noisy outliers, like the aforementioned noise in multimodal data.

5 Attention-Guided Clustering

We now describe Attention-Guided Clustering (AGC), a compression technique designed to maximize the utility of a fixed token budget for document compression in any modality. AGC (shown in Figure 2 (a)) combines three main components: (i) *Attention-based Centroid Selection*, which utilizes learned universal query tokens to identify semantically salient information; (ii) *Hard Clustering*, which uses hard assignment to group tokens to reduce redundancy while preserving distinct semantic details; and (iii) *Weighted Aggregation*, which constructs the final compressed representations by averaging tokens within each cluster weighted by their saliency to mitigate the optimization challenges of hard operations.

5.1 Attention-based Centroid Selection

The first component of our approach is to identify information-rich regions within a document. To estimate token importance without specific user queries, we introduce learned "universal queries," special tokens that probe the document for significant content.

Formally, we append a set of trainable tokens $\mathbf{X}_{\mathbf{\Psi}} \in \mathbb{R}^{|\mathbf{\Psi}| \times h}$, where $\mathbf{\Psi}$ denotes the set of indices for these tokens, to the document tokens $\mathbf{X}$ and pass the concatenated sequence into the bidirectional encoder F_{enc}, an L-layer transformer:

$$[\mathbf{Z}_{\mathbf{X}}^{(L)}, \mathbf{Z}_{\mathbf{\Psi}}^{(L)}] = F_{\text{enc}}([\mathbf{X}, \mathbf{X}_{\mathbf{\Psi}}]; \theta)$$

where θ represents the encoder parameters and $\mathbf{Z}^{(L)}$ denotes the hidden states at the last layer.

We then leverage the attention mechanism to quantify the importance of each document token. Let $\text{Attn}_i^{(L,\eta)} \in \mathbb{R}^n$ denote the attention weights from the universal query token $i \in \mathbf{\Psi}$ to all document tokens at the last layer L and head η. To obtain a global measure of importance, we average over heads and across universal query tokens to compute the saliency scores $\boldsymbol{\alpha} \in \mathbb{R}^n$:

$$\boldsymbol{\alpha} = \frac{1}{|\mathbf{\Psi}|\, H} \sum_{i \in \mathbf{\Psi}} \sum_{\eta=1}^{H} \text{Attn}_i^{(L,\eta)}.$$

The aim of $\boldsymbol{\alpha}$ is to capture high-level semantic relevance, allowing the model to distinguish signal from noise before clustering begins. Using these saliency scores, we then select cluster centroids. We select the top-m tokens with the highest saliency scores, where m is the target budget. Let $\mathcal{I} \subset \{1, \ldots, n\}$ denote the indices corresponding to the m largest values in $\boldsymbol{\alpha}$. We define the cluster centroids as $\mathcal{M} = \{\boldsymbol{\mu}_k\}_{k=1}^m$, where each centroid corresponds to a selected token representation $\boldsymbol{\mu}_k = \mathbf{Z}_{\mathbf{X},j}^{(L)}$ for some $j \in \mathcal{I}$.

5.2 Clustering

Next, we need to organize the rest of the tokens. We group every other token with the centroid it is most similar to, effectively gathering related context into coherent clusters. This ensures that even if a word isn't selected as a centroid, its information is preserved by being associated with a relevant cluster.

Formally, we assign every document token to its nearest centroid based on cosine similarity:

$$\mathcal{G}_k = \left\{ j \in \{1, \ldots, n\} \;\middle|\; k = \operatorname*{argmax}_{k' \in \{1,\ldots,m\}} \cos(\mathbf{Z}_{\mathbf{X},j}^{(L)}, \boldsymbol{\mu}_{k'}) \right\}.$$

Similar to H-Pool, this clustering step reduces redundancy by grouping semantically similar tokens. However, unlike H-Pool, which relies on iterative agglomerative merging, our approach anchors clusters around the globally salient centroids identified by the universal queries. This ensures that the compression is guided by semantic importance rather than just local geometric proximity. Furthermore, by employing hard assignment rather than fully soft operations (e.g., MemTok), we ensure that distinct semantic concepts remain separated in the latent space, alleviating the risk of over-smoothing.

5.3 Weighted Aggregation

With clusters established, we aggregate the tokens in each group into a compact representation. Naive averaging treats all inputs equally, ignoring the varying information density typical of multimodal data. Just as P-frames in video are compressed more heavily than I-frames [26], or text outweighs margins in a document, we must distinguish signal from redundancy. We therefore employ *Weighted Aggregation*, where the saliency score $\boldsymbol{\alpha}$ naturally serves as a learnable importance weight to prioritize critical content.

Formally, we construct the document's compressed representation $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_m] \in \mathbb{R}^{m \times h}$. We compute each cluster vector $\mathbf{c}_k$ as the weighted average of the document tokens assigned to it:

$$\mathbf{c}_k = \frac{\sum_{j \in \mathcal{G}_k} \boldsymbol{\alpha}_j \mathbf{Z}_{\mathbf{X},j}^{(L)}}{\sum_{j \in \mathcal{G}_k} \boldsymbol{\alpha}_j}.$$

This also ensures that while the structure is discrete, the contribution of each token remains continuous, allowing gradients to flow back to the feature encoder and capture fine-grained semantic variations.

6 Experiments

6.1 Datasets

We evaluate the multi-vector compression methods on several datasets spanning text, image, and video.

- BEIR [text, 48] is a collection of text retrieval tasks. For our evaluation we chose the set of publicly available datasets with corpora of fewer than 1M documents. We further exclude Quora, as it is a duplicate retrieval task in which the "documents" are duplicate questions, and therefore not in need or amenable to compression. The remaining datasets span medical, financial, and argument domains.
- ViDoRe v2 [visual document, 36] is a visual document retrieval benchmark designed to evaluate systems on visually rich PDFs where information is conveyed through both text and layout (e.g., figures, tables). It consists of four datasets spanning insurance, biomedical, economics, and ESG domains, featuring long-form and cross-document queries that require multimodal understanding.

Method	BEIR		ViDoRe		MSR-VTT		MultiVENT 2.0	
	R@10	nDCG@10	R@1	nDCG@5	R@1	nDCG@10	R@10	nDCG@10
Baseline	37.1	46.2	27.7	60.0	55.7	71.9	−*	−*
SeqResize	35.8	<u>43.9</u>	23.5	51.7	53.3	69.9	41.1	38.5
	96.5%	95.0%	84.8%	86.2%	95.7%	96.9%	N/A	N/A
MemTok	<u>36.3</u>	**45.0**	25.0	54.4	<u>54.2</u>	69.9	48.7	44.8
	97.8%	97.4%	90.3%	90.7%	97.3%	96.9%	N/A	N/A
H-Pool	35.5	41.2	<u>26.0</u>	<u>56.4</u>	54.1	<u>70.4</u>	<u>49.2</u>	**46.5**
	95.7%	89.2%	93.9%	94.0%	97.1%	97.6%	N/A	N/A
AGC (Ours)	**37.0**	**45.0**	**26.3**	**56.7**	**56.9**	**71.5**	**49.6**	<u>46.3</u>
	99.7%	97.4%	94.9%	94.5%	102.1%	99.2%	N/A	N/A

Table 1: Results of index compression on each retrieval benchmark. Compression budgets: BEIR 32, ViDoRe 64, MSR-VTT 32, MultiVENT 2.0 64. ∗ means baseline was unable to build due to compute. Second row of each method shows percent of baseline.

- MSR-VTT [vision-only video, 54] is a video captioning dataset converted to text-to-video retrieval, where each query is a sentence description of a video. There is only one relevant video per query and 1000 query-video pairs in test with no additional irrelevant videos.
- MultiVENT 2.0 [audiovisual video, 23] is a text-to-video retrieval dataset with queries that target visual and audio information. There are ten relevant videos per query with 2546 queries and 109,800 videos in test.

6.2 Experimental Setup

BEIR. We begin finetuning from an already finetuned single-vector encoder [29, 57].[3] We train for 10,000 steps with a distillation loss on 16-way MSMARCO [3] hard negatives scored by a reranker.[4] We train with a batch size of 20, learning rate of 10^{-4}, and bfloat16 precision. In both the training and evaluation settings, we use a maximum query length of 32 (with the usual ColBERT-style query augmentation mechanism), maximum document length of 300, and no query/document marker tokens. For retrieval, document embeddings are indexed and retrieved in a FastPlaid [43, 46] index with 4-bit residuals.

ViDoRe v2. We enable bidirectional attention and initialize the pretrained weights from Qwen2.5-VL-3B. We train on the ColPali train set[5] for 2 epochs with a global batch size of 112 and gradient accumulation step of 4, learning rate of 10^{-5}, and bfloat16 precision. We prepend "Passage: " and "Query: " for document and query respectively. Due to the combination of the large embedding dimension (2048) and quantity of embeddings (> 1000 per document, uncompressed), we cannot fit our full uncompressed data in a FastPlaid index, and therefore resort to a brute-force search over a flat index. For the compressed methods, we are able to use a FastPlaid index, but we continue to use the flat index for compression methods to fairly compare with the baselines.

MSR-VTT. We again enable bidirectional attention and initialize the pretrained weights from Qwen2.5-VL-3B, Qwen2.5-VL-7B, and Qwen3-VL-4B for different variants. We use a fixed number of frames of 24. We train on the MSR-VTT Train 9k split for 2 epochs with a global batch size of 28 and gradient accumulation step of 4, learning rate of 10^{-5}, and bfloat16 precision. We build a flat index of each method to gather the matching positions and strengths for analysis in subsection 6.5.

MultiVENT 2.0. We enable bidirectional attention and initialize the pretrained weights from Qwen2.5-Omni-3B. We train on a combination of the human written queries and synthetically generated queries [45]. We sample frames at most 24 frames and audio at 4KHz. We train for 2 epochs with a global batch size of 8 and gradient accumulation step of 4, learning rate of 10^{-5}, and bfloat16 precision. We cannot build any index over the uncompressed representations, and only utilize FastPlaid for the compressed representations.

Evaluation. For each dataset, we report appropriate recall at k (R@k) and normalized discounted cumulative gain at k (nDCG@k). We also report the percentage of base performance for each compression method calculated as $(\frac{\text{compression score}}{\text{base score}})$ and report the compression ratio as $1 - (\frac{\text{budget}}{\text{avg toks per doc}})$.

6.3 Results

In Table 1, we summarize the retrieval performance of the index compression methods in retrieval settings across each modality. Our main finding is that AGC performs the best across the modalities compared to the other compression techniques, maintaining 97% of the uncompressed model performance at nDCG@10. We also find that H-Pool performs well for non-parametric compression, often outperforming the other learned methods SeqResize and MemTok on non-text benchmarks.

Across the datasets, we find that the only method able to outperform the base model, which builds a full index with a one-to-one mapping between document tokens and vectors, is AGC (R@1 on MSR-VTT). This highlights two main takeaways. (1) Training multimodal retrieval methods with a compression objective can be beneficial. Multimodal (audio and visual) tokens do not always provide new information to the document representations and are often semantically redundant[6], meaning that information density does not scale linearly with document length, an assumption reasonably

[3] `Alibaba-NLP/gte-modernbert-base`
[4] `lightonai/ms-marco-en-bge-gemma`
[5] `vidore/colpali_train_set`
[6] Visual compression has traditionally relied on this trend [4].

Meth.	Avg	NF	FQA	SciF	SciD	TC	Tou	Arg
Base	46.2	37.1	43.3	74.4	18.6	78.5	27.5	44.1
Seq	43.9	31.0	36.8	70.3	18.7	75.8	24.8	50.0
	95.0	83.6	85.0	94.5	**100.5**	96.6	90.2	**113.4**
Mem	45.0	35.2	39.8	71.7	17.6	77.6	27.7	45.1
	97.4	94.9	**91.9**	96.4	94.6	98.9	**100.7**	102.3
H-P	41.2	33.7	36.2	72.5	18.4	62.0	17.6	48.3
	89.1	90.8	83.6	**97.4**	98.9	79.0	64.0	109.5
AGC	45.0	36.0	38.9	72.4	18.1	78.7	23.3	47.6
	97.4	**97.0**	89.8	97.3	97.3	**100.3**	84.7	107.9

Table 2: nDCG@10 (top) and percent performance relative to uncompressed baseline (bottom) on BEIR datasets. NF: NFCorpus, FQA: FiQA, SciF: SciFact, SciD: SciDocs, TC: TREC-COVID, Tou: Touche, Arg: arguana.

held in text. (2) The base model underutilizes its full document representation. In subsection 6.5, we show that the base model only utilizes about $\sim$ 1% of the representations. Practical results imply that full indices for multimodal collections offer diminishing returns relative to their cost.

BEIR. In Table 2, we explore the methods' performance for text retrieval on a subset of BEIR datasets, reporting both the absolute nDCG@10 and relative performance compared to the uncompressed ColBERT baseline. With a budget of 32 tokens, documents in BEIR are compressed by around 80%.[7] We find that there is not much difference between MemTok and AGC. Both methods compress the text document representations well and maintain stable performance across the different datasets. Additionally, we find that there is a larger gap between H-Pool and the learned methods, as the performance varies more significantly depending on the task.

ViDoRe. In Table 3, we break down the performance of each method on the ViDoRe topic splits. Comparing the runs with the same training configurations, we see that AGC and H-Pool significantly outperform SeqResize and MemTok. H-Pool and AGC appear to be relatively equivalent, with their averages only differing by 0.002 in nDCG@5. However, when looking at the breakdown by topic, we see that AGC is more stable across domains than H-Pool. We also compare to another learned compression method, MetaEmbed [53]. We find that AGC and H-Pool have comparable or better performance to MetaEmbed. This highlights the strengths of both AGC and H-Pool, even when training at smaller scales.[8]

MSR-VTT. In Table 4, we report more detailed comparisons of our baseline and method on MSR-VTT. We find that every compression method sets a new state-of-the-art on MSR-VTT over previous multi-vector approaches (ColQwen-Omni, Video-ColBERT) and dense approaches (OmniEmbed), even when compressing the index to only 5 vectors per document. We even see at budgets of 32 and 128, AGC performs better at R@1 than the base model, again

[7]NFCorpus: 87% (avg 237 toks), FiQA: 76% (avg 134 toks), SciFact: 86% (avg 230 toks), SciDocs: 83% (avg 188 toks), TREC-COVID: 81% (avg 170 toks), Touche: 79% (avg 153 toks), Arguana: 82% (avg 177 toks)

[8]We note that the comparison to MetaEmbed is not 1-to-1 because we are only capable of training at $\frac{1}{20}$th of the scale.

Method	Tok	Avg	Bio	Econ	ESG-R	ESG-H
Base	1297	60.0	61.4	53.9	57.0	67.6
ColPali	-	53.3	56.5	49.9	55.7	51.1
CQO	-	56.5	56.5	53.2	54.2	62.2
MEmbed	64	58.8	58.7	55.5	57.4	63.7
SeqResize	64	51.7	54.7	53.5	45.2	53.5
MemTok	64	54.3	56.8	53.0	46.4	**61.4**
H-Pool	64	56.4	**59.6**	52.1	53.4	60.6
AGC	64	**56.7**	59.0	**54.5**	**55.8**	57.3

Table 3: nDCG@5 breakdown by domain on ViDoRe v2 for the multilingual subsets. Bio: Biomedical, Econ: Economics, ESG-R: ESG Reports, ESG-H: ESG Human. ESGs are English. CQO: ColQwenOmni, MEmbed: MetaEmbed.

Tok	Method	R@1	R@10	nDCG@10
1	OmniEmbed-7B [33]	51.5	83.2	67.1
26	Video-ColBERT [42]	51.5	85.5	67.7
1702	ColQwen-Omni 3B	40.8	73.8	56.3
1318	Baseline 3B (Ours)	**55.7**	**88.3**	**71.9**
5	SeqResize	53.4	86.7	**69.5**
	MemTok	52.7	**87.3**	69.3
	H-Pool	52.6	86.1	68.9
	AGC	**53.9**	85.8	69.2
32	SeqResize	53.3	86.9	69.9
	MemTok	54.2	86.4	69.9
	H-Pool	54.1	**87.3**	70.4
	AGC	**56.9**	87.0	**71.5**
128	SeqResize	52.6	**87.6**	69.7
	MemTok	54.9	86.7	70.5
	H-Pool	54.4	87.2	70.9
	AGC	**56.4**	**87.6**	**71.6**

Table 4: Retrieval performance on MSR-VTT. We compare compressed methods against uncompressed baselines at budgets of 5, 32, and 128 tokens, alongside SOTA models.

showing that training for compression in multimodal retrieval may not only lead to efficient indices, but also best performance.

MultiVENT 2.0. In Table 1, we report the retrieval results for MultiVENT 2.0 on the compression methods. We are not able to build the index for the full model as comparable vision indices use a hidden dimension of 2048 [38, 42, 53, 55], demonstrating that for large multimodal indices compression is necessary. Unlike ViDoRe and MSR-VTT, evaluating on MultiVENT 2.0 requires leveraging the audio information to retrieve the relevant videos. We found inefficient audio sampling to be a major limitation of the Qwen-Omni model, suggesting interesting future work on how to pass audiovisual signal efficiently to MLLMs. For example, when training on the MultiVENT 2.0 data, we found that in order to fit a batch size of 8, the audio signal needed to be reduced from 16KHz (Qwen-Omni's training rate) to 4KHz to fit on the devices.[9]

[9]Sampling audio below 4KHz degrades speech intelligibility [2].

Tok	Appn Tok	R@1	R@10	nDCG@10
5	5	53.9	85.8	69.2
	32	53.1	86.7	69.6
32	5	54.2	87.2	70.6
	32	56.9	87.0	71.5
	128	55.4	87.6	71.2
128	32	55.2	86.8	70.8
	128	56.4	87.6	71.6

Table 5: Retrieval performance metrics of AGC across varying budgets and appending tokens on MSR-VTT. Each combination is used with the same configurations of training. Appn Tok: number of appending tokens.

Method	Train	Test	R@1	R@10	nDCG@10
Baseline	1318	1318	55.7	88.3	71.9
AGC	32	5	**53.6**	**87.4**	**70.1**
	32	32	**56.9**	87.0	**71.5**
	32	128	**56.4**	**87.5**	**71.7**
H-Pool	1318	5	52.6	86.1	68.9
	1318	32	54.1	**87.3**	70.4
	1318	128	54.4	87.2	70.9

Table 6: Generalizability of AGC and H-Pool compression methods on MSR-VTT.

6.4 Compression Ranges and Stability

Compression Ranges. In Table 4, we explore learning different ranges of compression on MSR-VTT for 5 tokens (99.62%), 32 tokens (97.57%) and 128 tokens (90.29%). For SeqResize, we see that performance seems to be relatively flat across the compression ratios, with similar R@1 and nDCG@10 results. This is an interesting finding, largely suggesting that SeqResize may underutilize the budget and lead to a suboptimal index (see further evidence in subsection 6.5). For all other methods, we see an increase in performance from the most extreme compression to lighter ratios. Additionally, we find H-Pool's performance impressive as a non-parametric technique at a budget of 5. However, AGC continues to have strongest performance at each ratio, demonstrating its robustness to a variety of compression ratios.

Budget Sweeps. In Table 5, we analyze the impact of varying token budgets and the number of appending tokens of AGC on retrieval performance on MSR-VTT. We observe that performance scales positively with both the size of the token budget and the quantity of appending tokens. Notably, even under the most extreme compression setting (a budget of 5), AGC maintains robust performance, outperforming the single dense vector encoder, OmniEmbed-7B, despite using a smaller 3B backbone. When looking at the number of appended query tokens and the budget, we find that it is generally optimal to align the number of appended query tokens and the budget size. Additionally, we see that 32 appended query tokens and a budget of 5 outperforms 5 appended query tokens at the same budget, but we don't see the same pattern for 128 appended query tokens and 32 budget. This suggests that it is important to avoid a low number of query tokens, but that performance doesn't necessarily scale to the number of appended query tokens at any budget.

Model Variant	R@1	R@10	nDCG@10
Qwen2.5-VL-3B	56.9	87.0	71.5
Qwen2.5-VL-7B	58.0	89.0	73.0
Qwen3-VL-4B	58.5	88.4	73.0

Table 7: Generalizability of AGC across different model sizes and variants. Results indicate consistent performance scaling with larger and newer model backbones.

Compression Transferability. In Table 6, we again explore different ranges of compression on MSR-VTT, but only train the model for a single compression ratio (i.e., training AGC for 32). We find that AGC provides the best ability to generalize to an unseen compression ratio after training. We attribute this finding to a weakness in the heuristic of H-Pool, redundant visual tokens should not be equally merged. By leveraging attention to select centroids and weight the merging, AGC is better able to preserve salient semantic concepts and reduce the redundancy along the temporal dimension. We do not compare to MemTok and SeqResize in this experiment as they do not generalize to new compression ratios.

Looking closer at the AGC results, we observe very close performance between methods trained for budgets of 5 and 128 and the method trained for 32 and tested on those budgets. This demonstrates that our model can transfer abilities between budgets at performance near training AGC for that compression ratio.

Model Size and Backbone. In Table 7, we also provide an experiment on the stability of our method with a different model size and with a different underlying model. When scaling AGC to a 7B parameter model, initializing from Qwen2.5-VL-7B, we find that performance is significantly improved. Additionally, we initialize our model from Qwen3-VL-4B and find that performance again improves over both Qwen2.5 versions. These results demonstrate that our method benefits from models with stronger representations and stronger encoding abilities, and should scale to any backbone or model size.

6.5 Index Utilization

In Figure 3, we visualize the difference in index utilization for full uncompressed indices and the four compression techniques and the similarity between document vectors in each index on MSR-VTT. To calculate the matching strength, we sum the maximum similarity scores across all relevant query-document pairs in MSR-VTT and normalize by the total number of match records at that query position. For the heatmaps, we calculate the cosine similarity between token vectors within each document, averaged across documents in the index of MSR-VTT.

Token Utilization Analysis. Our analysis of index statistics reveals that of the 1.3 million unique document tokens, only $\sim 1\%$ are active during a single evaluation pass, with the base model primarily utilizing the first 2%. This again stresses that building full indices for multimodal collections is unnecessary and that these indices

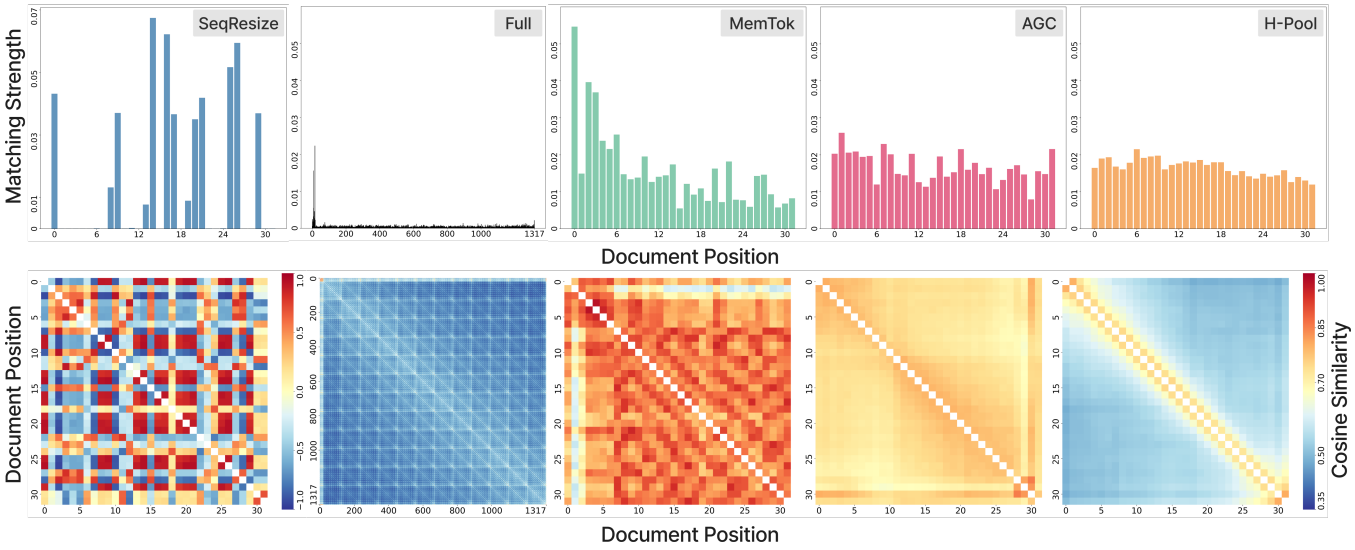

Figure 3: Index utilization and inter-position similarity analysis on MSR-VTT. Top row: Per-position matching strength for each method, computed by summing the maximum similarity matches between query tokens and document tokens across all relevant query-document pairs, averaged over query positions. Bottom row: Pairwise cosine similarity between document vectors within each document, averaged across all documents in the index.

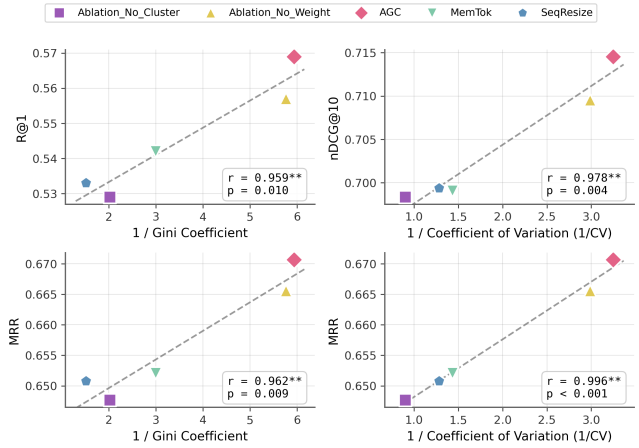

Figure 4: Correlation between retrieval performance metrics and distribution evenness measures on MSR-VTT dataset. Dashed lines indicate linear regression fits. All correlations are statistically significant ($p \leq 0.01$), with Pearson's r ranging from 0.959 to 0.996.

can greatly benefit from compression. For SeqResize, we see a unique trend amongst the compression methods, only selecting a few tokens from the document representation in late interaction. This underutilization of the budget corroborates the findings of Table 4, where SeqResize's performance seemed to plateau across the compression ratios. For MemTok and AGC, we see that both methods attempt to utilize their full document representations. Because MemTok's representations are appended to the document context, we see a significant bias towards the first few tokens in the representation. This result largely follows the trend in dense encoding with causal language models to append a token at the end of the sequence to use as the document representation [31, 34]. Unlike MemTok, AGC and H-Pool use representations from the document leading to a better utilization of the compressed representations in late interaction.

Token Similarity Analysis. In the heatmaps of Figure 3, we further investigate the internal structure of these indices by visualizing the cosine similarity between document tokens. For the full model, we observe that the first few tokens, which we previously noted dominate late interaction, exhibit a consistently high similarity ($\sim$ 0.7) to nearly all other tokens in the document. Additionally, these initial tokens are highly similar to one another. This similarity explains the significant imbalance in matching strength observed in the corresponding bar chart. SeqResize presents a distinct pattern where tokens that are never used in interaction display negative similarity. We interpret this as a modeling failure; tokens derived from the same document context should theoretically maintain a baseline degree of positive similarity, which SeqResize fails to capture. Conversely, MemTok demonstrates an over-smoothing problem, where the heatmap is dominated by high similarity scores. This lack of diversity restricts the expressive power of the index, as the tokens fail to capture distinct semantic nuances. H-Pool, by design, merges similar tokens and consequently produces the most diverse set of representations, as evidenced by the lower off-diagonal similarities. However, this suggests that similarity-based heuristics alone are not sufficient for optimal performance, as H-Pool does not perform as well as learned methods in many settings despite its high diversity. Finally, AGC shows a trend similar to H-Pool but maintains decent inter-token similarities. This balance highlights the efficacy of our approach: it avoids the representation collapse seen in MemTok while preserving necessary semantic overlaps that H-Pool lacks, resulting in a robust compressed index.

Metric	CV		Gini	
	Pearson r	p-value	Pearson r	p-value
R@1	0.974^{**}	0.005	0.959^{**}	0.010
nDCG@10	0.978^{**}	0.004	0.943^{*}	0.016
MRR	0.996^{**}	<0.001	0.962^{**}	0.009

Table 8: Pearson correlation analysis between retrieval metrics and inverse evenness metrics (1/evenness) on MSR-VTT, testing the hyperbolic relationship retrieval ~ 1/evenness. All variants use a fixed budget of 32. Evenness metrics measured on matching strength of (document position, query position) pairs. CV: Coefficient of Variation, Gini: Gini Coefficient. Significance levels: $^{**}p < 0.01$, $^{*}p < 0.05$.

Predicting Performance with Utilization. Following from the above observations, we explore if it is possible to predict compression performance by only looking at how evenly distributed the strength of maximum similarity matches are in a document representation. We calculate the Coefficient of Variation (CV) and the Gini coefficient. The CV assesses relative variability standardized by the mean $CV = \frac{\sigma}{\mu} \times 100$, while the Gini coefficient quantifies distributional concentration $G = \frac{2\sum ix_i}{n\sum x_i} - \frac{n+1}{n}$. For both metrics, lower values indicate a more evenly distributed activations.

In Table 8 and Figure 4, we show the Pearson's correlation [40] between the retrieval metrics (R@1, nDCG@10, and MRR) and the inverse evenness metrics (Coefficient of Variation (CV) and Gini Coefficient).[10] We find a rough correlation between the evenness of the distribution of maximum similarity matches and retrieval performance. These results suggest that training late interaction methods to maximize the utility of each token in its document representations will lead to strong performance, which we leave for future work to explore. Additionally, this finding suggests that during development it is satisfactory to estimate the downstream performance of a compression method with the distribution of maximum similarity matches on a small set of queries. This is especially beneficial for multimodal tasks, where building indices is storage and time expensive.

6.6 Method Ablation

In Table 9, we perform an ablation analysis on the modeling choices in AGC using MSR-VTT. We examine the contribution of three components: Attention-based Centroid Selection, Clustering, and Weighted Aggregation. First, removing the attention weights (w/o Attn Weight) from the aggregation step leads to a decline in performance. As discussed in subsection 5.3, weighting the contribution of tokens by their saliency scores is beneficial for balancing the hard assignment operation with optimization stability. Without these weights, the contribution of individual tokens becomes less continuous, rendering the optimization landscape rougher and less effective. Second, we analyze the impact of attention-based selection (w/o Attn Select) by replacing the learned universal query tokens with a random selection strategy. This prevents the model from distinguishing signal from noise. While this randomness helps maintain some diversity, it forces the model to compress mixed information into each vector. This saturates the capacity of each vector and weakens its ability to capture complex, discriminative semantics. Finally, we evaluate the model without clustering (w/o Cluster), relying solely on attention selection. Without the clustering operation to reduce redundancy and aggregate context, the resulting representations lack diversity and expressiveness. Consequently, token matches in one evaluation pass become highly concentrated on a narrow set of tokens, leading to worse performance. As shown in Figure 4, the progressive removal of Weighted Aggregation and Clustering leads to a simultaneous decline in retrieval performance and evenness of MaxSim matches. This confirms that AGC effectively incentivizes balanced utilization while improving retrieval.

[10] Because of the difficulty in obtaining different indices, we compute these correlations with 5 samples. Larger scale exploration is needed to further validate this finding.

Method	R@1	R@10	nDCG@10
AGC	56.9	87.0	71.5
w/o Attn Weight	55.7	86.5	71.0
w/o Attn Select	54.1	86.8	70.0
w/o Cluster	52.9	87.3	69.8

Table 9: Ablation Study Results on MSR-VTT. We observe a drop in performance as components are removed (individually, not in sequence) from the full AGC model.

7 Conclusion

In this work, we explore multi-vector index compression in any modality. We adapt three strong text-based compression methods to multimodal documents: SeqResize (projection-based), MemTok (token-based), and H-Pool (heuristic-based), and introduce AGC, a novel approach to multi-vector compression. AGC uses three main aspects—attention-based centroid selection, clustering, and weighted aggregation—to maximize the utility of a fixed document token budget. We find that AGC is a strong and robust method to compress documents in any modality under a variety of compression ratios, domains, and model specifications. AGC consistently outperforms the other compression methods across the modalities and even sets a new state-of-the-art on MSR-VTT. Finally, we visualize how each method utilizes its index, demonstrating that AGC and H-Pool properly utilize their budgets, and show that downstream retrieval performance roughly correlates to how well utilized a document representation is in late interaction.

Future Work. While our work corroborates the understanding that multi-vector embeddings can be compressed with a high compression ratio while retaining most retrieval performance, the budget is applied statically or potentially linearly in the case of H-Pool. A natural extension would be to develop compression mechanisms that allocate the budget in proportion to a document's inherent informational content, perhaps by using lightweight features like our document token utilization to calibrate the level of compression.

Acknowledgments

This material is based upon work supported by the NSF Graduate Research Fellowship under Grant No. DGE2139757. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] Antonio Acquavia, Craig Macdonald, and Nicola Tonellotto. 2023. Static Pruning for Multi-Representation Dense Retrieval. In *Proceedings of the ACM Symposium on Document Engineering 2023 (DocEng '23)*. Association for Computing Machinery, New York, NY, USA, 1–10. doi:10.1145/3573128.3604896

[2] Thomas Baer, Brian C. J. Moore, and Karolina Kluk. 2002. Effects of Low Pass Filtering on the Intelligibility of Speech in Noise for People with and without Dead Regions at High Frequencies. *The Journal of the Acoustical Society of America* 112, 3 Pt 1 (Sept. 2002), 1133–1144. doi:10.1121/1.1498853

[3] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. arXiv:1611.09268 [cs] doi:10.48550/arXiv.1611.09268

[4] H. B. Barlow. 2001. Redundancy reduction revisited. *Network: Computation in Neural Systems* 12 (2001), 241 – 253. https://api.semanticscholar.org/CorpusID:18767856

[5] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. 2022. Token Merging: Your ViT But Faster. In *The Eleventh International Conference on Learning Representations*.

[6] Vivek Chari and Benjamin Van Durme. 2025. Compactor: Calibrated Query-Agnostic KV Cache Compression with Approximate Leverage Scores. arXiv:2507.08143 [cs] doi:10.48550/arXiv.2507.08143

[7] Haonan Chen, Sicheng Gao, Radu Timofte, Tetsuya Sakai, and Zhicheng Dou. 2026. E5-Omni: Explicit Cross-modal Alignment for Omni-modal Embeddings. arXiv:2601.03666 [cs] doi:10.48550/arXiv.2601.03666

[8] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 2024. An Image Is Worth 1/2 Tokens After Layer 2: Plug-and-Play Inference Acceleration for Large Vision-Language Models. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LXXXI*. Springer-Verlag, Berlin, Heidelberg, 19–35. doi:10.1007/978-3-031-73004-7_2

[9] Benjamin Clavié, Antoine Chaffin, and Griffin Adams. 2024. Reducing the Footprint of Multi-Vector Retrieval with Minimal Performance Impact via Token Pooling. arXiv:2409.14683 [cs] doi:10.48550/arXiv.2409.14683

[10] Kuicai Dong, Yujing Chang, Derrick Goh Xin Deik, Dexun Li, Ruiming Tang, and Yong Liu. 2025. MMDocIR: Benchmarking Multimodal Retrieval for Long Documents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (Eds.). Association for Computational Linguistics, Suzhou, China, 30971–31005. doi:10.18653/v1/2025.emnlp-main.1576

[11] Yan Fang, Jingtao Zhan, Yiqun Liu, Jiaxin Mao, Min Zhang, and Shaoping Ma. 2022. Joint Optimization of Multi-vector Representation with Product Quantization. In *Natural Language Processing and Chinese Computing*, Wei Lu, Shujian Huang, Yu Hong, and Xiabing Zhou (Eds.). Springer International Publishing, Cham, 631–642. doi:10.1007/978-3-031-17120-8_49

[12] Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Celine Hudelot, and Pierre Colombo. 2024. ColPali: Efficient Document Retrieval with Vision Language Models.

[13] Michael Günther, Saba Sturua, Mohammad Kalim Akram, Isabelle Mohr, Andrei Ungureanu, Bo Wang, Sedigheh Eslami, Scott Martens, Maximilian Werk, Nan Wang, and Han Xiao. 2025. Jina-Embeddings-v4: Universal Embeddings for Multimodal Multilingual Retrieval. In *Proceedings of the 5th Workshop on Multilingual Representation Learning (MRL 2025)*, David Ifeoluwa Adelani, Catherine Arnett, Duygu Ataman, Tyler A. Chang, Hila Gonen, Rahul Raja, Fabian Schmidt, David Stap, and Jiayi Wang (Eds.). Association for Computational Linguistics, Suzhuo, China, 531–550. doi:10.18653/v1/2025.mrl-main.36

[14] Georg Heigold, Ehsan Variani, Tom Bagby, Cyril Allauzen, Ji Ma, Shankar Kumar, and Michael Riley. 2026. Massive Sound Embedding Benchmark (MSEB). arXiv:2602.07143 [cs] doi:10.48550/arXiv.2602.07143

[15] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. 2017. Localizing Moments in Video with Natural Language. In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, Venice, 5804–5813. doi:10.1109/ICCV.2017.618

[16] Sebastian Hofstätter, Omar Khattab, Sophia Althammer, Mete Sertkan, and Allan Hanbury. 2022. Introducing Neural Bag of Whole-Words with ColBERTer: Contextualized Late Interactions Using Enhanced Reduction. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management (CIKM '22)*. Association for Computing Machinery, New York, NY, USA, 737–747. doi:10.1145/3511808.3557367

[17] Xiaohu Huang, Hao Zhou, and Kai Han. 2025. PruneVid: Visual Token Pruning for Efficient Video Large Language Models. In *Findings of the Association for Computational Linguistics: ACL 2025*, Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (Eds.). Association for Computational Linguistics, Vienna, Austria, 19959–19973. doi:10.18653/v1/2025.findings-acl.1024

[18] Herve Jégou, Matthijs Douze, and Cordelia Schmid. 2011. Product Quantization for Nearest Neighbor Search. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 1 (Jan. 2011), 117–128. doi:10.1109/TPAMI.2010.57

[19] Rohan Jha, Bo Wang, Michael Günther, Georgios Mastrapas, Saba Sturua, Isabelle Mohr, Andreas Koukounas, Mohammad Kalim Akram, Nan Wang, and Han Xiao. 2024. Jina-ColBERT-v2: A General-Purpose Multilingual Late Interaction Retriever. In *Proceedings of the Fourth Workshop on Multilingual Representation Learning (MRL 2024)*, Jonne Sälevä and Abraham Owodunni (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 159–166. doi:10.18653/v1/2024.mrl-1.11

[20] Ziyan Jiang, Rui Meng, Xinyi Yang, Semih Yavuz, Yingbo Zhou, and Wenhu Chen. 2024. VLM2Vec: Training Vision-Language Models for Massive Multimodal Embedding Tasks.

[21] Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Virtual Event China, 39–48. doi:10.1145/3397271.3401075

[22] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-Captioning Events in Videos. In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, Venice, 706–715. doi:10.1109/ICCV.2017.83

[23] Reno Kriz, Kate Sanders, David Etter, Kenton Murray, Cameron Carpenter, Hannah Recknor, Jimena Guallar-Blasco, Alexander Martin, Eugene Yang, and Benjamin Van Durme. 2025. MultiVENT 2.0: A Massive Multilingual Benchmark for Event-Centric Video Retrieval. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Nashville, TN, USA, 24149–24158. doi:10.1109/CVPR52734.2025.02249

[24] Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, et al. 2022. Matryoshka representation learning. *Advances in Neural Information Processing Systems* 35 (2022), 30233–30249.

[25] Carlos Lassance, Maroua Maachou, Joohee Park, and Stéphane Clinchant. 2021. A Study on Token Pruning for ColBERT. arXiv:2112.06540 [cs] doi:10.48550/arXiv.2112.06540

[26] Didier Le Gall. 1991. MPEG: A Video Compression Standard for Multimedia Applications. *Commun. ACM* 34, 4 (April 1991), 46–58. doi:10.1145/103085.103090

[27] Mingxin Li, Yanzhao Zhang, Dingkun Long, Keqin Chen, Sibo Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2026. Qwen3-VL-Embedding and Qwen3-VL-Reranker: A Unified Framework for State-of-the-Art Multimodal Retrieval and Ranking. arXiv:2601.04720 [cs] doi:10.48550/arXiv.2601.04720

[28] Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. 2024. SnapKV: LLM Knows What You Are Looking for before Generation. In *Proceedings of the 38th International Conference on Neural Information Processing Systems (NIPS '24, Vol. 37)*. Curran Associates Inc., Red Hook, NY, USA, 22947–22970.

[29] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards General Text Embeddings with Multi-stage Contrastive Learning. arXiv:2308.03281 [cs] doi:10.48550/arXiv.2308.03281

[30] Ting Liu, Liangtao Shi, Richang Hong, Yue Hu, Quanjun Yin, and Linfeng Zhang. 2024. Multi-Stage Vision Token Dropping: Towards Efficient Multimodal Large Language Model. arXiv:2411.10803 [cs] doi:10.48550/arXiv.2411.10803

[31] Zheng Liu, Chaofan Li, Shitao Xiao, Yingxia Shao, and Defu Lian. 2024. Llama2Vec: Unsupervised Adaptation of Large Language Models for Dense Retrieval. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 3490–3500. doi:10.18653/v1/2024.acl-long.191

[32] Maxime Louis, Hervé Déjean, and Stéphane Clinchant. 2025. PISCO: Pretty Simple Compression for Retrieval-Augmented Generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (Eds.). Association for Computational Linguistics, Vienna, Austria, 15506–15521. doi:10.18653/v1/2025.findings-acl.800

[33] Xueguang Ma, Luyu Gao, Shengyao Zhuang, Jiaqi Samantha Zhan, Jamie Callan, and Jimmy Lin. 2025. Tevatron 2.0: Unified Document Retrieval Toolkit across Scale, Language, and Modality. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25)*. Association for Computing Machinery, Padua, Italy, 4061–4065. doi:10.1145/3726302.3730135

[34] Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2024. Fine-Tuning LLaMA for Multi-Stage Text Retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24)*. Association for Computing Machinery, New York, NY, USA, 2421–2425. doi:10.1145/3626772.3657951

[35] Sean MacAvaney, Antonio Mallia, and Nicola Tonellotto. 2025. Efficient Constant-Space Multi-vector Retrieval. In *Advances in Information Retrieval*, Claudia Hauff, Craig Macdonald, Dietmar Jannach, Gabriella Kazai, Franco Maria Nardini, Fabio Pinelli, Fabrizio Silvestri, and Nicola Tonellotto (Eds.). Springer Nature Switzerland, Cham, 237–245. doi:10.1007/978-3-031-88714-7_22

[36] Quentin Macé, António Loison, and Manuel Faysse. 2025. ViDoRe Benchmark V2: Raising the Bar for Visual Retrieval. arXiv:2505.17166 [cs] doi:10.48550/arXiv.2505.17166

[37] Ryan McGrady. 2024. What We Discovered on 'Deep YouTube'. *The Atlantic* (26 January 2024). https://www.theatlantic.com/technology/archive/2024/01/how-many-videos-youtube-research/677250/ Accessed: 2025-01-11.

[38] Gabriel de Souza P. Moreira, Ronay Ak, Mengyao Xu, Oliver Holworthy, Benedikt Schifferer, Zhiding Yu, Yauhen Babakhin, Radek Osmulski, Jiarui Cai, Ryan Chesler, Bo Liu, and Even Oldridge. 2026. Nemotron ColEmbed V2: Top-Performing Late Interaction Embedding Models for Visual Document Retrieval. arXiv:2602.03992 [cs] doi:10.48550/arXiv.2602.03992

[39] Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive Text Embedding Benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, Andreas Vlachos and Isabelle Augenstein (Eds.). Association for Computational Linguistics, Dubrovnik, Croatia, 2014–2037. doi:10.18653/v1/2023.eacl-main.148

[40] Karl Pearson. 1895. VII. Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London* 58 (1895), 240 – 242. https://api.semanticscholar.org/CorpusID:121644161

[41] Guanghui Qin and Benjamin Van Durme. 2023. Nugget: Neural Agglomerative Embeddings of Text. In *Proceedings of the 40th International Conference on Machine Learning*. PMLR, 28337–28350.

[42] Arun Reddy, Alexander Martin, Eugene Yang, Andrew Yates, Kate Sanders, Kenton Murray, Reno Kriz, Celso M. De Melo, Benjamin Van Durme, and Rama Chellappa. 2025. Video-ColBERT: Contextualized Late Interaction for Text-to-Video Retrieval. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Nashville, TN, USA, 19691–19701. doi:10.1109/CVPR52734.2025.01834

[43] Keshav Santhanam, Omar Khattab, Christopher Potts, and Matei Zaharia. 2022. PLAID: An Efficient Engine for Late Interaction Retrieval. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. ACM, Atlanta GA USA, 1747–1756. doi:10.1145/3511808.3557325

[44] Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. 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*. Association for Computational Linguistics, Seattle, United States, 3715–3734. doi:10.18653/v1/2022.naacl-main.272

[45] Tyler Skow, Alexander Martin, Benjamin Van Durme, Rama Chellappa, and Reno Kriz. 2026. RANKVIDEO: Reasoning Reranking for Text-to-Video Retrieval. arXiv:2602.02444 [cs] doi:10.48550/arXiv.2602.02444

[46] Raphaël Sourty. 2025. FastPlaid: A High-Performance Engine for Multi-Vector Search. https://github.com/lightonai/fast-plaid

[47] Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. 2025. DyCoke : Dynamic Compression of Tokens for Fast Video Large Language Models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Nashville, TN, USA, 18992–19001. doi:10.1109/CVPR52734.2025.01769

[48] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models. *NeurIPS Datasets and Benchmarks* abs/2104.08663 (2021).

[49] João Veneroso, Rajesh Jayaram, Jinmeng Rao, Gustavo Hernández Ábrego, Majid Hadian, and Daniel Cer. 2025. CRISP: Clustering Multi-Vector Representations for Denoising and Pruning. arXiv:2505.11471 [cs] doi:10.48550/arXiv.2505.11471

[50] David Wan, Han Wang, Elias Stengel-Eskin, Jaemin Cho, and Mohit Bansal. 2025. CLaMR: Contextualized Late-Interaction for Multimodal Content Retrieval. arXiv:2506.06144 [cs] doi:10.48550/arXiv.2506.06144

[51] Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. 2019. VaTeX: A Large-Scale, High-Quality Multilingual Dataset for Video-and-Language Research. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Seoul, Korea (South), 4580–4590. doi:10.1109/ICCV.2019.00468

[52] Joe H. Ward. 1963. Hierarchical Grouping to Optimize an Objective Function. *J. Amer. Statist. Assoc.* 58, 301 (March 1963), 236–244. doi:10.1080/01621459.1963.10500845

[53] Zilin Xiao, Qi Ma, Mengting Gu, Chun-cheng Jason Chen, Xintao Chen, Vicente Ordonez, and Vijai Mohan. 2025. MetaEmbed: Scaling Multimodal Retrieval at Test-Time with Flexible Late Interaction. arXiv:2509.18095 [cs] doi:10.48550/arXiv.2509.18095

[54] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. MSR-VTT: A Large Video Description Dataset for Bridging Video and Language. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Las Vegas, NV, USA, 5288–5296. doi:10.1109/CVPR.2016.571

[55] Mengyao Xu, Gabriel Moreira, Ronay Ak, Radek Osmulski, Yauhen Babakhin, Zhiding Yu, Benedikt Schifferer, and Even Oldridge. 2025. Llama Nemoretriever Colembed: Top-Performing Text-Image Retrieval Model. arXiv:2507.05513 [cs] doi:10.48550/arXiv.2507.05513

[56] Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. 2025. VisionZip: Longer Is Better but Not Necessary in Vision Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Nashville, TN, USA, 19792–19802.

[57] Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, Meishan Zhang, Wenjie Li, and Min Zhang. 2024. mGTE: Generalized Long-Context Text Representation and Reranking Models for Multilingual Text Retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, Franck Dernoncourt, Daniel Preoţiuc-Pietro, and Anastasia Shimorina (Eds.). Association for Computational Linguistics, Miami, Florida, US, 1393–1412. doi:10.18653/v1/2024.emnlp-industry.103

[58] Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A. Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. 2025. SparseVLM: Visual Token Sparsification for Efficient Vision-Language Model Inference. In *Proceedings of the 42nd International Conference on Machine Learning*. PMLR, 74840–74857.

[59] Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. 2023. H2O: Heavy-Hitter Oracle for Efficient Generative Inference of Large Language Models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems (NIPS '23)*. Curran Associates Inc., Red Hook, NY, USA, 34661–34710.

[60] Jiaying Zhu, Yurui Zhu, Xin Lu, Wenrui Yan, Dong Li, Kunlin Liu, Xueyang Fu, and Zheng-Jun Zha. 2025. VisionSelector: End-to-End Learnable Visual Token Compression for Efficient Multimodal LLMs. arXiv:2510.16598 [cs] doi:10.48550/arXiv.2510.16598

[61] Yuxuan Zong and Benjamin Piwowarski. 2025. Towards Lossless Token Pruning in Late-Interaction Retrieval Models. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25)*. Association for Computing Machinery, New York, NY, USA, 2407–2417. doi:10.1145/3726302.3730100