MINER: MINING MULTIMODAL INTERNAL REPRESENTATION FOR EFFICIENT RETRIEVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual document retrieval has become essential for accessing information in visually rich documents. Existing approaches fall into two camps. Late-interaction retrievers achieve strong quality through fine-grained token-level matching but store hundreds of vectors per page, incurring large index footprints and high serving costs. By contrast, dense single-vector retrievers retain storage and latency advantages but consistently lag in quality because they compress all information into a single final-layer embedding. In this work, we first conduct a layerwise diagnostic on single-vector retrievers, revealing that retrieval-relevant signal resides in internal representations. Motivated by these findings, we propose **MINER** (**M**ining Multimodal **I**nternal Represe**N**tation for **E**fficient **R**etrieval), a lightweight plug-in module that probes and fuses internal signals across transformer layers into a single compact embedding without modifying the backbone or sacrificing single-vector efficiency. The first *Retrieval-Aligned Layer Probing* stage attaches a lightweight probe at each layer, surfacing which dimensions carry retrieval-relevant information. The subsequent *Adaptive Sparse Multi-Layer Fusion* stage applies performance-adaptive neuron-level masking to the selected layers and fuses the surviving signals into the final dense vector. Across ViDoRe V1/V2/V3, MINER outperforms existing dense single-vector retrievers on the majority of benchmarks, with up to 4.5% nDCG@5 improvement over its corresponding backbone. In settings where models support both late-interaction and dense retrieval, MINER substantially narrows the nDCG@5 performance gap while retaining the storage and serving efficiency of dense retrieval. Our source code is available at this anonymous repository.

1 INTRODUCTION

Visually rich documents such as presentation slides, scanned reports, and scientific posters convey information through a tightly coupled combination of text, layout, and figures (Ding et al., 2026). Accessing information in such documents requires retrieval systems that can jointly reason over visual and textual cues (Gao et al., 2025; Ma et al., 2024). This need has driven growing interest in visual document retrieval, where the goal is to retrieve relevant document pages directly from their rendered images, bypassing the fragile OCR-and-chunking pipelines that traditional text-based retrieval systems depend on (Faysse et al., 2025; Yu et al., 2025).

Existing visual document retrieval methods largely fall into two families. Late-interaction retrievers, originating from ColBERT (Khattab & Zaharia, 2020) and extended to the multimodal setting by ColPali (Faysse et al., 2025), represent each query and document as a set of token-level embeddings and compute relevance through fine-grained matching. This preserves rich local signals and yields strong retrieval quality, but at a substantial cost: each document page may require storing over a thousand patch-level vectors, leading to large index footprints and expensive scoring at retrieval time (Ma et al., 2025; Santhanam et al., 2022). Dense single-vector retrievers take the opposite approach, compressing each input into a single embedding and scoring by a simple dot product (Karpukhin et al., 2020; Lin et al., 2025; Günther et al., 2025). This yields compact indices and efficient serving, but the compression into one final-layer vector creates an information bottleneck: retrieval-relevant signals present in intermediate representations may be discarded before reaching the output embedding.

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.

A growing body of recent work suggests that the internal representations of deep networks carry richer task-relevant information than their final outputs alone (Evci et al., 2022; Tu et al., 2023; Zhang et al., 2024). In natural language processing, SPIN (Jiao et al., 2024) and SIREN (Jiao et al., 2026) have demonstrated that intermediate layers encode semantically useful information that benefits downstream tasks such as text classification. For vision tasks, Perception Encoder (Bolya et al., 2025) shows that the strongest visual embeddings often emerge before the last layer rather than at the network output. These findings raise a natural question for multimodal retrieval:

Can internal representations of single-vector retrievers be leveraged to recover the retrieval-relevant information lost to the final layer bottleneck, thereby improving retrieval quality without sacrificing single-vector efficiency?

To answer this question, we first conduct a systematic layerwise analysis of single-vector retrievers. Using normalized Centered Kernel Alignment (CKA) (Kornblith et al., 2019) and a novel alignment-ratio diagnostic, we reveal that retrieval-relevant signal does reside in internal representations, but is distributed unevenly across depths. Earlier layers contain useful but misaligned structure, while later layers are progressively more aligned with the final retrieval space. This analysis yields a principled criterion for selecting and partitioning the layers that are most amenable to lightweight extraction.

Motivated by these findings, we propose **MINER** (**M**ining Multimodal **I**nternal Represe**N**tation for **E**fficient **R**etrieval), a lightweight plug-in module that enhances dense retrieval embeddings by selectively extracting and fusing retrieval-relevant signals from internal layers. MINER operates in two stages. The first stage, *Retrieval-Aligned Layer Probing*, attaches a lightweight shared probe at each selected layer and trains it to align per-layer text and vision readouts to their cross-modal final-layer anchors. Guided by the principled criterion we derived from our layerwise analysis, we select layers that contain richer retrieval-relevant signals and partition them into two sets. Layers in the structurally directly aligned regime are probed with simple element-wise reweighting, which is referred to as *Base Probing*. In contrast, layers in the structurally less aligned regime receive a more expressive row-normalized linear projection, termed *Normalized Projection Probing*. The second stage, *Adaptive Sparse Multi-Layer Fusion*, applies performance-adaptive neuron-level masking to the probed layers and aggregates the surviving signals into a single dense embedding through a learned cross-layer weighted sum with a global bias. Importantly, without modifying the backbone model, MINER is agnostic to the readout mechanism (EOS-token or pooling) and preserves the storage and serving efficiency of single-vector retrievers. Across ViDoRe V1/V2/V3 (Faysse et al., 2025; Macé et al., 2025; Loison et al., 2026), MINER consistently outperforms existing dense single-vector retrievers and, in several settings, substantially narrows the gap to strong late-interaction baselines.

In summary, our contributions are as follows:

1. We reveal that retrieval-relevant signal in single-vector retrievers is distributed unevenly across transformer layers, and that different layers require qualitatively different extraction strategies. We introduce normalized CKA and alignment ratio as principled diagnostic tools.
2. We propose MINER, a plug-in module guided by the layerwise diagnostic, comprising Retrieval-Aligned Layer Probing and Adaptive Sparse Multi-Layer Fusion, which improves retrieval quality without modifying the backbone or increasing storage and serving cost.
3. MINER consistently improves dense retrieval quality across multiple backbones and the suite of ViDoRe benchmarks, achieving up to 4.5% nDCG@5 gain over the dense baseline, while matching its storage footprint and incurring only $1.07\times$ query latency overhead, substantially narrowing the gap to late-interaction methods at $42.4\times$ lower index cost.

2 Related Work

We review two lines of work most relevant to MINER: visual document retrieval methods and studies on internal representations for retrieval.

Visual document retrieval has increasingly moved away from OCR-then-chunking pipelines toward direct page-image retrieval. There are two main camps for existing methods. *(1)* Late-interaction retrievers, originating from ColBERT (Khattab & Zaharia, 2020), have been especially influential in this transition, with ColPali (Faysse et al., 2025) demonstrating strong retrieval performance on

visually rich documents using VLM-based multi-vector representations. This line has been further developed through more realistic and challenging benchmarks such as ViDoRe V2 and ViDoRe V3 (Macé et al., 2025; Loison et al., 2026; Dong et al., 2025; Chen et al., 2025b), as well as follow-up studies on reproducibility and storage efficiency (Qiao et al., 2025; Ma et al., 2025). In parallel, *(2)* single-vector retrievers (Karpukhin et al., 2020) remain attractive because of its compact indexing and efficient serving. FLMR (Lin et al., 2023) explores fine-grained multimodal retrieval for retrieval-augmented visual question answering (VQA), and MM-Embed (Lin et al., 2025) studies universal multimodal retrieval with multimodal LLMs. On the model side, Jina-embeddings-v4 (Günther et al., 2025) and Eager-Embed-v1 (Balarini, 2025) represent recent dense multimodal embedding systems designed for practical retrieval. Compared with late interaction, these single-vector methods preserve the efficiency advantages of indexing and serving, but they typically lag in retrieval quality.

Recent works suggest that useful signals are often distributed across intermediate layers rather than concentrated only in the final output. Perception Encoder (Bolya et al., 2025) shows that strong visual embeddings may emerge before the final network output, while recent dense retrieval work demonstrates that multi-layer representations can improve retrieval quality beyond standard last-layer pooling (Xie & Lukasiewicz, 2025). SPIN (Jiao et al., 2024) and SIREN (Jiao et al., 2026) further show that probing techniques for sparse selection and integration of internal neurons can improve downstream task performance.

Overall, prior works have improved multimodal retrieval either by increasing matching granularity through late interaction or by training stronger dense embedding models. In contrast, guided by recent studies suggesting that valuable task information resides in internal representations, our work extracts retrieval-relevant internal signals to improve compact single-vector multimodal retrieval.

3 Layerwise Analysis of Retrieval-Relevant Internal Representations

We now turn to the question raised in Section 1. To investigate this, we conduct a layerwise analysis on three single-vector retrievers: Jina-embeddings-v4 (Jina) (Günther et al., 2025), Eager-Embed-v1 (Eager) (Balarini, 2025) and MoCa-3B (MoCa) (Chen et al., 2025a).

Formally, we consider a single-vector retriever with a backbone of L transformer layers. Given an input, let $\mathbf{H}^{(l)}$ denote its hidden states at layer l, and let $r(\cdot)$ be the model-specific readout operator (e.g., EOS-token extraction or pooling). The corresponding layerwise readout is $\mathbf{x}^{(l)} = r(\mathbf{H}^{(l)}) \in \mathbb{R}^D$, where D is the embedding dimension. For each input indexed by i in a paired text-vision dataset of size N, we denote its layer-l readout as $\mathbf{x}_i^{(l)} \in \mathbb{R}^D$ and its corresponding cross-modal final-layer anchor as $\mathbf{a}_i \in \mathbb{R}^D$, defined as the paired input's final-layer readout (i.e., the text final-layer readout when input is a vision input, and vice versa). Our analysis compares each layer-l readout against this cross-modal anchor. Two questions naturally arise from this comparison: *(1)* which layers actually contain retrieval-relevant signals that are amenable to lightweight extraction? and *(2)* should all such layers be treated in the same way, or do different layers call for different extraction strategies? We address these two questions in turn below and conclude with a principled criterion that directly motivates the design of MINER in Section 4.

Which layers are the most amenable to lightweight extraction? A natural diagnostic is to measure how structurally similar the per-layer representation space is to the final cross-modal anchor space. We adopt linear Centered Kernel Alignment (CKA) (Kornblith et al., 2019), which compares two representation spaces at the dataset level rather than pointwise, and is therefore robust to coordinate-level mismatches that are common across layers of deep networks. The detailed formula is in Appendix A. A high CKA_l indicates that the layer-l representation space is structurally similar to the final retrieval space, whereas a low CKA_l suggests that layer l either has not yet developed sufficient retrieval-relevant structure, or expresses it in a geometry too distant from the final retrieval space to be recovered with a lightweight probe. However, using CKA_l directly as a layer-selection criterion is problematic because its absolute scale can vary across backbone models. To obtain a cutoff that is invariant to such scale differences, we min-max normalize CKA_l across all L layers of the backbone and retain only those layers whose normalized score $\widehat{\mathrm{CKA}}_l$ exceeds a cutoff $\tau_{\mathrm{CKA}} \in [0, 1]$:

$$\widehat{\mathrm{CKA}}_l = \frac{\mathrm{CKA}_l - \mathrm{CKA}_{\min}}{\mathrm{CKA}_{\max} - \mathrm{CKA}_{\min}}, \qquad \mathcal{S}_{\mathrm{cand}} = \left\{ l \in \{1, \ldots, L\} : \widehat{\mathrm{CKA}}_l \geq \tau_{\mathrm{CKA}} \right\}, \tag{1}$$

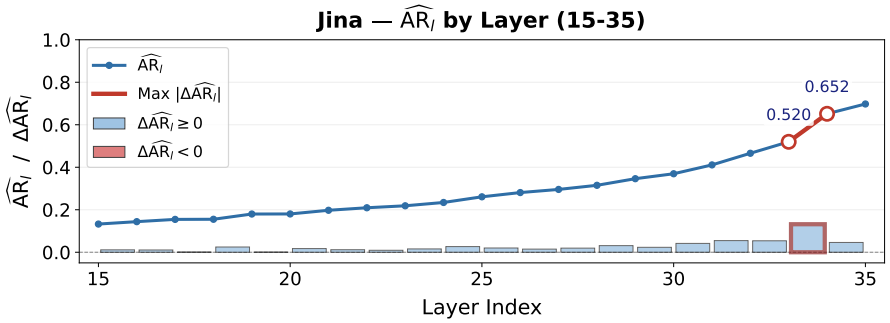

Figure 1: The line shows the layerwise normalized Alignment Ratio ($\widehat{\mathrm{AR}}_l$) of Jina backbone from layer 15 to 35 (the candidate set $\mathcal{S}_{\mathrm{cand}}$ of layers), and the bars show $\Delta\widehat{\mathrm{AR}}_l$, the change in $\widehat{\mathrm{AR}}_l$ from layer $l-1$ to layer l. The largest step change occurs between layer 33 and 34 ($\Delta\widehat{\mathrm{AR}}_{34} = 0.1319$).

where $\mathrm{CKA}_{\min}$ and $\mathrm{CKA}_{\max}$ are the minimum and maximum CKA values across all L layers. This yields a candidate set $\mathcal{S}_{\mathrm{cand}}$ of layers whose representations are structurally close enough to the final retrieval space to be amenable to lightweight extraction. In our study, we set $\tau_{\mathrm{CKA}} = 0.6$, which retains layers whose representations are sufficiently close to the final anchor space. We later analyze the sensitivity of MINER to this choice in Section 5.5.

Should all candidate layers be treated the same way? Structural similarity alone does not determine how easy it is to extract useful signal: a layer may share dataset-level geometry with the anchor space without being directly aligned with it at the sample level. This gap arises because CKA is invariant to orthogonal transformations of the representation space, while sample-level alignment is not. To capture this distinction, we complement the dataset-level CKA with a sample-level alignment measure. Concretely, for each layer l, we compute the average per-sample cosine similarity between its readouts $\{\mathbf{x}_i^{(l)}\}_{i=1}^N$ and their corresponding cross-modal anchors $\{\mathbf{a}_i\}_{i=1}^N$, denoted as $c_l \in [-1, 1]$. Here, c_l quantifies how well, on average, individual layer-l readouts already point in the same direction as their cross-modal anchors. We then define the *alignment ratio* of layer l as $\mathrm{AR}_l = c_l / \mathrm{CKA}_l$. Intuitively, AR_l measures how much of a layer's structural similarity to the anchor space is already realized as direct pointwise alignment. A high AR_l indicates that the layer is not only structurally similar to the final retrieval space but also coordinate-aligned with it, so its signal can be extracted with a simple, geometry-preserving operation. A low AR_l indicates that the useful structure is present but expressed in a less aligned subspace, which calls for a more expressive transformation to realign it before fusion. Figure 1 visualizes the min-max normalized $\widehat{\mathrm{AR}}_l$ and its adjacent-layer change $\Delta\widehat{\mathrm{AR}}_l = \widehat{\mathrm{AR}}_l - \widehat{\mathrm{AR}}_{l-1}$ for Jina, with additional visualizations for Eager and MoCa provided in Appendix B. Within the candidate set $\mathcal{S}_{\mathrm{cand}}$, two distinct regimes emerge. Earlier retained layers exhibit relatively low and slowly varying $\widehat{\mathrm{AR}}_l$, while a sharp transition is identified by the largest positive adjacent-layer change $\Delta\widehat{\mathrm{AR}}_l$. Across all three backbones, this criterion places the final three layers in the higher-alignment regime. These layers are therefore not selected a priori; the transition point is determined independently for each backbone by the same alignment-ratio analysis. This pattern is consistent across all three backbones, as shown for Jina in Figure 1 and further supported by the Eager and MoCa visualizations in Figures 5 and 6. It confirms that candidate layers fall into two qualitatively different alignment regimes, motivating differentiated extraction strategies.

Together, these two diagnostics yield a principled criterion: normalized CKA selects which layers to extract from, and the alignment ratio determines how each selected layer should be treated.

4 Mining Multimodal Internal Representations for Efficient Retrieval

We now formally introduce the problem we aim to address. In visual document retrieval, given a query $q \in \mathcal{Q}$ and a corpus of rendered document pages $\mathcal{D}$, the goal is to retrieve the most relevant document $d \in \mathcal{D}$ for q. A single-vector retriever uses a backbone f, with L transformer layers as introduced in Section 3, to encode the query and document into single embeddings $\mathbf{e}_q = f(q)$ and $\mathbf{e}_d = f(d)$, and

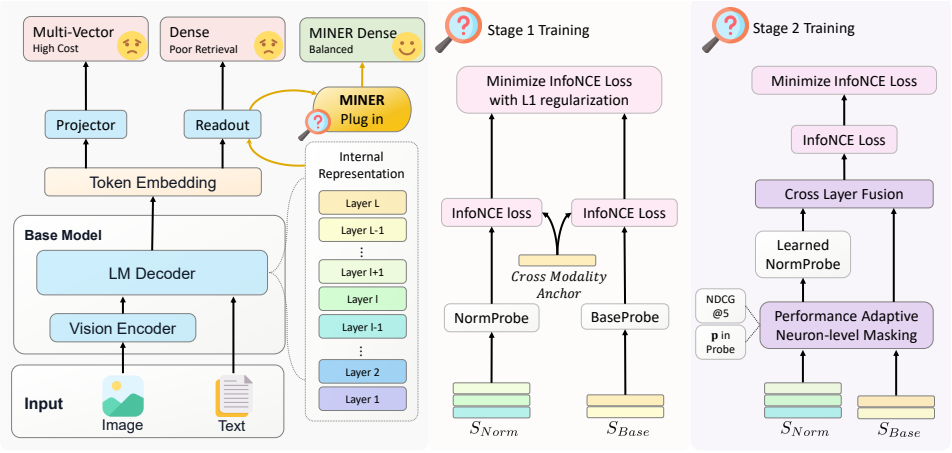

Figure 2: Overview of MINER. MINER acts as a plug-in module that extracts readout-aligned internal representations from the Language Model (LM) decoder and improves the final embedding.

ranks documents by the inner product $s(q,d) = \mathbf{e}_q^\top \mathbf{e}_d$. In standard single-vector retrievers, $\mathbf{e}_q$ and $\mathbf{e}_d$ are taken from the final-layer readout of f. In contrast, we propose **MINER** (**M**ining Multi-Modal **I**nternal Represe**N**tation for **E**fficient **R**etrieval), a lightweight plug-in module that constructs $\mathbf{e}_q$ and $\mathbf{e}_d$ by mining retrieval-relevant signals from multiple internal layers of f, guided by the criterion derived in Section 3. Figure 2 illustrates the overall MINER pipeline. MINER consists of two stages: *Retrieval-Aligned Layer Probing* and *Adaptive Sparse Multi-Layer Fusion*. MINER is agnostic to the readout mechanism of f (EOS-token or pooling) and does not modify f, thereby preserving the storage and serving advantages of single-vector retrieval.

4.1 Retrieval-Aligned Layer Probing

Stage 1 attaches a lightweight probe to each layer in $\mathcal{S}_{\text{cand}}$ identified in Section 3, with the parameterization of each probe determined by the layer's alignment regime, and trains all probes with a shared cross-modal siamese objective. Recall from Section 3 the layerwise readout $\mathbf{x}_i^{(l)} \in \mathbb{R}^D$ of input i at layer l. For each paired text–vision sample (t_i, v_i) in our training set $\{(t_i, v_i)\}_{i=1}^N$, we use the modality-explicit notation $\mathbf{x}_{t_i}^{(l)}$ and $\mathbf{x}_{v_i}^{(l)}$ for the layer-l readouts of the text and vision inputs, with the final-layer readouts $\mathbf{x}_{t_i}^{(L)}$ and $\mathbf{x}_{v_i}^{(L)}$ serving as the text and vision *anchors*. These are the cross-modal anchors $\mathbf{a}_i$ from Section 3, now written in modality-explicit form.

Probe parameterization. Following the alignment-ratio analysis in Section 3, we partition $\mathcal{S}_{\text{cand}}$ into the last three layers, $\mathcal{S}_{\text{base}}$, which are already directly aligned with the final retrieval space, and the remaining earlier layers, $\mathcal{S}_{\text{norm}}$, which are structurally similar but expressed in a less aligned subspace. Each subset is equipped with its own probe parameterization, shared across the text and vision modalities since both pass through the same backbone f and ultimately map into a common retrieval space. For each layer $l \in \mathcal{S}_{\text{base}}$, we apply a *Base Probe (BaseProbe)*, which learns a vector $\mathbf{p}_l \in \mathbb{R}^D$ whose relative magnitudes provide a within-layer importance ranking:

$$\text{BaseProbe}_l(\mathbf{x}) = \mathbf{p}_l \odot \mathbf{x}. \tag{2}$$

This parameterization preserves the original coordinate system and is appropriate when the layer is already coordinate-aligned with the anchor space, so only neuron-level reweighting is needed. For each layer $l \in \mathcal{S}_{\text{norm}}$, we apply a *Normalized Projection Probe (NormProbe)*, which additionally learns a projection matrix $\mathbf{W}_l \in \mathbb{R}^{D\times D}$ and applies element-wise reweighting followed by a row-normalized linear projection:

$$\text{NormProbe}_l(\mathbf{x}) = \tilde{\mathbf{W}}_l(\mathbf{p}_l \odot \mathbf{x}), \qquad \tilde{\mathbf{W}}_l^{(j)} = \frac{\mathbf{W}_l^{(j)}}{\|\mathbf{W}_l^{(j)}\|_2}, \tag{3}$$

where $\mathbf{W}_l^{(j)}$ denotes the j-th row of $\mathbf{W}_l$ and $\tilde{\mathbf{W}}_l^{(j)}$ is ℓ_2-normalized counterpart. The row normalization constrains the scale of the projection weights and allows the less aligned subspace to be realigned to the anchor space, while $\mathbf{p}_l$ provides a relative within-layer importance ranking.

Siamese training objective. Each probe is trained to align its layer-l readout with the *cross-modal* final-layer anchor in a siamese manner: the probed text representation should align with the vision anchor, and vice versa. For notational convenience, we write $\mathrm{probe}_l(\cdot)$ to denote $\mathrm{BaseProbe}_l(\cdot)$ if $l \in \mathcal{S}_{\text{base}}$ and $\mathrm{NormProbe}_l(\cdot)$ if $l \in \mathcal{S}_{\text{norm}}$. The probe at layer l then minimizes

$$\mathcal{L}_l = \frac{1}{N}\sum_{i=1}^{N}\Big[\tfrac{1}{2}\,\ell\big(\mathrm{probe}_l(\mathbf{x}_{t_i}^{(l)}),\,\mathbf{x}_{v_i}^{(L)}\big) + \tfrac{1}{2}\,\ell\big(\mathrm{probe}_l(\mathbf{x}_{v_i}^{(l)}),\,\mathbf{x}_{t_i}^{(L)}\big)\Big] + \lambda\|\mathbf{p}_l\|_1, \tag{4}$$

where $\ell(\cdot,\cdot)$ is the InfoNCE loss (van den Oord et al., 2018) computed over in-batch negatives and hard negatives using cosine similarity, and $\lambda \geq 0$ is the coefficient of the ℓ_1 regularizer applied to the importance vector $\mathbf{p}_l$ (Tibshirani, 1996).

4.2 Adaptive Sparse Multi-Layer Fusion

After probing, MINER aggregates the per-layer signals through two steps: *performance-adaptive neuron-level masking*, which retains only the most retrieval-relevant neurons within each layer, and *cross-layer fusion*, which integrates the retained signals into the final single embedding.

Performance-adaptive neuron-level masking. Layers in $\mathcal{S}_{\text{cand}}$ differ in how much they contribute to retrieval, and stronger layers should be assigned higher importance coverage targets. To capture this, we compute a normalized layer utility score α_l for each layer $l \in \mathcal{S}_{\text{cand}}$ based on its standalone validation nDCG@5, $\alpha_l = (\mathrm{nDCG@5}_l - \mathrm{nDCG@5}_{\min})/(\mathrm{nDCG@5}_{\max} - \mathrm{nDCG@5}_{\min})$, where $\mathrm{nDCG@5}_{\min}$ and $\mathrm{nDCG@5}_{\max}$ are the minimum and maximum standalone nDCG@5 across all layers in $\mathcal{S}_{\text{cand}}$. We then convert α_l into a layer-specific Top-P cumulative importance threshold $P_l \in (0,1]$, defined as $P_l = \alpha_l(1-\rho)+\rho$, where $\rho \in (0,1]$ is a floor hyperparameter that guarantees a minimum importance coverage target for every layer. Stronger layers receive larger P_l, corresponding to higher coverage neurons. Using the importance vector p_l learned during probing, we rank the D dimensions in descending order of $|p_l|$ and normalize their absolute weights to sum to one. We then select dimensions in this order until their cumulative normalized importance reaches or exceeds P_l, constructing a hard binary mask $m_l \in \{0,1\}^D$ with 1 for selected dimensions and 0 otherwise. Thus, the probing stage learns relative importance scores over dimensions, whereas the Top-P masking operation is the step that introduces explicit neuron-level sparsity.

Cross-layer fusion. Combining the layer partition from Stage 1 and the masking above, the processed readout $\mathbf{h}^{(l)}$ of an input at layer l is $\mathbf{h}^{(l)} = \mathbf{m}_l \odot \mathbf{x}^{(l)}$ for $l \in \mathcal{S}_{\text{base}}$ or $\mathbf{h}^{(l)} = \tilde{\mathbf{W}}_l\big(\mathbf{m}_l \odot \mathbf{x}^{(l)}\big)$ for $l \in \mathcal{S}_{\text{norm}}$, where $\tilde{\mathbf{W}}_l$ are the parameters learned by NormProbe (Eq. 3). Layers in $\mathcal{S}_{\text{base}}$ contribute directly after masking, whereas layers in $\mathcal{S}_{\text{norm}}$ are first realigned into the anchor space before contributing. The fusion head then learns a per-layer weight matrix $\mathbf{U} \in \mathbb{R}^{|\mathcal{S}_{\text{cand}}| \times D}$, whose row $\mathbf{u}_l$ provides dimension-wise fusion weights for layer l, together with a global bias $\mathbf{b} \in \mathbb{R}^D$. The final dense embedding of an input is then

$$\mathbf{e} = \sum_{l \in \mathcal{S}_{\text{cand}}} \mathbf{u}_l \odot \mathbf{h}^{(l)} + \mathbf{b}. \tag{5}$$

At training time, the fusion head is optimized with the same cross-modal siamese objective as the probes. Given paired samples $\{(t_i, v_i)\}_{i=1}^N$, we minimize $\mathcal{L}_{\text{fusion}} = \frac{1}{N}\sum_{i=1}^{N}\ell\big(\mathbf{e}_{t_i}, \mathbf{e}_{v_i}\big)$, where $\mathbf{e}_{t_i}$ and $\mathbf{e}_{v_i}$ are the fused embeddings of the text and vision inputs computed via Eq. 5, and $\ell(\cdot,\cdot)$ is the InfoNCE loss as in Eq. 4. At inference time, $\mathbf{e}$ instantiates to the fused embeddings of the query q and document d, which are then ranked by the inner product as $s(q,d)$.

5 Experiment

We conduct comprehensive experiments to answer the following research questions: ***RQ1:*** Does MINER improve upon its dense embedding backbone, and how competitive is it with representative late-interaction retrievers? ***RQ2:*** Does MINER retain the storage and retrieval efficiency benefits as standard single-vector retrievers? ***RQ3:*** Are the major design components of MINER necessary? ***RQ4:*** Is MINER robust to key hyperparameters? We study more research questions in Appendix C. The experiment setup is introduced in Section 5.1 first, followed by the results and analysis to each of the research questions.

Table 1: NDCG@5 on ViDoRe V2. **Bold** indicates the best score in each column and underline indicates the second-best score. Blue shading denotes the best single-vector retriever in each column. * indicates that MINER has statistically significant improvement over its corresponding backbone.

Model	Size	Bio	Econ	ESG-Human	ESG-Syn	Avg.
Multi-Vector Retrievers						
ColPali	3B	54.6	48.6	58.5	54.9	54.2
ColQwen2	2B	56.3	50.6	60.4	52.5	55.0
ColQwen2.5	3B	59.2	53.3	**66.4**	**58.3**	**59.3**
Jina(LI)	4B	60.9	51.9	65.1	52.5	57.6
Single-Vector Retrievers						
SigLip	877M	27.7	18.6	44.5	35.0	31.4
VLM2Vec	4B	33.2	23.3	36.3	27.8	30.1
Jina	4B	58.5	55.1	54.8	44.8	53.3
Eager	4B	63.4	56.6	51.7	52.2	56.0
MoCa	3B	59.7	57.0	63.5	53.1	58.3
Ours: MINER Single-Vector Retriever						
MINER-Jina	4B	58.6(+0.2%)	56.8(+3.1%)*	56.8(+3.7%)	48.7(+8.7%)*	55.2(+3.6%)*
MINER-Eager	4B	**64.1**(+1.1%)*	57.8(+2.1%)	59.1(+14.3%)*	52.9(+1.4%)	58.5(+4.5%)*
MINER-MoCa	3B	59.2(−0.8%)	**59.6**(+4.6%)*	64.8(+2.1%)	52.9(−0.4%)	59.1(+1.4%)

5.1 Experiment Setup

We evaluate MINER on three visual document retrieval backbones: Jina-Embeddings-v4 (Jina) (Günther et al., 2025), Eager-Embed-v1 (Eager) (Balarini, 2025), and MoCa-3B (MoCa) (Chen et al., 2025a). Jina supports both dense retrieval and late interaction, enabling a direct comparison between dense, late-interaction, and MINER-enhanced dense embeddings under the same backbone. Eager supports dense single-vector retrieval. MoCa is a bidirectional multimodal embedding backbone obtained through modality-aware continual pre-training followed by heterogeneous contrastive fine-tuning. The three backbones differ in their readout mechanisms: Eager uses an EOS-token readout, whereas Jina and MoCa use pooling-based readouts. This allows us to test whether MINER generalizes across different readout designs. For each backbone, MINER uses the complete original training corpus and validation split, without subsampling or introducing additional training pairs or hard-negative mining. This ensures that any performance gain comes from better utilization of the model's internal representations rather than from introducing additional data. As for the Performance-adaptive neuron-level masking introduced in Section 4.2, it uses the training data's corresponding validation set. Implementation details are provided in Appendix D.

Following GQR (Uzan et al., 2026), we evaluate on the ViDoRe benchmark suite, including ViDoRe V1 (Faysse et al., 2025), V2 (Macé et al., 2025), and V3 (Loison et al., 2026), which is a comprehensive visual document retrieval benchmark suite covering diverse document types, layouts, and query styles. We follow the default evaluation protocol of each benchmark: ViDoRe V1 and V2 are evaluated using nDCG@5 (Järvelin & Kekäläinen, 2002), while ViDoRe V3 is evaluated using nDCG@10. We use ViDoRe V2 as the primary benchmark because ViDoRe V1 is increasingly saturated in recent work, while ViDoRe V3 is relatively new and currently includes fewer established baselines. For fair comparison, we include single-vector retrievers with fewer than 5B parameters that appear on the official ViDoRe leaderboard[1]. Among multi-vector late-interaction retrievers, we report the strongest representative model from each ViDoRe-developed backbone family. Brief descriptions of the baseline models used in our comparisons are provided in Appendix E.

5.2 Results

Table 1 presents the results on ViDoRe V2, while the ViDoRe V1 and V3 result tables are provided in Appendix F (Table 5 and Table 6). Unless otherwise stated, all MINER results use the same

[1] https://huggingface.co/spaces/vidore/vidore-leaderboard

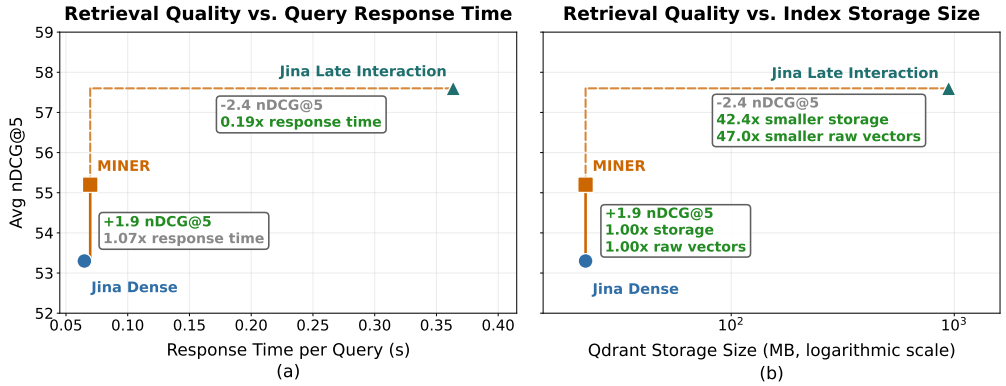

Figure 3: Efficiency-performance trade-off on ViDoRe V2 using Jina, by comparing dense retrieval, late interaction, and MINER across average nDCG@5, query response latency, and storage size.

sparsity floor $\rho = 0.2$ and CKA cutoff $\tau_{\text{CKA}} = 0.6$ across backbones. The statistical significance is evaluated using a paired t-test over per-query retrieval scores between our proposed MINER and its corresponding backbone, where we use the significance level of $\alpha = 0.05$.

Across backbones and benchmark versions, MINER generally improves over its corresponding backbone. On ViDoRe V2, MINER improves Jina by 3.6% and Eager by 4.5% relative to their respective dense backbones. MINER-MoCa also improves the average score from 58.3 to 59.1, corresponding to a 1.4% relative gain. The gains are particularly consistent on ViDoRe V3, where MINER improves all three backbones in average nDCG@10, with all three improvements being statistically significant. MINER-Jina further achieves the best average single-vector score among the compared single-vector retrievers. On ViDoRe V1, even though many subsets are already saturated, MINER-Eager improves its dense backbone from 82.5 to 84.8 nDCG@5, while MINER-Jina and MINER-MoCa remain on par with their strong backbones. To complement these aggregate results, Appendix G provides a query-level analysis of MINER-Jina on ViDoRe V2, where MINER recovers more than twice as many top-5 retrieval failures as it introduces regressions.

Overall, these results answer ***RQ1*** affirmatively: MINER demonstrates its effectiveness across multiple backbones and benchmarks, and in several cases narrows the gap between single-vector retrievers and representative late-interaction retrievers.

5.3 Efficiency Analysis

We evaluate the trade-off between retrieval performance, query-time latency, and storage cost. All efficiency measurements are conducted using Qdrant as the vector database on the full ViDoRe V2 benchmark. For each dataset, we index the complete document corpus and measure the average response time required to retrieve results for all queries. We report the average query response time and the average storage size. Full results are provided in Appendix H.

Figure 3 compares dense retrieval, late interaction, and MINER on Jina, which supports both retrieval modes under the same backbone. MINER produces a single embedding per query/document with the same dimensionality as the dense backbone, so its storage size and search complexity are identical to standard dense retrieval; the only overhead is computing the fused query embedding, resulting in a marginal end-to-end latency increase while improving retrieval quality by $+1.9$ nDCG@5. Compared to late interaction, MINER is $5.3\times$ faster and requires over $40\times$ less index storage, narrowing the quality gap with a substantially better efficiency profile. These results answer ***RQ2***: MINER retains the storage and retrieval efficiency of single-vector dense embeddings while recovering a large portion of the dense-to-late-interaction performance gap.

5.4 Ablation Studies

To answer ***RQ3***, we conduct ablation studies from two complementary perspectives. First, we ablate the major components of MINER on ViDoRe V2. **All Neurons** directly applies *cross-layer fusion* over all selected layers S_{cand}, without using probing and masking components. **All Base** applies *BaseProbe*, *performance-adaptive neuron-level masking* and *cross-layer fusion* over S_{cand}. **All**

Table 2: Ablation study on ViDoRe V2.

Backbone	All Neurons	All Base	All Norm	ConcateLinear	MINER
Jina	54.6	54.9	50.0	49.8	**55.2**
Eager	57.1	57.4	53.1	47.8	**58.5**
MoCa	58.4	58.4	56.0	54.2	**59.1**

Norm applies *NormProbe*, *performance-adaptive neuron-level masking* and *cross-layer fusion* over S_{cand}. The second direction is **ConcateLinear**, which concatenates the representations from all selected layers in $\mathcal{S}_{\text{cand}}$ and learns a linear projection over the concatenated representation. These variants use the same frozen backbone, selected internal layers, and training data as the complete MINER, but remove specific components.

As shown in Table 2, **All Neurons** already improves over the corresponding single-vector baselines in Table 1 across all three backbones, confirming that internal layers carry useful retrieval signal. However, this ablated variant consistently underperforms full MINER, showing that the gains are not obtained simply by fusing all dimensions. The two probing variants further play complementary roles: **All Base** remains below full MINER, while **All Norm** performs worse, indicating that uniformly applying normalized projection to every selected layer can distort representations that are already well aligned. **ConcateLinear** provides a stronger capacity-based control: it accesses full representations from all selected layers and uses more trainable parameters than MINER, yet performs substantially worse across all three backbones. This shows that MINER's gains stem from targeted neuron selection and structured cross-layer fusion rather than greater capacity. These ablations answer ***RQ3***: both sparse neuron selection and the design of *BaseProbe* and *NormProbe* are important to MINER's effectiveness. We also visualize the final neuron-selection in Appendix I.

5.5 Sensitivity Analysis

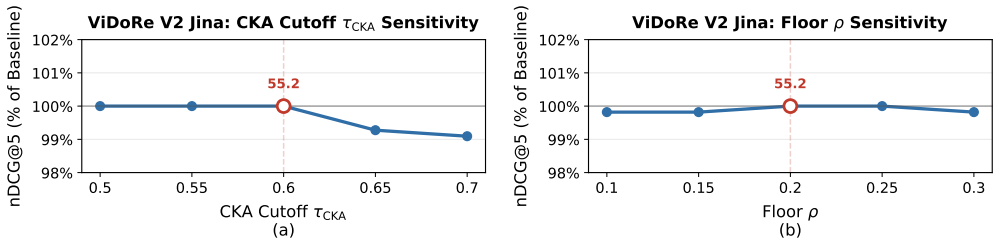

Figure 4: Hyperparameters sensitivity analysis, reported as a percentage of the default configuration.

We finally evaluate whether MINER is robust to key hyperparameters. We focus on two hyperparameters: the CKA cutoff τ_{CKA} for selecting candidate layers and the sparsity floor ρ used in dynamic Top-P masking. We conduct the sensitivity analysis on MINER-Jina over ViDoRe V2 and report performance as a percentage of the default setting, where the default configuration uses CKA cutoff 0.6 and floor $\rho = 0.2$.

As shown in Figure 4, MINER remains stable across a wide range of CKA cutoffs τ_{CKA} and sparsity floors ρ, indicating that the method is not highly sensitive to the exact number of selected layers or to the precise retention floor. We observe the same trend on MINER-Eager and MINER-MoCa. Additional sensitivity analysis plots for MINER with these two are provided in Appendix J. These results answer ***RQ4*** affirmatively: MINER is robust to the key hyperparameters.

6 Conclusion and Discussion

We presented MINER, a lightweight plug-in module for improving dense visual document retrieval by extracting retrieval-useful information from internal LM-decoder representations. Across multiple backbones and ViDoRe benchmarks, MINER consistently improves dense retrieval quality, narrows the gap to late-interaction methods, and preserves the storage and retrieval efficiency advantages of dense embeddings. We discuss the future work and limitations in Appendix K.

AI USE STATEMENT

We used generative AI tools to help implement portions of the training pipeline, identify related papers and candidate models for our experiments, and improve the manuscript. For writing, the tools helped identify unclear passages, suggest more precise wording for claims, and correct grammar and spelling. The authors made the final decisions about references and experimental models, reviewed and tested AI-assisted code, and reviewed the manuscript edits. We take responsibility for the final content of this paper.

ETHICS STATEMENT

Our experiments use existing pretrained backbones, their associated training and validation data, and established ViDoRe benchmarks. We did not recruit human participants, collect a new dataset, or release new data as part of this work.

REPRODUCIBILITY STATEMENT

Our paper can be easily reproduced since, in Sections 3 and 4, we fully explain the layer-selection diagnostics, probing objectives, masking rule, and fusion method. Further, in Section 5.1, we released the backbones we used, their training and validation splits, benchmarks, and evaluation metrics. In Appendix D, we also released the full training setups. Appendices F, H, and J provide additional benchmark, efficiency, and sensitivity results. We also released the source code for our implementations via an anonymous repository linked in the abstract.

REFERENCES

Juan Pablo Balarini. Eager embed v1: Multimodal dense embeddings for retrieval, 2025.

Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Abdul Rasheed, Junke Wang, Marco Monteiro, Hu Xu, Shiyu Dong, Nikhila Ravi, Shang-Wen Li, Piotr Dollar, and Christoph Feichtenhofer. Perception Encoder: The best visual embeddings are not at the output of the network. In *Advances in Neural Information Processing Systems*, 2025.

Haonan Chen, Hong Liu, Yuping Luo, Liang Wang, Nan Yang, Furu Wei, and Zhicheng Dou. MoCa: Modality-aware continual pre-training makes better bidirectional multimodal embeddings. *ArXiv preprint*, abs/2506.23115, 2025a.

Jian Chen, Ming Li, Jihyung Kil, Chenguang Wang, Tong Yu, Ryan Rossi, Tianyi Zhou, Changyou Chen, and Ruiyi Zhang. VisR-Bench: An empirical study on visual retrieval-augmented generation for multilingual long document understanding. *ArXiv preprint*, abs/2508.07493, 2025b.

Yihao Ding, Soyeon Caren Han, Jean Lee, and Eduard Hovy. Deep learning based visually rich document content understanding: A survey. *Artificial Intelligence Review*, 2026.

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*, pp. 30971–31105, 2025.

Utku Evci, Vincent Dumoulin, Hugo Larochelle, and Michael C Mozer. Head2toe: Utilizing intermediate representations for better transfer learning. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 6009–6033. PMLR, 2022.

Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. ColPali: Efficient document retrieval with vision language models. In *International Conference on Learning Representations*, 2025.

Sensen Gao, Shanshan Zhao, Xu Jiang, Lunhao Duan, Yong Xien Chng, Qing-Guo Chen, Weihua Luo, Kaifu Zhang, Jia-Wang Bian, and Mingming Gong. Scaling beyond context: A survey of multimodal retrieval-augmented generation for document understanding. *ArXiv preprint*, abs/2510.15253, 2025.

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

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.

Ziyan Jiang, Rui Meng, Xinyi Yang, Semih Yavuz, Yingbo Zhou, and Wenhu Chen. VLM2Vec: Training vision-language models for massive multimodal embedding tasks. In *International Conference on Learning Representations*, 2025.

Difan Jiao, Yilun Liu, Zhenwei Tang, Daniel Matter, Jürgen Pfeffer, and Ashton Anderson. SPIN: Sparsifying and integrating internal neurons in large language models for text classification. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 4666–4682, 2024.

Difan Jiao, Yilun Liu, Ye Yuan, Zhenwei Tang, Linfeng Du, Haolun Wu, and Ashton Anderson. LLM safety from within: Detecting harmful content with internal representations. *ArXiv preprint*, abs/2604.18519, 2026.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 6769–6781, 2020.

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

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97. PMLR, 2019.

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. Qwen3-VL-Embedding and Qwen3-VL-Reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. *ArXiv preprint*, abs/2601.04720, 2026.

Sheng-Chieh Lin, Chankyu Lee, Mohammad Shoeybi, Jimmy Lin, Bryan Catanzaro, and Wei Ping. MM-Embed: Universal multimodal retrieval with multimodal LLMs. In *International Conference on Learning Representations*, 2025.

Weizhe Lin, Jinghong Chen, Jingbiao Mei, Alexandru Coca, and Bill Byrne. Fine-grained late-interaction multi-modal retrieval for retrieval augmented visual question answering. In *Advances in Neural Information Processing Systems*, 2023.

António Loison, Quentin Macé, Antoine Edy, Victor Xing, Tom Balough, Gabriel Moreira, Bo Liu, Manuel Faysse, Céline Hudelot, and Gautier Viaud. ViDoRe V3: A comprehensive evaluation of retrieval augmented generation in complex real-world scenarios. *ArXiv preprint*, abs/2601.08620, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Xueguang Ma, Sheng-Chieh Lin, Minghan Li, Wenhu Chen, and Jimmy Lin. Unifying multimodal retrieval via document screenshot embedding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 6492–6505, 2024.

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

Quentin Macé, António Loison, and Manuel Faysse. ViDoRe benchmark V2: Raising the bar for visual retrieval. *ArXiv preprint*, abs/2505.17166, 2025.

Qdrant Team. Qdrant: Vector similarity search engine and vector database, 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*, pp. 3335–3345, 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*, pp. 3715–3734. Association for Computational Linguistics, 2022.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 01 1996. ISSN 0035-9246.

Cheng-Hao Tu, Zheda Mai, and Wei-Lun Chao. Visual query tuning: Towards effective usage of intermediate representations for parameter and memory efficient transfer learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7725–7735, 2023.

Omri Uzan, Asaf Yehudai, Roi Pony, Eyal Shnarch, and Ariel Gera. Guided query refinement: Multimodal hybrid retrieval with test-time optimization. In *International Conference on Learning Representations*, 2026.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *ArXiv preprint*, abs/1807.03748, 2018.

Zhongbin Xie and Thomas Lukasiewicz. Investigating multi-layer representations for dense passage retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 24522–24536, 2025.

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 *International Conference on Learning Representations*, 2025.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Taolin Zhang, Jiawang Bai, Zhihe Lu, Dongze Lian, Genping Wang, Xinchao Wang, and Shu-Tao Xia. Parameter-efficient and memory-efficient tuning for vision transformer: a disentangled approach. In *European Conference on Computer Vision*, pp. 346–363. Springer, 2024.

A Additional Formula

Centered Kernel Alignment. For each layer l, we stack the readouts $\{\mathbf{x}_i^{(l)}\}_{i=1}^N$ into a matrix $\mathbf{X}^{(l)} \in \mathbb{R}^{N\times D}$, and likewise stack the corresponding cross-modal anchors $\{\mathbf{a}_i\}_{i=1}^N$ into $\mathbf{A} \in \mathbb{R}^{N\times D}$. The linear CKA between the layer-l representation space and the anchor space, denoted CKA_l, is then defined as

$$\mathrm{CKA}_l = \frac{\left\|(\mathbf{X}^{(l)})^\top \mathbf{A}\right\|_F^2}{\left\|(\mathbf{X}^{(l)})^\top \mathbf{X}^{(l)}\right\|_F \left\|\mathbf{A}^\top \mathbf{A}\right\|_F}, \tag{6}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, $\mathbf{X}^{(l)}$ and $\mathbf{A}$ are column-centered across samples before computing equation 6.

B Additional Layer-wise Analysis

Normalization of the alignment ratio. Since c_l is based on cosine similarity and is bounded in $[-1, 1]$, while $\mathrm{CKA}_l \in [0, 1]$, the raw alignment ratio $\mathrm{AR}_l = c_l/\mathrm{CKA}_l$ can in principle take negative values when the average cosine alignment is negative. Empirically, however, we do not observe negative alignment ratios for any of the candidate layers across the three backbones studied. To obtain a bounded and comparable measure across layers and backbones, we use the min–max normalized alignment ratio $\widehat{\mathrm{AR}}_l$ in our layerwise analysis. This normalization maps the observed alignment ratios to $[0, 1]$ while preserving their relative ordering and the layerwise transition patterns used by our analysis.

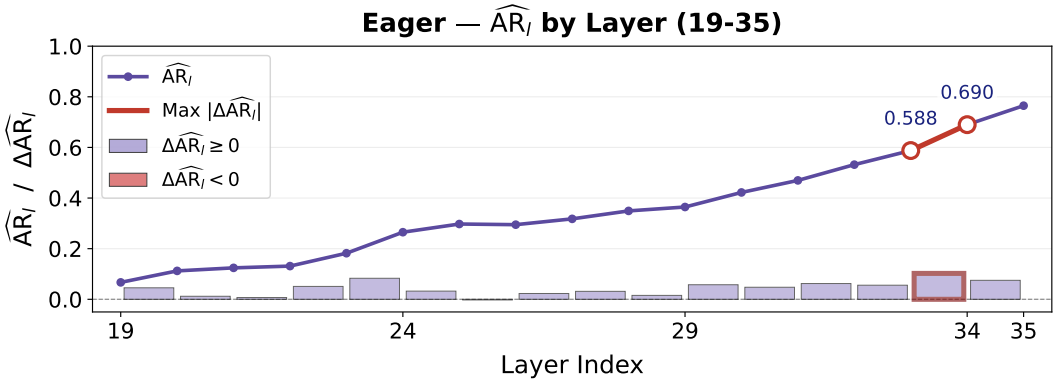

Figure 5: The line shows the layerwise normalized Alignment Ratio ($\widehat{\mathrm{AR}}_l$) of Eager backbone from layer 19 to 35 (the candidate set S_{cand} of layers), and the bars show $\Delta\widehat{\mathrm{AR}}_l$, the change in $\widehat{\mathrm{AR}}_l$ from layer $l-1$ to layer l. The largest step occurs between layers 33 and 34 ($\Delta\widehat{\mathrm{AR}}_{34} = 0.1023$).

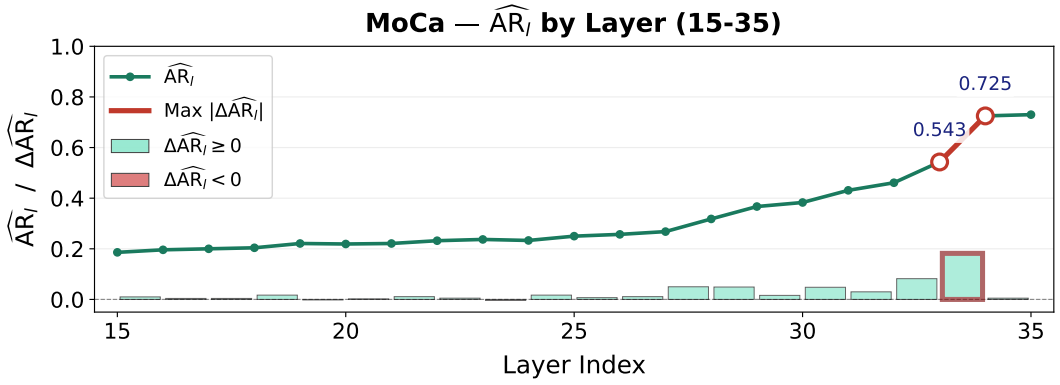

Figure 6: The line shows the layerwise normalized Alignment Ratio ($\widehat{\mathrm{AR}}_l$) of MoCa backbone from layer 15 to 35 (the candidate set S_{cand} of layers), and the bars show $\Delta\widehat{\mathrm{AR}}_l$, the change in $\widehat{\mathrm{AR}}_l$ from layer $l-1$ to layer l. The largest single-step change occurs between layers 33 and 34 ($\Delta\widehat{\mathrm{AR}}_{34} = 0.1820$).

C Additional Research Questions

RQ5: Does feature misalignment occur across internal layers, and does the proposed alignment ratio reliably characterize it?

We evaluate whether the proposed alignment ratio captures the expected change in layerwise feature alignment after retrieval-aligned probing. Figure 7, Figure 8 and Figure 9 compare the normalized alignment ratio AR_n before and after probing on their respective backbones. Across different backbones and among the analyzed internal layers, probing consistently increases AR_n, indicating that the learned probes make intermediate representations more aligned with the final cross-modal retrieval space. Useful retrieval information exists in internal layers, but it may be misaligned relative to the final embedding space. The consistent increase in AR_n after probing suggests that our NormProbe can effectively re-align some of the misaligned layers to the final embedding space.

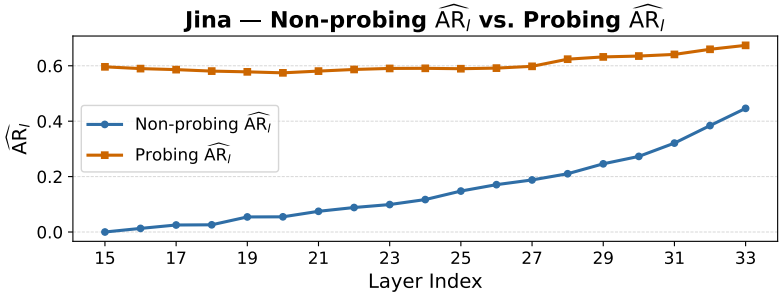

Figure 7: Layerwise normalized alignment ratio $\widehat{\mathrm{AR}}_l$ before and after retrieval-aligned probing on Jina. Probing consistently increases $\widehat{\mathrm{AR}}_l$ across the analyzed internal layers, indicating that intermediate representations become more aligned with the final cross-modal retrieval space after the learned alignment step.

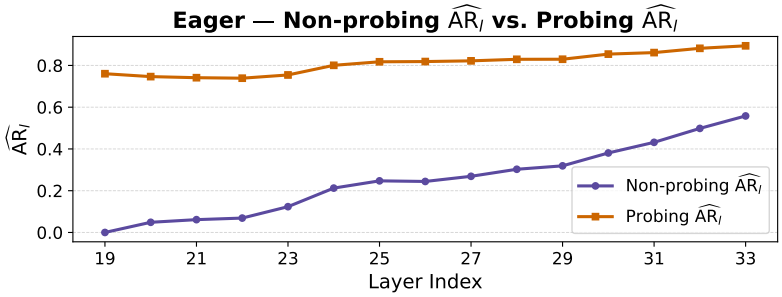

Figure 8: Layerwise normalized alignment ratio $\widehat{\mathrm{AR}}_l$ before and after retrieval-aligned probing on Eager. Probing consistently increases $\widehat{\mathrm{AR}}_l$ across the analyzed internal layers, indicating that intermediate representations become more aligned with the final cross-modal retrieval space after the learned alignment step.

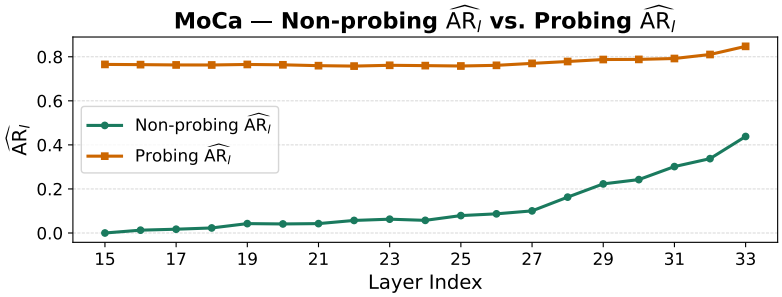

Figure 9: Layerwise normalized alignment ratio $\widehat{\mathrm{AR}}_l$ before and after retrieval-aligned probing on MoCa. Probing consistently increases $\widehat{\mathrm{AR}}_l$ across the analyzed internal layers, indicating that intermediate representations become more aligned with the final cross-modal retrieval space after the learned alignment step.

D Training Details

Table 3: Training details for the probing stage.

Backbone	Probes	Batch	Epochs	Warm up	LR	$\lambda_{\ell_1}(p_\ell)$	GPUs	Time
Jina	Base + WNorm	1024	40	1 epoch	2×10^{-4}	3×10^{-4}	2× RTX 4090	17h
Eager	Base + WNorm	1024	40	1 epoch	2×10^{-4}	3×10^{-4}	2× RTX 4090	14h
MoCa	Base + WNorm	1024	40	1 epoch	2×10^{-4}	5×10^{-4}	2× RTX 4090	11h

All probing models are trained with AdamW (Loshchilov & Hutter, 2019) for 40 epochs using a batch size of 1024, a learning rate of 2×10^{-4}, and random seed 42. We use a linear warmup for the first epoch and keep the learning rate fixed afterward. The weight decay is set to 0.01 for all trainable parameters except the Hadamard scaling vector p_ℓ, which is instead regularized by the ℓ_1 penalty reported in Table 3. Base and WNorm probes for each backbone are trained jointly with distributed data parallelism on two RTX 4090 GPUs, and the reported training cost corresponds to wall-clock time.

Table 4: Training details for the aggregation head.

Backbone	CKA	S_{norm}	S_{base}	Batch	Epochs	Warm up	LR	Floor	GPUs	Time
Jina	0.6	15–33	34–36	512	40	1 epoch	1×10^{-5}	0.2	1× RTX 4090	8h
Eager	0.6	19–33	34–36	512	40	1 epoch	1×10^{-5}	0.2	1× RTX 4090	5h
MoCa	0.6	15–33	34–36	512	40	1 epoch	1×10^{-5}	0.2	1× RTX 4090	6h

After AR analysis, we partition $\mathcal{S}_{\text{cand}}$ into $\mathcal{S}_{\text{norm}}$ and $\mathcal{S}_{\text{base}}$ for all three models as shown in Table 4. All aggregation heads are trained with AdamW (Loshchilov & Hutter, 2019) for 40 epochs using a batch size of 512, a learning rate of 1×10^{-5}, and random seed 42. We use a linear warmup during the first epoch, after which the learning rate remains fixed. Weight decay is set to 1×10^{-4} for all trainable parameters except the bias term $\mathbf{b}$. Each head is trained on a single RTX 4090 GPU, and training cost is reported as wall-clock time. For consistency, we train all heads for the same number of epochs, despite possible differences in convergence speed.

E Baseline Information

For completeness, we briefly describe the baseline models included in our comparisons.

SigLIP. SigLIP is a CLIP-style vision-language model trained with a sigmoid pairwise loss rather than a softmax contrastive loss, enabling image-text representation learning without global batch-level normalization; we use the shape-optimized SoViT-400M variant pretrained on WebLI at 384×384 resolution (Zhai et al., 2023).

VLM2Vec. VLM2Vec is an instruction-guided multimodal embedding model that converts a pretrained vision-language model into a fixed-dimensional embedding model through contrastive training on the MMEB benchmark; we use the full TIGER-Lab VLM2Vec checkpoint in our dense-retrieval comparison (Jiang et al., 2025).

Jina-Embeddings-v4 (Jina). Jina-Embeddings-v4 is a 4B-parameter multimodal and multilingual embedding model built on Qwen2.5-VL-3B-Instruct, supporting both dense single-vector retrieval and late-interaction multi-vector retrieval for text, images, and visually rich documents (Günther et al., 2025).

Eager-Embed-v1 (Eager). Eager-Embed-v1 is a 4B-parameter multimodal dense embedding model finetuned from Qwen3-VL-4B-Instruct for efficient visual document retrieval, using single-vector embeddings to index documents without the max-sim scoring used by ColBERT-style multi-vector retrievers (Balarini, 2025).

MoCa-3B (MoCa). MoCa-3B is a bidirectional multimodal embedding model trained from Qwen2.5-VL-3B-Instruct using modality-aware continual pre-training followed by heterogeneous contrastive fine-tuning, supporting text, image, and interleaved multimodal inputs for general multimodal retrieval and visual document retrieval (Chen et al., 2025a).

ColPali. ColPali is a PaliGemma-3B-based visual document retriever that produces ColBERT-style multi-vector representations of queries and document images, using late interaction between text tokens and image patches for fine-grained visual document retrieval (Faysse et al., 2025).

ColQwen2. ColQwen2 extends Qwen2-VL-2B-Instruct into a ColBERT-style late-interaction retriever, generating multi-vector representations for both text queries and document images and scoring them through token-level MaxSim-style matching (Faysse et al., 2025).

ColQwen2.5. ColQwen2.5 is a Qwen2.5-VL-3B-Instruct-based late-interaction visual document retriever that generates ColBERT-style multi-vector representations of text queries and document images, using dynamic image resolutions with up to 768 image patches for fine-grained document matching (Faysse et al., 2025).

F ADDITIONAL VIDORE RESULTS

Table 5: NDCG@5 on ViDoRe V1. **Bold** indicates the best score in each column and underline indicates the second-best score. Blue shading denotes the best single-vector retriever in each column. * indicates that MINER has statistically significant improvement over its corresponding backbone.

Model	ArXiv	Doc	Info	TabFQuAD	TAT	ShiftProj	Syn-AI	Syn-Energy	Syn-Gov	Syn-Health	Avg.
Multi-Vector Retrieval											
ColPali-v1.3	83.3	58.4	85.5	86.7	70.8	77.3	97.4	94.6	96.1	97.4	84.8
ColQwen2.5-v0.2	**88.9**	**63.6**	92.5	90.8	82.1	87.9	**99.6**	96.1	95.8	98.0	89.5
ColQwen2-v1.0	88.0	61.5	92.5	89.0	**82.2**	89.9	99.0	95.9	95.5	98.8	89.2
Jina(LI)	88.5	60.1	**93.8**	**95.6**	80.3	**92.9**	99.3	**97.3**	96.6	**99.1**	**90.4**
Single-Vector Baselines											
SigLIP	50.2	31.3	69.7	60.3	27.5	25.0	67.8	73.5	75.3	83.1	56.4
VLM2Vec	42.8	26.7	66.7	63.5	21.4	25.0	53.5	63.5	64.0	70.7	49.8
Jina	84.8	51.8	87.0	95.1	65.6	82.0	96.8	90.0	93.8	96.5	84.3
Eager	85.3	46.1	87.2	89.6	62.0	73.1	97.5	94.3	93.1	97.0	82.5
MoCa	86.5	55.8	89.5	93.3	65.8	82.6	97.7	94.0	96.0	95.6	85.7
Ours: MINER Single-Vector Retrieval											
MINER-Jina	84.6(-0.2%)	52.6(+1.5%)	86.7(-0.3%)	95.2(+0.1%)	64.3(-2.0%)	82.1(+0.1%)	97.1(+0.3%)	90.5(+0.6%)	93.9(+0.1%)	96.8(+0.3%)	84.4(+0.1%)
MINER-Eager	86.8(+1.8%)*	57.8(+25.4%)*	88.2(+1.1%)*	90.4(+0.9%)*	63.3(+2.1%)	77.1(+5.5%)*	97.7(+0.2%)	95.2(+1.0%)	94.3(+1.3%)	97.1(+0.1%)	84.8(+2.8%)*
MINER-MoCa	86.3(-0.2%)	55.3(-0.9%)	88.8(-0.8%)	93.0(-0.3%)	66.9(+1.7%)*	82.7(+0.1%)	98.5(+0.8%)	93.3(-0.7%)	**96.7**(+0.7%)	94.8(-0.8%)	85.6(-0.1%)

Table 6: NDCG@10 on ViDoRe V3. **Bold** indicates the best score in each column and underline indicates the second-best score. Blue shading denotes the best single-vector retriever in each column. * indicates that MINER has statistically significant improvement over its corresponding backbone.

Model	CS	Energy	Fin-EN	Fin-FR	HR	Industrial	Pharma	Physics	Avg.
Multi-Vector Retrieval									
ColPali	65.3	46.9	34.4	21.8	44.8	35.6	53.1	41.7	43.0
ColQwen2	68.6	48.6	39.0	20.0	45.1	38.3	52.2	41.6	44.2
ColQwen2.5	**72.3**	59.5	52.3	39.1	51.2	41.3	57.9	45.9	52.4
Jina(LI)	71.8	**63.5**	**59.3**	**46.1**	**59.5**	**50.4**	**63.1**	46.6	**57.5**
Dense / Single-Vector Baselines									
Jina	67.4	55.3	44.5	31.9	51.9	39.0	58.0	44.2	49.0
Eager	69.6	53.9	42.1	28.9	48.5	33.6	59.5	44.8	47.6
MoCa	68.6	54.1	49.3	33.1	47.3	38.5	54.6	46.9	49.1
Ours: MINER Single-Vector Retrieval									
MINER-Jina	68.4(+1.5%)*	56.4(+2.0%)*	44.9(+0.9%)*	32.4(+1.6%)*	52.1(+0.4%)	39.8(+2.1%)*	58.9(+1.6%)*	44.9(+1.6%)*	49.7(+1.4%)*
MINER-Eager	70.5(+1.3%)*	55.5(+3.0%)*	43.8(+4.0%)*	29.7(+2.8%)*	49.6(+2.3%)*	35.7(+6.3%)*	60.1(+1.0%)*	45.4(+1.3%)*	48.8(+2.5%)*
MINER-MoCa	70.1(+2.2%)*	54.9(+1.5%)*	49.1(-0.4%)	32.9(-0.6%)	47.7(+0.8%)	38.8(+0.8%)	55.4(+1.5%)*	**47.3**(+0.9%)*	49.5(+0.8%)*

G Query-Level Case Study

To complement the aggregate retrieval results, we conduct a query-level case study comparing Jina and MINER-Jina on ViDoRe V2 using Recall@5. We define a *recovery* as a query for which MINER-Jina retrieves the relevant document within the top 5 while the Jina dense backbone does not, and a *regression* as the converse. Table 7 summarizes the number of recoveries and regressions across the four ViDoRe V2 datasets.

Table 7: Query-level recoveries and regressions of MINER-Jina relative to the Jina dense backbone on ViDoRe V2 under Recall@5.

Dataset	MINER Regression	MINER Recovery
Biomedical	10	14
Economics	0	2
ESG-Human	1	1
ESG	0	7
Pooled	**11**	**25**

Across the four datasets, MINER-Jina produces 25 recoveries compared with 11 regressions, indicating that its aggregate improvement is accompanied by a larger number of corrected retrieval failures than newly introduced errors. The largest numbers of recoveries occur on Biomedical and ESG, with 14 and 7 recovered queries, respectively. Qualitative inspection further suggests that MINER tends to help on dense labeled diagrams, data tables, analytical prose, near-duplicate slides requiring fine-grained distinction, and queries involving multiple constraints, while regressions more often occur on sparse section-title or highlight slides. We observe both recoveries and regressions across multiple languages, without a clear language-specific pattern.

H Full Efficiency Analysis

Table 8: Per-dataset efficiency and retrieval-quality results on ViDoRe v2 using Jina. QPS and storage are measured with Qdrant (Qdrant Team, 2026) for each subset. NDCG@5 follows the main ViDoRe v2 evaluation. Higher QPS and NDCG@5 are better; lower storage is better. MINER remains single-vector retrieval with the same final embedding size and storage footprint as the dense Jina baseline. We denote the late interaction as LI.

Dataset	Method	QPS ↑	Storage (MB) ↓	NDCG@5 ↑
Bio	Jina	15.21	19.89	58.5
	Jina(LI)	2.46	859.64	60.9
	MINER-Jina	14.31	19.89	58.6
Econ	Jina	16.26	8.85	55.1
	Jina(LI)	4.80	378.74	51.9
	MINER-Jina	14.97	8.85	56.8
ESG-Human	Jina	15.16	30.11	54.8
	Jina(LI)	1.99	1267.57	65.1
	MINER-Jina	14.12	30.11	56.8
ESG-Syn	Jina	15.01	30.11	44.8
	Jina(LI)	1.75	1267.57	52.5
	MINER-Jina	14.06	30.11	48.7
Avg.	Jina	15.41	22.24	53.3
	Jina(LI)	2.75	943.38	57.6
	MINER-Jina	14.36	22.24	55.2

Table 8 provides the per-dataset breakdown behind the aggregate efficiency results in Section 5.3. Across all ViDoRe V2 subsets, MINER has the same storage footprint as the dense Jina baseline because it preserves single-vector retrieval with the same final embedding dimensionality. Its QPS is consistently slightly lower than dense retrieval, reflecting the modest overhead of computing the fused representation, but the gap remains small and stable across datasets. In contrast, late interaction incurs substantially larger and more variable storage and latency costs because it indexes multiple visual token embeddings per document, whose number can vary with document content and rendered page resolution. This leads to much lower QPS and more than $40\times$ larger index storage on average.

The retrieval-quality gains of MINER are also consistent across datasets. Compared with the dense Jina baseline, MINER improves nDCG@5 on every ViDoRe V2 subset, with gains of 0.1, 1.7, 2.0, and 3.9 on Bio, Econ, ESG-Human, and ESG-Syn, respectively. These results show that the aggregate improvement in Figure 3 is not driven by a single dataset, but instead reflects a stable quality-efficiency trade-off across the benchmark.

I Neuron Selection Analysis

Figure 10, Figure 11, and Figure 12 show the percentage of neurons retained by the Top-p masking procedure at each selected internal layer under the hyperparameter $\rho = 0.2$ and $\tau_{\mathrm{CKA}} = 0.6$. Across all three backbones, the neuron-selection profiles are highly similar, despite differences in model training, backbone design, and readout mechanism. In particular, the retained fraction is relatively low in earlier and middle layers, increases sharply in the late layers, and exhibits a consistent drop around layer 34 before rising again near the final layer. This shared structure suggests that the masking procedure is not simply fitting backbone-specific noise, but instead captures a recurring pattern in how useful cross-modal information is distributed across internal representations. Earlier and middle layers appear to contain useful but sparse features, requiring selective extraction, whereas the final layers are more directly aligned with the dense embedding interface and therefore retain substantially more dimensions. The consistent dip around layer 34 further indicates that high layer index alone is not sufficient: even among late layers, some representations contain more redundant information than neighboring layers.

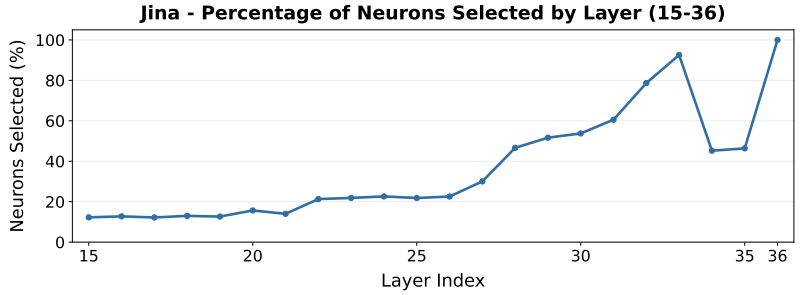

Figure 10: Percentage of neurons selected by the Top-p masking procedure for Jina across layers 15–36.

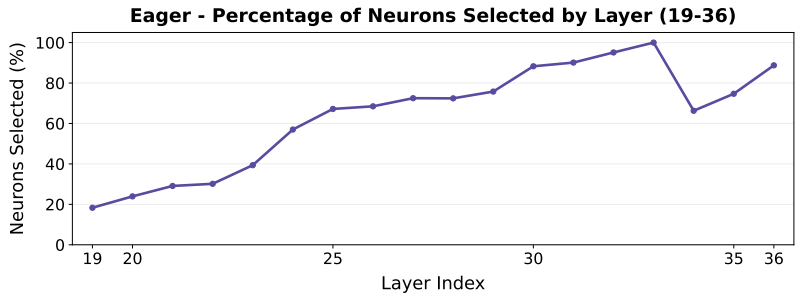

Figure 11: Percentage of neurons selected by the Top-p masking procedure for Eager across layers 19–36.

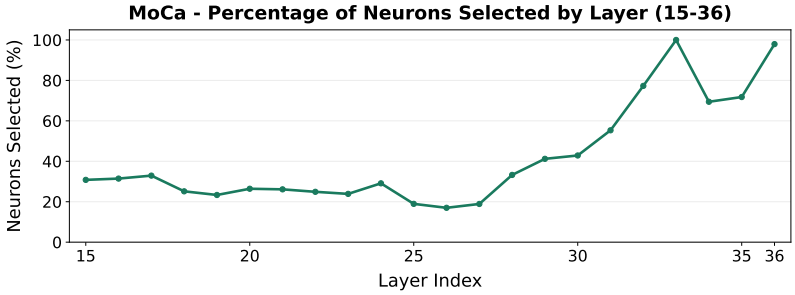

Figure 12: Percentage of neurons selected by the Top-p masking procedure for MoCa across layers 15–36.

J Additional Sensitivity Analysis

Table 9 reports the candidate layer ranges selected under different CKA cutoff values for each backbone. As the cutoff increases, the selected range becomes progressively more restrictive and shifts toward later layers, reflecting the stronger alignment of late-layer representations with the final embedding space. Despite this change in the number of candidate layers, the sensitivity plots in Figures 13 and 14 show that retrieval performance remains stable across the tested cutoffs and sparsity floors. This indicates that MINER does not rely on a fragile layer-selection cutoff, but instead remains robust across a reasonable range of CKA-based candidate sets.

Table 9: Candidate layer ranges under different CKA cutoffs.

Backbone	CKA cutoff	S_{cand}
Jina	0.50	12–36
	0.55	13–36
	0.60	15–36
	0.65	22–36
	0.70	28–36
Eager	0.50	10–36
	0.55	12–36
	0.60	19–36
	0.65	21–36
	0.70	22–36
MoCa	0.50	12–36
	0.55	13–36
	0.60	15–36
	0.65	19–36
	0.70	21–36

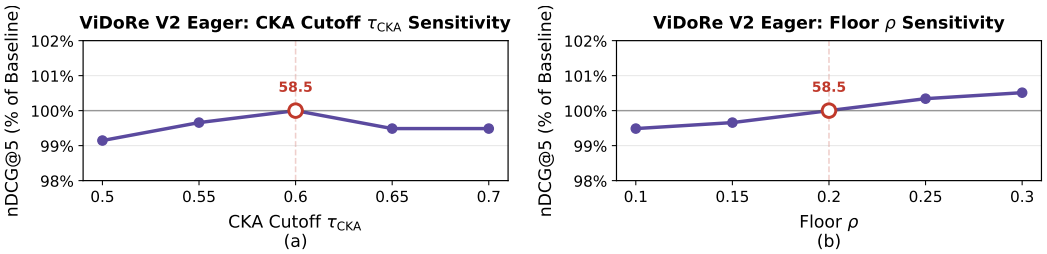

Figure 13: Hyperparameters sensitivity analysis of Eager, reported as a percentage of the default configuration

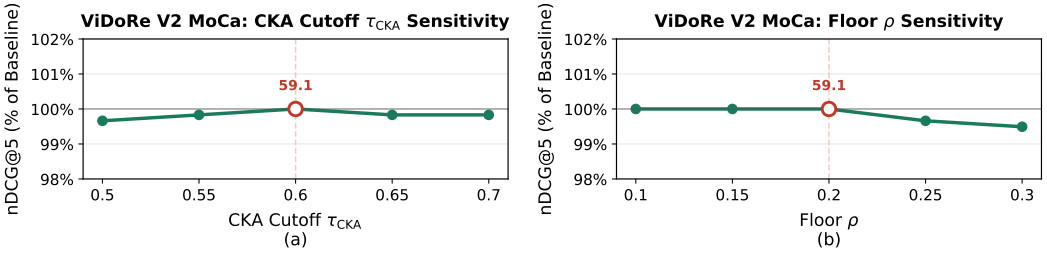

Figure 14: Hyperparameters sensitivity analysis of MoCa, reported as a percentage of the default configuration

K FUTURE WORK & LIMITATION

Future work can extend MINER to stronger multimodal embedding backbones, broader retrieval domains, and training with alternative retrieval datasets when the original backbone training data are unavailable. Importantly, MINER does not inherently require access to the original training data used to train its backbone. In our experiments, we intentionally train MINER using the corresponding backbone's original training data as a controlled evaluation choice, ensuring that performance gains arise from better utilization of internal representations rather than from additional supervision or data. In principle, the MINER module can instead be trained using other suitable paired retrieval data. Our method is currently limited to open-source backbones, as it requires access to the model's internal hidden states. For the same controlled-evaluation reason, we also do not report results on open-weight models whose original training data are not disclosed, such as Qwen3-VL-Embedding-8B (Li et al., 2026). This is an experimental constraint of our study rather than a requirement of MINER itself.