miniReranker: Efficient Multimodal Reranking through Visual Cache Reuse and Interaction Sparsity

Yingqi Fan[1], Xuan Lu[1], Anhao Zhao[1], Junlong Tong[1], Ping Nie[2], Kai Zou[2], Yunpu Ma[3], Wei Zhang [1], Xiaoyu Shen[1]*

[1]Eastern Institute of Technology, Ningbo, [2]University of Waterloo, [3]Netmind.ai, [4]Munich Center for Machine Learning, LMU

Correspondence: yingqi949@gmail.com xyshen@eitech.edu.cn

Abstract

Multimodal large language models (MLLMs) have emerged as powerful rerankers for multimodal retrieval, but their fine-grained token-level interactions come at a substantial computational cost. In point-wise reranking, each query must be independently paired with many candidate documents, resulting in repeated computation over highly overlapping multimodal inputs. In this work, we propose *miniReranker*, an efficient MLLM reranking framework that reduces redundancy at multiple levels. First, we introduce a *vision-first* reformulation that aligns with native MLLM prompting formats while maximizing reuse of expensive visual representations through prefix caching. We then identify redundancy in model depth, cross-segment interactions, and visual token representations, motivating three complementary optimizations: *early exit*, *interaction bands*, and *embedder-guided token pruning*. Built on Qwen3-VL, *miniReranker* achieves similar performance while reducing active parameters to 58% and achieving nearly 3× training acceleration. When reranking top-100 candidates, it reduces reranking runtime by up to >99%, demonstrating that efficient MLLM reranking can be achieved without sacrificing quality.

1 Introduction

Multimodal large language models (MLLMs) have recently emerged as powerful rerankers for multimodal retrieval systems (Abootorabi et al., 2025; Huang et al., 2026). Unlike dual-encoder retrievers that independently map queries and documents into global embeddings (Zhang et al., 2025b; Lu et al., 2025), MLLM rerankers jointly process query-document pairs through token-level interactions, enabling fine-grained cross-modal reasoning and substantially stronger relevance modeling (Dai et al., 2025; Li et al., 2026b; Lu et al., 2026c,b). However, this stronger interaction capability comes at a significant computational cost. In modern retrieval systems, each query must be compared against dozens or hundreds of candidate documents, forcing the reranker to repeatedly process highly overlapping multimodal inputs (Chen et al., 2025b).

Existing point-wise MLLM rerankers predominantly inherit the query-first formulation from traditional text retrieval systems, arranging inputs as $[\text{ins}, q, d]$ regardless of modality composition (Bai et al., 2025; Chen et al., 2025c; Xu et al., 2025; Liu et al., 2025; Lin et al., 2025). While natural for purely textual retrieval, this design is suboptimal for multimodal reranking from both effectiveness and efficiency perspectives. From an effectiveness standpoint, modern MLLMs are primarily pretrained on visual-question-answering (VQA) style data, where visual content naturally precedes textual instructions and questions (Liu et al., 2024a; Chen et al., 2024c). Query-first prompting therefore frequently breaks the model's native input format, particularly in text-to-visual retrieval settings. From an efficiency standpoint, sequence order determines which representations can be reused through prefix KV caching (Qin et al., 2026; Lu et al., 2026a). Because visual inputs typically dominate the computational cost (Lin et al., 2024), a fixed query-first formulation often forces expensive visual documents to be repeatedly encoded across candidate pairs. Simply adopting a document-first formulation is not a universal solution, as it becomes suboptimal when the query itself contains visual content. Motivated by these observations, we introduce a simple *vision-first* reformulation that consistently places visual inputs before textual inputs. This strategy simultaneously restores alignment with MLLM pretraining formats and maximizes reuse of expensive visual computation, yielding improvements in both reranking effectiveness and efficiency.

*Corresponding Author

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 39488–39505
October 24-29, 2026 ©2026 Association for Computational Linguistics

Although vision-first prompting eliminates a substantial fraction of redundant computation, the remaining cost is still considerable. Beyond sequence ordering, dense MLLM rerankers perform extensive computation throughout the entire network depth, repeatedly execute cross-segment attention across all layers, and process large numbers of visual tokens whose contributions to the final relevance decision may be marginal (Fan et al., 2026). To better understand these inefficiencies, we analyze reranking behavior through layer-wise logit probing (nostalgebraist, 2020). Our analysis reveals that reranking signals emerge much earlier than final-layer predictions, that effective query-document interactions are concentrated within a narrow subset of layers, and that many visual tokens contribute little to the final relevance score.

Building on these observations, we propose *miniReranker*, an efficient MLLM reranking framework that combines vision-first prompting with three complementary compression strategies. First, we employ *early exit* to truncate unnecessary upper transformer layers after relevance signals have largely converged. Second, we introduce an *interaction band* that restricts expensive query-document attention to the layers where meaningful cross-segment information exchange actually occurs. Third, we perform *embedder-guided token pruning*, leveraging attention information already produced by the retrieval-stage encoder to remove redundant visual tokens without requiring additional forward passes.

We instantiate *miniReranker* on top of Qwen3-VL-Instruct (Bai et al., 2025) and fine-tune it using a point-wise `yes`/`no` relevance objective on a newly constructed multimodal reranking dataset. We evaluate *miniReranker* on MMEB-v2 (Meng et al., 2025), covering image, visual-document, and video tasks. Results show that *miniReranker* outperforms the original instruct models and remains competitive with existing multimodal embedding/reranking baselines, while preserving $\sim 96\%$ of the dense reranker performance.

In terms of efficiency, *miniReranker* reduces active parameters to about 58% of the dense model, achieves nearly $3\times$ training acceleration, and substantially lowers online reranking latency. In particular, when reranking Top-100 candidates with a single query, *miniReranker* reduces video reranking runtime to $< 1\%$ of the dense implementation and image reranking runtime to $< 15\%$. Such savings become even more significant as the number of queries increases. Our main contributions are summarized as follows:

- We propose a *vision-first* prompt reformulation that enables reusable visual pre-caching while improving reranking effectiveness.
- We reveal that reranking computation in MLLMs contains substantial depth-, interaction-, and token-level redundancy.
- We conduct extensive experiments on MMEB-v2 across 78 tasks, showing that `miniReranker` preserves most of the dense reranking performance while reducing reranking runtime by over 99% in high-reuse settings.

2 Preliminaries

Modern multimodal search systems typically follow a two-stage "retrieve-and-rerank" pipeline (Nogueira and Cho, 2020; Nogueira et al., 2019). Given a query q, a fast, coarse-grained retriever first extracts a manageable candidate set of documents $\mathcal{D} = \{d_1, \ldots, d_K\}$ from a large corpus. A multimodal reranker is then tasked with precisely rescoring and reordering these candidates. Because both the query and the documents can be represented in either textual or visual (image/video) formats, this formulation accommodates diverse paradigms, including text-to-visual (T→V), visual-to-text (V→T), and visual-to-visual (V→V) (Wei et al., 2023).

2.1 Multimodal Point-wise Reranking

While list-wise approaches evaluate multiple candidates simultaneously, they require concatenating numerous documents into a single, exceptionally long sequence (Qin et al., 2024; Cao et al., 2007; Xia et al., 2008). Because modern MLLMs often suffer from severe performance degradation and attention dilution when processing long lists of interleaved visual inputs (Liu et al., 2024b; Xiao et al., 2024), the point-wise formulation has emerged as the prevailing paradigm.

MLLM Architecture. To process these heterogeneous modalities, MLLMs map textual and visual inputs into a unified token space (Wu et al., 2026b). Text is tokenized directly, whereas visual inputs are encoded by a vision model (e.g., a Vision Transformer (Dosovitskiy et al., 2021)) and subsequently projected into the language model's embedding space (Liu et al., 2024a). The pointwise MLLM reranker, parameterized by L transformer layers and a hidden dimension d_h, then operates

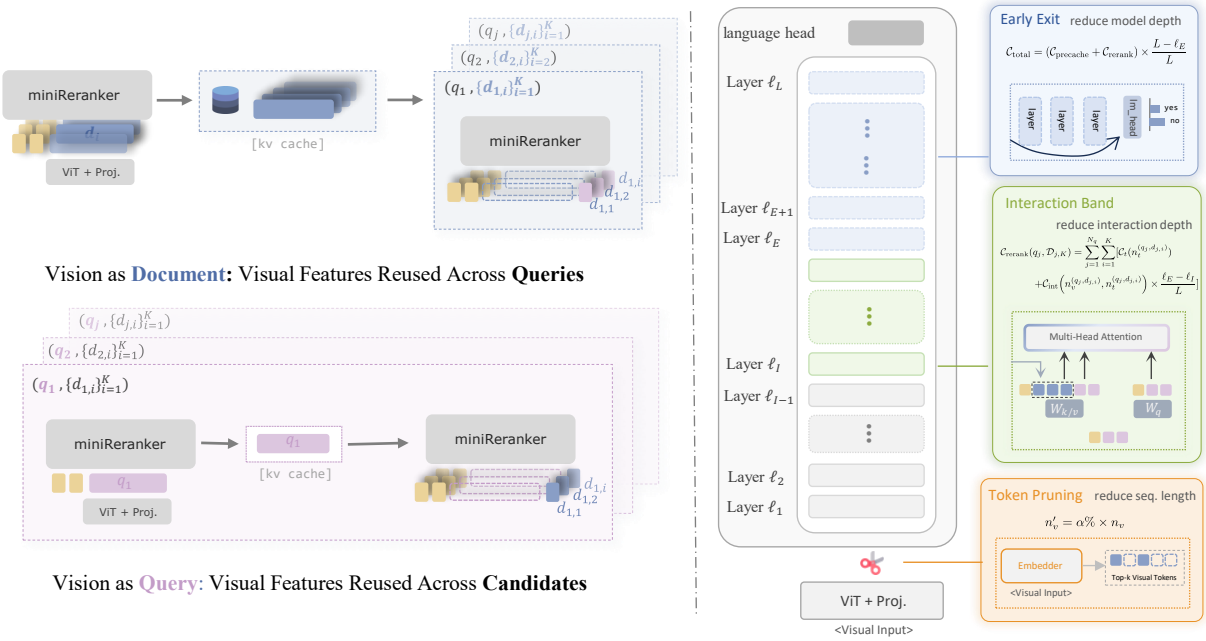

Figure 1: Overview of **miniReranker**. Left: the proposed *Vision-first* reformulation enables reusable visual pre-caching for both vision-as-document and vision-as-query settings. Right: `miniReranker` further improves efficiency through three complementary compression strategies: (1) *Early Exit*, which reduces depth-wise computation by terminating inference at intermediate layers; (2) *Interaction Band*, which restricts cross-segment attention to a narrow range of effective layers; and (3) *Token Pruning*, which removes redundant visual tokens to reduce sequence length and interaction cost.

over a concatenated sequence comprising system instruction tokens, n_q query tokens, and n_d document tokens.

Point-wise Scoring. A point-wise MLLM reranker evaluates each query–document pair entirely independently. Guided by a predefined task instruction ins, the model estimates the relevance of a document d to the query q by generating a binary yes/no decision at the next-token prediction step, modeling the conditional probability $p(\cdot \mid \mathrm{ins}, q, d)$.

Let z_{yes} and z_{no} denote the output logits corresponding to the yes and no tokens. The final relevance score $s(q, d)$ used to rank the candidates is computed as the normalized probability of the positive class (Li et al., 2026b; Dai et al., 2025):

$$s(q, d) = \sigma(z_{\text{yes}} - z_{\text{no}}), \tag{1}$$

where $\sigma(\cdot)$ is the sigmoid function.

2.2 Complexity Analysis

The sequence order of the query and document dictates how much of this computation must be performed online. Because the task instruction ins is fixed across queries, its representations can always be pre-cached offline; thus, we omit it from the online complexity analysis.

Query-First Configuration. MLLM rerankers can adopt a query-first sequence order, formatted as $[\mathrm{ins}, q, d]$. Because the query q is identical for all K candidates for a given user request, its key-value representations and FFN outputs can be computed once and broadcasted. However, because q precedes d, the document tokens d are dynamic relative to the prefix and must be encoded online K times. For each candidate, the n_d document tokens attend to the query prefix and themselves, and pass through the FFN. The total online complexity to rerank K documents is therefore:

$$\mathcal{C}_{\text{q-first}} = \mathcal{O}\Big(L\big[n_q^2 d_h + n_q d_h^2\big] + KL\big[n_d(n_q + n_d)d_h + n_d d_h^2\big]\Big).$$

The first line represents the one-time query encoding, while the second line captures the repeated self/cross-attention and FFN operations of the document tokens.

Document-First Configuration. Alternatively, a document-first configuration orders the input as $[\mathrm{ins}, d, q]$. Because the candidate documents are often fixed in a corpus, the representations of the document tokens d (both their self-attention keys/values and their FFN transformations) can be pre-computed and cached offline. During online infer-

ence, for each of the K candidates, the MLLM only needs to encode the n_q query tokens. These query tokens perform attention against the cached document representations and pass through the FFN to generate the yes/no logits. This reduces the online computational complexity to:

$$\mathcal{C}_{\text{d-first}} = \mathcal{O}\Big(KL\big[n_q(n_d + n_q)d_h + n_q d_h^2\big]\Big).$$

3 Methods

We propose an efficient framework that optimizes MLLM reranking through two modifications. First, a *vision-first* reformulation (Sec. 3.1) aligns the input with native MLLM pre-training formats while maximizing the reuse of heavy visual computations. Second, to reduce redundancy in cross-modal information flows, we introduce three inference-time optimizations: early exiting, reduced interaction bands, and token pruning (Sec. 3.2).

3.1 Vision-First Prompt Reformulation

Limitations of Query-First Rerankers. Existing point-wise MLLM rerankers predominantly inherit the *query-first* configuration from traditional text-based retrieval (Bai et al., 2025; Chen et al., 2025c; Xu et al., 2025; Liu et al., 2025; Lin et al., 2025; Wang et al., 2024; Tong et al., 2026a), strictly formatting the input sequence as $[\text{ins}, q, d]$. However, blindly applying this rigid, text-centric design to multimodal models introduces two major bottlenecks. First, from an accuracy perspective, modern MLLMs are typically pre-trained on Visual Question Answering (VQA) formats (Liu et al., 2024a; Dai et al., 2023; Alayrac et al., 2022; Chen et al., 2024c), where the model is optimized to process visual tokens *before* textual ones. In the common T→V setting (text query, visual document), a query-first setup forces the MLLM to process the text query before the visual document, directly breaking its native pre-training alignment. Second, from an efficiency perspective, sequence order dictates which segment must be recomputed K times online (Sec. 2.2). Because visual tokens often dominate the computational cost, the query-first setup forces heavy visual documents in the T→V setting to be repeatedly re-encoded online for every candidate. Conversely, rigidly adopting a document-first setup to fix this would create a massive bottleneck in the V→T setting, where the heavy visual query would then have to be repeatedly encoded.

Dynamic Sequence Ordering. To resolve both the accuracy and efficiency bottlenecks simultaneously, we advocate for a dynamic configuration that strictly places the *visual modality ahead of the textual modality*. Concretely, for the T→V setting, we adopt a *document-first* order $[\text{ins}, d, q]$, which restores the natural VQA prompting style for higher accuracy and allows the heavy visual documents to be cached offline. For the V→T setting, we adopt a *query-first* order $[\text{ins}, q, d]$, which maintains the VQA alignment (vision precedes text) and ensures the massive visual query is encoded only once per user request and broadcasted. In the V→V setting where both inputs are visual, we place the document first. We empirically find it performs similarly to query-first in terms of accuracy, but offers the critical system advantage of enabling offline pre-caching for the static document corpus. As shown in Tab. 1, adhering to this format consistently yields superior performance both in zero-shot settings (Instruct) and after supervised fine-tuning (reranker†)[1].

Model	2B		4B	8B
	Instruct	reranker†	Instruct	Instruct
Q-first	61.5	71.3	67.5	69.7
D-first	52.7	69.6	61.9	64.2
V-first	**63.6**	**73.0**	**68.8**	**70.8**

Table 1: Prompt format comparison.

Theoretical FLOPs Reduction. We quantify the exact computational savings ($\Delta\mathcal{C}$) of our dynamic formulation by evaluating the difference between the optimal and sub-optimal sequence orders. By expanding the equations from Sec. 2.2, the cross-attention interaction cost ($\mathcal{O}(KLn_q n_d d_h)$) perfectly cancels out during subtraction, beautifully isolating the pure FLOPs saved in self-attention and FFN encoding. In the **T→V setting** ($n_q \ll n_d$), comparing our vision-first (document-first) approach against the standard query-first baseline yields an exact online reduction of $\Delta\mathcal{C}_{\text{T}\to\text{V}} = \mathcal{C}_{\text{q-first}} - \mathcal{C}_{\text{d-first}} = \mathcal{O}\big(Ld_h[Kn_d(n_d+d_h) - (K-1)n_q(n_q+d_h)]\big)$. Because the visual document is massive ($n_d \gg n_q$), the positive term overwhelmingly dominates, demonstrating the elimination of K repeated online encodings of the heavy visual document. Conversely, in the **V→T setting** ($n_q \gg n_d$), our vision-first formulation equals a query-first setup. Compared to a rigid document-

[1] † denotes models fine-tuned using our datasets; experimental details are provided in Sec. 4.

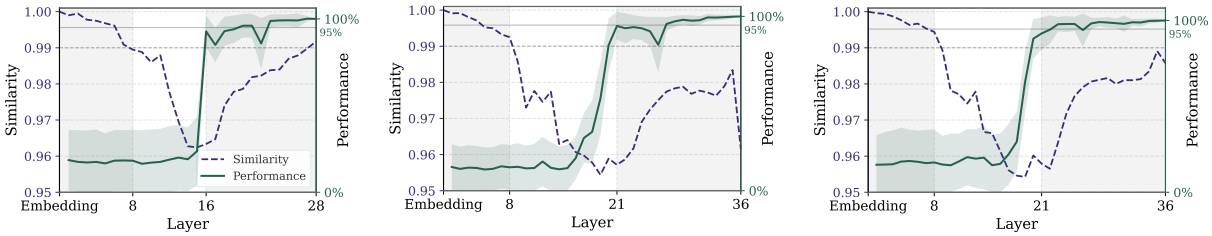

Figure 2: **Layer-wise Logit Probing** reveals substantial depth-wise redundancy in multimodal reranking, while **Cross-segment Interaction Analysis** shows that effective cross-segment information exchange is concentrated within a narrow range of intermediate layers.

first strategy, the reduction in online FLOPs is $\Delta\mathcal{C}_{\mathrm{V}\to\mathrm{T}} = \mathcal{C}_{\mathrm{d-first}} - \mathcal{C}_{\mathrm{q-first}} = \mathcal{O}\big(Ld_h[(K-1)n_q(n_q+d_h) - Kn_d(n_d+d_h)]\big)$. By broadcasting the visual query, we successfully avoid $K-1$ redundant online encodings of the massive visual tokens. Finally, in the **V→V setting** where both modalities are heavy, our vision-first (document-first) approach exploits global corpus redundancy. Across a system serving N_q queries over a static corpus of N_d documents, the query-first baseline re-encodes visual documents N_qK times. Pre-caching these documents reduces the system-level document encoding FLOPs by exactly $\Delta\mathcal{C}_{\mathrm{V}\to\mathrm{V}} = \mathcal{O}\big((N_qK-N_d)Ln_d(n_d+d_h)d_h\big)$, yielding immense computational relief at scale.

3.2 Model-Level Compression

While the vision-first prompt reformulation optimizes cross-modal sequence ordering, MLLMs still suffer from massive internal computational overhead during the forward pass. To systematically eliminate this redundancy, we analyze the reranking process from a model-level perspective and introduce three orthogonal compression strategies targeting model depth, cross-segment attention density, and sequence length.

Early Exit: Truncating Model Depth. Unlike open-ended visual generation, which depends on deep iterative reasoning to autoregressively decode precise output tokens, point-wise reranking only requires the model to estimate a relevance score from the prefill representations. Prior studies have shown that the deepest layers of large language models are mainly responsible for linguistic refinement and final token disambiguation, transforming internal representations into fluent natural language outputs (Chuang et al., 2024; Fan et al., 2025; Lad et al., 2025; Skean et al., 2025; Liu et al., 2026; Zhao et al., 2025; Tong et al., 2026b; Zhao et al., 2026). Since reranking does not require autoregressive text generation, these late-stage decoding capabilities might be largely unnecessary for the task.

To validate this assumption, we conduct a layer-wise probing analysis using the logit lens technique (nostalgebraist, 2020). At each transformer layer l, we extract the hidden state $h^{(l)}$ at the final token position, apply the final layer normalization LN, and project it to the vocabulary space $\mathcal{X}$ using the model's output head $\phi(\cdot)$:

$$z_x^{(l)} = \phi(\mathrm{LN}(h^{(l)}))_x, \quad x \in \mathcal{X}. \tag{2}$$

We then compute the relevance score using the yes/no logits via Eq. 1. As shown in Fig. 2, reranking accuracy rises sharply in the intermediate layers and saturates early, achieving ∼95% of the final-layer performance using only <60% of the network depth. [2] Motivated by this, we employ an early-exit strategy that safely terminates the forward pass at a predefined intermediate layer, significantly reducing depth-wise computation without meaningful accuracy loss.

Interaction Band: Localizing Query-Document Attention. Fig. 2 reveals that scoring capabilities emerge abruptly within a narrow subset of layers, implying that effective query-document interaction might also be highly localized. To reduce computational redundancy, we first investigate how relevance signals from the first segment (query or document, depending on sequence order) reach the final decision token. By selectively masking attention, we compare a *direct pathway* (final token attends directly to the first segment) against an *indirect pathway* (information fuses into the second segment, which then informs the final token). Blocking the direct pathway barely affects accuracy

[2] See App. B.1 for comparisons with general VQA tasks.

(61.5→61.0), whereas blocking the indirect pathway (disabling query-document attention) causes a catastrophic collapse (61.5→5.5). This confirms that relevance information must be fused into the second segment before reaching the final token.

To identify *where* this critical fusion occurs, we fix the second segment, vary the first, and measure the cosine similarity of the second segment's internal representations across layers (Fig. 2). In shallow layers, similarity remains near 1.0, indicating independent evolution. As depth progresses, similarity drops sharply, marking the exact onset of active cross-modal integration. Crucially, this divergence perfectly aligns with the accuracy spike observed in our early-exit analysis. We define this concentrated region of effective query-document attention as the *interaction band*.

To exploit this localized integration, we design a selective interaction mechanism. During training, a sparse attention mask forces segments to evolve independently in non-interaction layers by blocking query-document attention, retaining standard causal attention solely within the interaction band. Consequently, during inference, we only load the first segment's key-value (KV) cache for this specific subset of layers. This preserves essential cross-segment information exchange while safely eliminating expensive quadratic attention operations elsewhere. Empirically, we set this interaction band to layers 8–16 for Qwen3-VL-2B, and layers 8–21 for Qwen3-VL-4B/8B.

Token Pruning: Condensing Visual Sequences. While early exiting and the interaction band effectively reduce computational depth and cross-attention density, the raw sequence length—typically dominated by massive numbers of visual patches—remains a fundamental bottleneck. To directly compress the visual sequence before it enters the MLLM backbone, we introduce an embedder-guided token pruning strategy.

Rather than introducing expensive operations to dynamically evaluate token importance during the reranking pass, we obtain these importance scores entirely for free. Because the visual inputs have already been processed by an upstream first-stage retriever (e.g., `Qwen3-VL-Embedding`) during the initial recall phase, we simply repurpose its readily available computation. Specifically, we extract the attention weights directed from the embedder's final sequence token to the visual tokens. However, because attention patterns fluctuate drastically across different network depths (Kang et al., 2025; Dhouib et al., 2025; Zhao et al., 2024; Jiang et al., 2025a; Wu et al., 2026a), relying on a single layer (e.g., the 3rd layer (Chen et al., 2024a)) yields unstable token selection. To ensure robust semantic representation, we aggregate the attention scores across *all* layers of the first-stage embedder. [3]

Selection	`Image`	`VisDoc`	`Video`	overall
Dense	59.1	74.4	49.4	61.5
3rd Layer	53.2	70.2	47.9	57.2
14th Layer	57.3	71.6	49.0	59.7
Last Layer	56.8	72.4	**49.5**	59.9
All Layers	**58.2**	**73.0**	49.2	**60.7**

Table 2: Comparison of visual token selection strategies under the same pruning ratio.

Based on these aggregated scores, we select the top $\alpha\%$ visual tokens and discard the remainder before passing the sequence to the reranker. As shown in Tab. 2, compared to single-layer heuristics, this layer-aggregated approach accurately preserves the most salient semantic matching features. It integrates seamlessly into the standard retrieval pipeline with zero computational overhead, allowing us to achieve an aggressive 50% visual token compression rate while successfully recovering near-dense baseline accuracy.

4 Experiments

4.1 Experimental Setup

Training. We construct a point-wise reranking training dataset covering three major multimodal categories: (i) image, (ii) visual document, and (iii) video, resulting in a total of approximately 736K training pairs. We fine-tune only the LLM component of `Qwen3-VL-Instruct` for one epoch with a supervised point-wise reranking objective, using LoRA adaptation and a learning rate of 1×10^{-4}; more details are provided in App. C.

Compression Setting. We configure `miniReranker` according to the empirical analysis in the previous section. Specifically, the early-exit layer is set to Layer 16 for the 2B model and Layer 21 for the 4B/8B models. Across all model sizes, cross-segment interaction is enabled from Layer 8 onward, and only half of visual tokens are retained after pruning.

Evaluation. We evaluate on MMEB-v2, covering 36 image tasks, 24 visual document tasks, and

[3] See App. E for additional token selection baselines.

Model	Image					Video					VisDoc					All
	CLS	QA	RET	GD	Overall	CLS	QA	RET	MRET	Overall	VDRv1	VDRv2	VR	OOD	Overall	
ColPali v1.3 (3B) (Faysse et al., 2024)	40.3	11.5	48.1	40.3	34.9	26.7	37.8	21.6	25.5	28.2	83.6	52.0	81.1	43.1	71.0	44.4
GME (2B) (Zhang et al., 2025b)	54.4	29.9	66.9	55.5	51.9	34.9	42.0	25.6	32.4	33.9	86.1	54.0	82.5	43.1	72.7	54.1
VLM2Vec (2B) (Meng et al., 2025)	58.7	49.3	65.0	72.9	59.7	33.4	30.5	20.6	33.0	29.0	49.8	13.5	51.8	33.5	41.6	47.0
Qwen3-VL-Reranker-2B (Li et al., 2026b)	—	—	—	—	74.0	—	—	—	—	53.2	—	—	—	—	83.2	75.2
Qwen3-VL-2B-Instruct (Bai et al., 2025)	59.2	61.2	60.1	74.2	61.7	60.2	55.4	20.9	36.7	44.0	87.2	71.1	83.8	72.4	81.2	63.6
Qwen3-VL-2B-reranker†	67.1	71.8	71.8	82.2	71.6	63.8	60.3	57.8	49.5	58.8	93.6	66.7	92.1	75.5	85.7	73.0
Qwen3-VL-2B-miniReranker†	65.6	65.7	69.7	76.5	68.2	62.5	50.9	57.6	45.1	55.0	93.3	64.3	92.1	73.8	84.9	70.3(96.3%)
Qwen3-VL-4B-reranker†	69.7	76.1	75.2	88.7	75.4	63.2	63.6	60.6	48.3	60.1	94.3	66.8	93.9	75.9	86.5	75.3
Qwen3-VL-4B-miniReranker†	68.3	71.5	73.5	81.8	72.5	64.5	64.0	59.2	52.7	60.9	93.8	68.1	92.7	75.3	86.2	74.0(98.3%)
GME (7B)	57.7	34.7	71.2	59.3	56.0	37.4	50.4	28.4	38.2	38.6	89.4	55.6	85.0	44.4	75.2	57.8
LamRA-Qwen2.5 (7B)	51.7	34.1	66.9	56.7	52.4	32.9	42.6	23.2	37.6	33.7	56.3	33.3	58.2	40.1	50.2	47.4
VLM2Vec-Qwen2VL (7B)	62.7	56.9	69.4	82.2	65.5	39.1	30.0	29.0	40.6	34.0	56.9	9.4	59.1	38.1	46.4	52.3
Qwen3-VL-Reranker-8B	—	—	—	—	78.2	—	—	—	—	61.0	—	—	—	—	85.8	79.2
Qwen3-VL-8B-reranker†	70.0	77.8	75.2	90.8	76.2	65.6	67.1	61.7	53.3	62.9	94.8	68.2	94.0	76.4	87.1	76.5
Qwen3-VL-8B-miniReranker†	69.2	73.9	73.7	89.7	74.3	67.2	67.3	60.9	48.0	62.5	94.1	68.6	93.4	76.3	86.7	75.4(98.6%)

Table 3: **Main results** on MMEB-v2 multimodal reranking benchmarks across image, video, and visual-document tasks. CLS: classification, QA: question answering. RET: retrieval, GD: grounding, MRET: moment retrieval, VDR: ViDoRe, VR: VisRAG, OOD: out-of-domain. † denotes models finetuned using our training recipe.

Model	Image					Video					VisDoc					All
	CLS	QA	RET	GD	Overall	CLS	QA	RET	MRET	Overall	VDRv1	VDRv2	VR	OOD	Overall	
Prompting Format																
Qwen3-VL-2B-reranker^{VF}	67.1	71.8	71.8	82.2	**71.6**	63.8	60.3	57.8	49.5	**58.8**	93.6	66.7	92.1	75.5	**85.7**	**73.0**
Qwen3-VL-2B-reranker^{QF}	67.1	70.9	70.5	84.1	71.2	63.1	58.4	47.7	39.4	53.6	91.7	67.5	91.5	73.6	84.6	71.3$_{\downarrow 1.7}$
Qwen3-VL-2B-reranker^{DF}	66.8	62.2	71.5	84.9	69.1	59.1	31.8	57.0	50.9	49.5	93.4	65.4	92.3	75.5	85.5	69.6$_{\downarrow 3.4}$
Compression Components																
Qwen3-VL-2B-reranker^{EE}	67.0	69.7	72.0	83.1	71.3	61.8	56.7	56.6	47.9	56.6	93.6	67.1	92.3	75.3	85.8	72.3$_{(99.0\%)}$
Qwen3-VL-2B-reranker^{TP}	66.9	71.6	71.3	79.3	71.0	64.7	59.6	57.5	50.4	58.9	93.5	66.1	92.1	74.5	85.4	72.7$_{(99.6\%)}$
Qwen3-VL-2B-reranker^{IB}	67.2	69.9	71.6	77.3	70.5	62.1	53.9	59.0	47.4	56.5	93.7	63.9	92.4	74.5	85.2	71.8$_{(98.4\%)}$

Table 4: **Ablation study** of prompting formulations and individual compression components after supervised fine-tuning.

18 video tasks (Meng et al., 2025). Reranking is performed over 1 positive sample and 19 hard negatives retrieved by `Qwen3-VL-Embedding-2B`. For image and video tasks, we report Hit@1, while for visual document tasks, we report NDCG@5. More details are provided in App. D.

4.2 Main Results

Validating fine-tuning effectiveness. As shown in Tab. 3, our finetuned point-wise rerankers consistently outperform their corresponding instruct models and achieve strong overall performance compared with existing multimodal embedding and reranking baselines at similar scales, establishing a strong dense reranking baseline for subsequent compression experiments.

`miniReranker`. Compared with the corresponding dense reranker trained under the same setup, `miniReranker` preserves $> 96\%$ of the reranking performance for the 2B model and $> 98\%$ for the 4B/8B models, while substantially reducing computation. The proposed compression strategy shows consistent effectiveness across model scales.

4.3 Ablation Study

Tab. 4 reports ablations of prompting formulations and compression components after training.

Prompting Formulation. Among query-first (QF), document-first (DF), and the proposed vision-first (VF) prompting, VF achieves the best overall reranking performance, confirming that the vision-first formulation remains effective even after reranking finetuning.

Compression Components. We further evaluate each compression strategy independently based on the vision-first formulation. Early exit (EE), interaction band restriction (IB), and embedder-guided token pruning (TP) all preserve competitive reranking performance, showing that each component is individually effective for improving efficiency.

5 Efficiency Analysis

We analyze the efficiency of 2B-miniReranker from three perspectives: parameter usage, training efficiency, and reranking latency.

Parameter Usage. Our early exit strategy directly reduces the number of active parameters during both training and inference. For the 2B

model, only the first 16 out of 28 layers are used, corresponding to 57.1% of the original parameters (1.14B active parameters). For the 4B and 8B models, only 21 out of 36 layers are retained, resulting in 58.3% parameter usage (2.2B active parameters for the 4B model and 4.6B for the 8B model).

Training Hours. Our compression framework mitigates multimodal reranker training bottlenecks via: (1) *early exit*, reduce the number of updated parameters; and (2) *visual token pruning*, shorten the long multimodal sequences. These optimizations jointly reduce both forward and backward computation costs, *miniReranker* achieves nearly 3× training acceleration compared with the dense reranker.

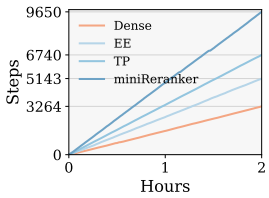

Figure 3: Training throughput comparison.

Reranking Latency: Vision as Query. For the vision-as-query setting, we evaluate on image task MS COCO_{i2t} and the video task UCF101. We fix the number of queries to 1 and gradually increase Top-k from 10 to 100. As shown in Fig. 4, miniReranker consistently achieves substantially lower latency than the vanilla implementation and the dense visual-reuse baseline. For MS COCO_{i2t}, reranking Top-100 candidates requires only around 15% of the original latency. For UCF101, which involves significantly longer video sequences, the latency is further reduced to $< 1\%$ of the original runtime. Moreover, compared with the dense visual-reuse baseline, the compressed `miniReranker` further reduces latency by ∼66%.

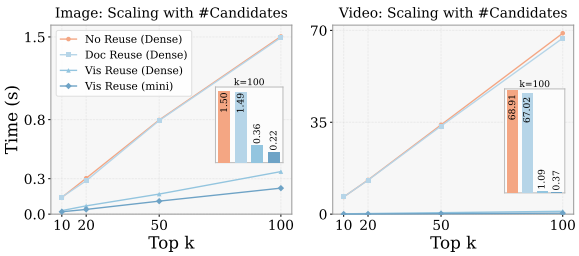

Figure 4: Latency scaling in the *vision-as-query* setting.

Reranking Latency: Vision as Document. For the vision-as-document setting, we evaluate on MS COCO_{t2i} and the video retrieval benchmark MSR-VTT. As shown in Fig. 5, the two upper subfigures measure latency under increasing Top-k with a single query, demonstrating that vision-first reuse generalizes effectively across both vision-to-text and text-to-vision reranking settings. The lower subfigures evaluate latency under increasing numbers of queries with fixed Top-100 reranking, showing that the latency advantage of *miniReranker* further amplifies as the reranking workload scales, reaching $< 1\%$ of the original runtime on the long-video benchmark MSR-VTT.

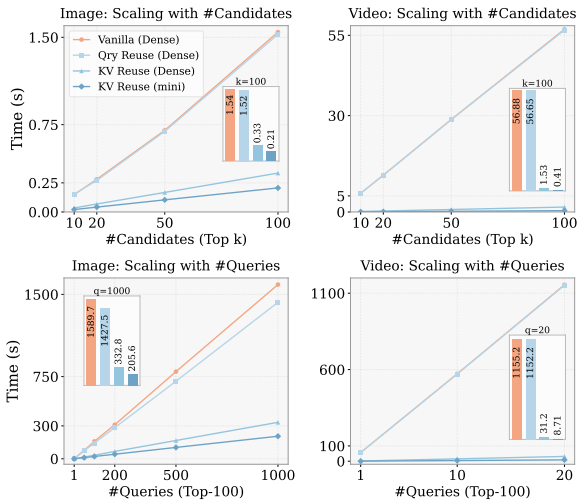

Figure 5: Latency scaling in the *vision-as-document* setting.

Reranking Latency: Ablation. We further analyze the contribution of each compression component to reranking acceleration. We scale the number of candidates and report the latency averaged over the two reuse scenarios and tasks. As shown in Fig. 6, early exit provides the largest gain on image tasks, where the sequence length is relatively short and the cost is mainly dominated by model depth. In contrast, for video tasks with much longer visual sequences, visual token pruning and interaction-band restriction become more important, as they directly reduce the sequence length and cross-segment attention cost. Combining all components yields the lowest latency across both modalities, indicating that the proposed techniques address complementary sources of computation.

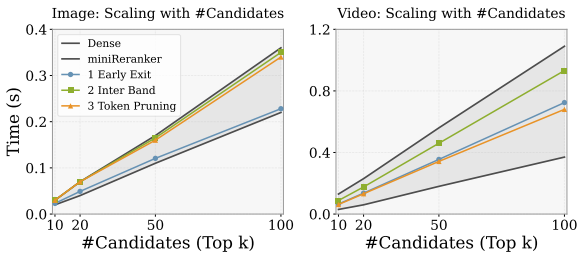

Figure 6: Latency breakdown of compression components.

Overall, *miniReranker* consistently improves efficiency across training and inference. It reduces

the active parameters to about 58% of the dense model, achieves nearly $3\times$ training speedup, and substantially lowers reranking latency under both vision-as-query and vision-as-document settings. The reduction is especially pronounced for video reranking; in our measured settings, *miniReranker* reduces the runtime to less than 1% of the dense baseline. These results show that vision-first reuse, early exit, interaction-band restriction, and visual token pruning jointly provide an effective compression framework for efficient multimodal reranking.

6 Conclusions

In this work, we present `miniReranker`, an efficient MLLM-based point-wise reranking framework. By combining vision-first prompting with model-level compression, our method substantially reduces reranking computation while preserving most dense-model performance. Experiments across image, video, and visual-document tasks demonstrate strong efficiency–effectiveness tradeoffs for scalable multimodal reranking.

Limitations

Our work primarily focuses on point-wise multimodal reranking, where relevance is independently estimated for each query–document pair via prefill-only yes/no scoring. While this formulation is widely adopted in existing multimodal reranking systems, it does not capture richer interactions across candidates that may arise in list-wise reranking or agentic retrieval pipelines. Extending the proposed compression strategies to such settings remains an important direction for future work.

Ethical Considerations

This work focuses on improving the efficiency of multimodal reranking systems through model compression and computation reuse, which may help reduce computational cost and energy consumption for large-scale deployment. Our method does not involve the collection of new user data or introduce additional ethical concerns beyond those already associated with the underlying multimodal foundation models and retrieval systems.

References

Mohammad Mahdi Abootorabi, Amirhosein Zobeiri, Mahdi Dehghani, Mohammadali Mohammadkhani, Bardia Mohammadi, Omid Ghahroodi, Mahdieh Soleymani Baghshah, and Ehsaneddin Asgari. 2025. Ask in any modality: A comprehensive survey on multimodal retrieval-augmented generation. *Preprint*, arXiv:2502.08826.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, and 8 others. 2022. Flamingo: a visual language model for few-shot learning. *Preprint*, arXiv:2204.14198.

Yuwei An, Yihua Cheng, Seo Jin Park, and Junchen Jiang. 2025. Hyperrag: Enhancing quality-efficiency tradeoffs in retrieval-augmented generation with reranker kv-cache reuse. *Preprint*, arXiv:2504.02921.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, page 129–136, New York, NY, USA. Association for Computing Machinery.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 2024a. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. *Preprint*, arXiv:2403.06764.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. 2024b. Are we on the right way for evaluating large vision-language models? *Preprint*, arXiv:2403.20330.

Shijie Chen, Bernal Jimenez Gutierrez, and Yu Su. 2025a. Attention in large language models yields efficient zero-shot re-rankers. In *The Thirteenth International Conference on Learning Representations*.

Zhanpeng Chen, Chengjin Xu, Yiyan Qi, and Jian Guo. 2025b. MLLM is a strong reranker: Advancing multimodal retrieval-augmented generation via knowledge-enhanced reranking and noise-injected training.

Zhanpeng Chen, Chengjin Xu, Yiyan Qi, Xuhui Jiang, and Jian Guo. 2025c. VLM is a strong reranker: Advancing multimodal retrieval-augmented generation via knowledge-enhanced reranking and noise-injected training. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 8140–8158, Suzhou, China. Association for Computational Linguistics.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. 2024c. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *Preprint*, arXiv:2312.14238.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2024. Dola: Decoding by contrasting layers improves factuality in large language models. *Preprint*, arXiv:2309.03883.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Preprint*, arXiv:2305.06500.

Ziqi Dai, Xin Zhang, Mingxin Li, Yanzhao Zhang, Dingkun Long, Pengjun Xie, Meishan Zhang, Wenjie Li, and Min Zhang. 2025. Supervised fine-tuning or contrastive learning? towards better multimodal llm reranking. *Preprint*, arXiv:2510.14824.

Mohamed Dhouib, Davide Buscaldi, Sonia Vanier, and Aymen Shabou. 2025. Pact: Pruning and clustering-based token reduction for faster visual language models. *Preprint*, arXiv:2504.08966.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. *Preprint*, arXiv:2010.11929.

Hervé Déjean and Stéphane Clinchant. 2025. Reranking with compressed document representation. *Preprint*, arXiv:2505.15394.

Yingqi Fan, Junlong Tong, Anhao Zhao, and Xiaoyu Shen. 2026. What do visual tokens really encode? uncovering sparsity and redundancy in multimodal large language models. *Preprint*, arXiv:2603.00510.

Yingqi Fan, Anhao Zhao, Jinlan Fu, Junlong Tong, Hui Su, Yijie Pan, Wei Zhang, and Xiaoyu Shen. 2025. $\mathcal{V}isi\mathcal{P}runer$: Decoding discontinuous cross-modal dynamics for efficient multimodal llms. *Preprint*, arXiv:2510.17205.

Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. 2024. Colpali: Efficient document retrieval with vision language models. *Preprint*, arXiv:2407.01449.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. 2025. Mme: A comprehensive evaluation benchmark for multimodal large language models. *Preprint*, arXiv:2306.13394.

Nilesh Gupta, Chong You, Srinadh Bhojanapalli, Sanjiv Kumar, Inderjit S Dhillon, and Felix X. Yu. 2025. Scalable in-context ranking with generative models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Haohang Huang, Xuan Lu, Mingyi Su, Xuan Zhang, Ziyan Jiang, Ping Nie, Kai Zou, Tomas Pfister, Wenhu Chen, Wei Zhang, Xiaoyu Shen, and Rui Meng. 2026. MMEB-v3: Measuring the performance gaps of omni-modality embedding models. In *Third Conference on Language Modeling*.

Drew A. Hudson and Christopher D. Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. *Preprint*, arXiv:1902.09506.

Pengfei Jiang, Hanjun Li, Linglan Zhao, Fei Chao, Ke Yan, Shouhong Ding, and Rongrong Ji. 2025a. Visa: Group-wise visual token selection and aggregation via graph summarization for efficient mllms inference. In *Proceedings of the 33rd ACM International Conference on Multimedia*, MM '25, page 11130–11139. ACM.

Ziyan Jiang, Rui Meng, Xinyi Yang, Semih Yavuz, Yingbo Zhou, and Wenhu Chen. 2025b. Vlm2vec: Training vision-language models for massive multimodal embedding tasks. *Preprint*, arXiv:2410.05160.

Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. 2025. See what you are told: Visual attention sink in large multimodal models. *Preprint*, arXiv:2503.03321.

Vedang Lad, Jin Hwa Lee, Wes Gurnee, and Max Tegmark. 2025. The remarkable robustness of llms: Stages of inference? *Preprint*, arXiv:2406.19384.

Minghan Li, Eric Gaussier, and Guodong Zhou. 2026a. Efficient long-document reranking via block-level embeddings and top-k interaction refinement. *Preprint*, arXiv:2501.17039.

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. 2026b. Qwen3-vl-embedding and qwen3-vl-reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. *Preprint*, arXiv:2601.04720.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Evaluating object hallucination in large vision-language models. *Preprint*, arXiv:2305.10355.

Junyan Lin, Haoran Chen, Dawei Zhu, and Xiaoyu Shen. 2024. To preserve or to compress: An in-depth study of connector selection in multimodal large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5666–5680, Miami, Florida, USA. Association for Computational Linguistics.

Sheng-Chieh Lin, Chankyu Lee, Mohammad Shoeybi, Jimmy Lin, Bryan Catanzaro, and Wei Ping. 2025. MM-EMBED: UNIVERSAL MULTIMODAL RETRIEVAL WITH MULTIMODAL LLMS. In *The Thirteenth International Conference on Learning Representations*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. *Preprint*, arXiv:2310.03744.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024b. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Wenjie Liu, Hao Wu, Xin Qiu, Xudong Wang, Yingqi Fan, Yihan Zhang, Anhao Zhao, Yunpu Ma, and Xiaoyu Shen. 2026. Vica: Efficient multimodal llms with vision-only cross-attention. *Preprint*, arXiv:2602.07574.

Yikun Liu, Yajie Zhang, Jiayin Cai, Xiaolong Jiang, Yao Hu, Jiangchao Yao, Yanfeng Wang, and Weidi Xie. 2025. Lamra: Large multimodal model as your advanced retrieval assistant. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4015–4025. IEEE.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. 2024c. Mmbench: Is your multi-modal model an all-around player? *Preprint*, arXiv:2307.06281.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Preprint*, arXiv:2209.09513.

Xuan Lu, Haohang Huang, Yingqi Fan, Junlong Tong, Yuxuan Zhang, Ping Nie, Rui Meng, and Xiaoyu Shen. 2026a. Comprank: Efficient llm reranking via token-level compression and decoding-free scoring. *Preprint*, arXiv:2606.11700.

Xuan Lu, Haohang Huang, Rui Meng, Yaohui Jin, Wenjun Zeng, and Xiaoyu Shen. 2026b. Rethinking reasoning in document ranking: Why chain-of-thought falls short. In *The Fourteenth International Conference on Learning Representations*.

Xuan Lu, Haohang Huang, Rui Meng, Yaohui Jin, Wenjun Zeng, and Xiaoyu Shen. 2026c. Tools are underdocumented: Simple document expansion boosts tool retrieval. In *The Fourteenth International Conference on Learning Representations*.

Xuan Lu, Kangle Li, Haohang Huang, Rui Meng, Wenjun Zeng, and Xiaoyu Shen. 2026d. Beyond global similarity: Towards fine-grained, multi-condition multimodal retrieval. *Preprint*, arXiv:2603.01082.

Xuan Lu, Sifan Liu, Bochao Yin, Yongqi Li, Xinghao Chen, Hui Su, Yaohui Jin, Wenjun Zeng, and Xiaoyu Shen. 2025. MultiConIR: Towards multi-condition information retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 13471–13494, Suzhou, China. Association for Computational Linguistics.

Rui Meng, Ziyan Jiang, Ye Liu, Mingyi Su, Xinyi Yang, Yuepeng Fu, Can Qin, Zeyuan Chen, Ran Xu, Caiming Xiong, Yingbo Zhou, Wenhu Chen, and Semih Yavuz. 2025. Vlm2vec-v2: Advancing multimodal embedding for videos, images, and visual documents. *Preprint*, arXiv:2507.04590.

Rodrigo Nogueira and Kyunghyun Cho. 2020. Passage re-ranking with bert. *Preprint*, arXiv:1901.04085.

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with bert. *Preprint*, arXiv:1910.14424.

nostalgebraist. 2020. Interpreting gpt: The logit lens.

Ruoyu Qin, Weiran He, Yaoyu Wang, Zheming Li, Xinran Xu, Yongwei Wu, Weimin Zheng, and Mingxing Zhang. 2026. Prefill-as-a-service: Kvcache of next-generation models could go cross-datacenter. *Preprint*, arXiv:2604.15039.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. 2024. Large language models are effective text rankers with pairwise ranking prompting. *Preprint*, arXiv:2306.17563.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. *Preprint*, arXiv:1904.08920.

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. 2025. Layer by layer: Uncovering hidden representations in language models. *Preprint*, arXiv:2502.02013.

Yiqun Sun, Pengfei Wei, and Lawrence B. Hsieh. 2026. Very efficient listwise multimodal reranking for long documents. *Preprint*, arXiv:2605.11864.

Junlong Tong, Yingqi Fan, Anhao Zhao, Yunpu Ma, and Xiaoyu Shen. 2026a. Streamingthinker: Large language models can think while reading. *Preprint*, arXiv:2510.17238.

Junlong Tong, Yao Zhang, Anhao Zhao, Yingqi Fan, Yunpu Ma, and Xiaoyu Shen. 2026b. Proactivellm: Learning active interaction for streaming large language models. *Preprint*, arXiv:2606.00523.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu,

Chang Zhou, Jingren Zhou, and Junyang Lin. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *Preprint*, arXiv:2409.12191.

Cong Wei, Yang Chen, Haonan Chen, Hexiang Hu, Ge Zhang, Jie Fu, Alan Ritter, and Wenhu Chen. 2023. Uniir: Training and benchmarking universal multimodal information retrievers. *Preprint*, arXiv:2311.17136.

Hao Wu, Yingqi Fan, Dai Jinyang, Junlong Tong, Yunpu Ma, and Xiaoyu Shen. 2026a. Hidrop: Hierarchical vision token reduction in MLLMs via late injection, concave pyramid pruning, and early exit. In *The Fourteenth International Conference on Learning Representations*.

Hao Wu, Junlong Tong, Xudong Wang, Yang Tan, Changyu Zeng, Anastasia Antsiferova, and Xiaoyu Shen. 2026b. From data to model: A survey of the compression lifecycle in mllms.

Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. 2008. Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 1192–1199, New York, NY, USA. Association for Computing Machinery.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. *Preprint*, arXiv:2309.17453.

Mingjun Xu, Jinhan Dong, Jue Hou, Zehui Wang, Sihang Li, Zhifeng Gao, Renxin Zhong, and Hengxing Cai. 2025. Mm-r5: Multimodal reasoning-enhanced reranker via reinforcement learning for document retrieval. *arXiv preprint arXiv:2506.12364*.

Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Junhao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, and Maosong Sun. 2025. Visrag: Vision-based retrieval-augmented generation on multi-modality documents. *Preprint*, arXiv:2410.10594.

Ruohong Zhang, Liangke Gui, Zhiqing Sun, Yihao Feng, Keyang Xu, Yuanhan Zhang, Di Fu, Chunyuan Li, Alexander Hauptmann, Yonatan Bisk, and 1 others. 2024. Direct preference optimization of video large multimodal models from language model reward. *arXiv preprint arXiv:2404.01258*.

Wuwei Zhang, Fangcong Yin, Howard Yen, Danqi Chen, and Xi Ye. 2025a. Query-focused retrieval heads improve long-context reasoning and re-ranking. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 23791–23805, Suzhou, China. Association for Computational Linguistics.

Xin Zhang, Yanzhao Zhang, Wen Xie, Mingxin Li, Ziqi Dai, Dingkun Long, Pengjun Xie, Meishan Zhang, Wenjie Li, and Min Zhang. 2025b. Bridging modalities: Improving universal multimodal retrieval by multimodal large language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9274–9285.

Anhao Zhao, Junlong Tong, Yingqi Fan, Ping Nie, Wenjie Li, and Xiaoyu Shen. 2026. Poweropd: Stabilizing on-policy distillation with bounded power transformation. *Preprint*, arXiv:2606.17199.

Anhao Zhao, Fanghua Ye, Yingqi Fan, Junlong Tong, Zhiwei Fei, Hui Su, and Xiaoyu Shen. 2025. Skipgpt: Dynamic layer pruning reinvented with token awareness and module decoupling. *Preprint*, arXiv:2506.04179.

Wangbo Zhao, Yizeng Han, Jiasheng Tang, Zhikai Li, Yibing Song, Kai Wang, Zhangyang Wang, and Yang You. 2024. A stitch in time saves nine: Small vlm is a precise guidance for accelerating large vlms. *Preprint*, arXiv:2412.03324.

Changjiang Zhou, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Fan Yixing, and Xueqi Cheng. 2026. Longranker: Efficient one-pass document reranking with long-context large language models. In *Proceedings of the ACM Web Conference 2026*, pages 2004–2013.

A Related Work

MLLM-based Reranking and Retrieval. Recent multimodal retrieval systems increasingly adopt unified embedding models that support diverse modalities and instruction-conditioned retrieval (Zhang et al., 2025b; Lin et al., 2025; Li et al., 2026b). While recent work explores MLLMs as rerankers using cross-encoder formulations. Prior studies show that instruction-tuned MLLMs can serve as effective multimodal relevance judges through point-wise yes/no scoring (Chen et al., 2025c; Liu et al., 2025; Lin et al., 2025; Li et al., 2026b). Other work additionally studies reasoning-enhanced reranking and application-specific multimodal retrieval settings (Xu et al., 2025; Lu et al., 2026d). These studies collectively show the effectiveness of MLLM-based reranking. In contrast, our work targets the computational redundancy of point-wise multimodal reranking itself.

Efficient LLM-based Reranking. Prior work on efficient reranking mainly focuses on reducing autoregressive decoding, compressing long-context interactions, or caching document-side computation. Some methods replace generation with attention-based relevance estimation (Chen et al., 2025a; Zhang et al., 2025a), while others improve list-wise or long-context reranking through sparse attention, hierarchical ranking, or token pruning (Zhou et al., 2026; Gupta et al., 2025; Sun et al., 2026). Another line of work reduces online reranking cost via document-side caching or compressed document representations (An et al., 2025; Déjean and Clinchant, 2025; Li et al., 2026a). While they mainly target text or document-centric reranking, our work provides a unified efficiency framework for multimodal point-wise rerankers.

B Early Exit: Layer-wise Logit Probing

B.1 General VQA Tasks

To examine whether the early-exit behavior is specific to point-wise reranking, we further evaluate layer-wise probing on general VQA-style tasks

Prefill-only Tasks. We consider two types of *prefill-only* tasks: (1) yes/no tasks, including POPE (Li et al., 2023) and MME (Fu et al., 2025); and (2) multiple-choice tasks, including MMBench (Liu et al., 2024c), ScienceQA (Lu et al., 2022), and MMStar (Chen et al., 2024b). As shown in Fig. 7, both task types exhibit a much later emergence of reliable prediction signals than reranking. Across these datasets, the intermediate layers before around layer 20 generally fail to recover the final-layer performance, while performance only becomes comparable to the final layer at around layer 22 or later. This trend suggests that general VQA tasks require deeper-layer computation to form stable answer predictions, in contrast to point-wise reranking where strong relevance signals already emerge in intermediate layers.

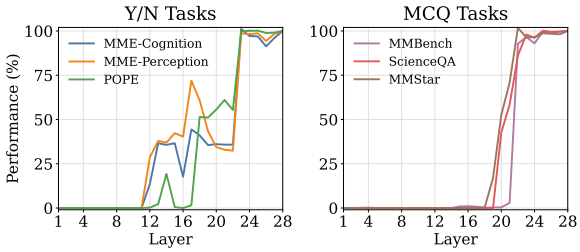

Figure 7: **Layer-wise probing on general VQA tasks.** We evaluate prefill-only yes/no tasks and multiple-choice tasks using intermediate-layer logits. Unlike point-wise reranking, general VQA tasks only recover final-layer performance at much deeper layers, typically around layer 22 or later.

Open-ended Tasks. We also evaluate open-ended VQA tasks, where the model needs to generate free-form answers. Since full layer-wise probing with decoding is computationally expensive, we only test representative early-exit layers selected in our reranking experiments. Specifically, we evaluate GQA (Hudson and Manning, 2019) and TextVQA (Singh et al., 2019) at the 16-th layer, which is effective for our point-wise reranking setting. However, early exiting at this layer fails to produce correct answers on these open-ended tasks, resulting in 0.0 scores on both datasets. This further indicates that open-ended generation depends more heavily on late-layer computation, including answer formulation and token-level decoding refinement.

Overall, these results show a clear difference between general VQA and point-wise reranking. While general VQA tasks require deeper layers to support answer prediction or generation, point-wise reranking mainly relies on relevance discrimination signals that emerge much earlier. This supports our motivation for applying early exit specifically to multimodal reranking rather than treating it as a generic compression strategy for all multimodal tasks.

B.2 Finetuned Models

We further analyze the layer-wise reranking behavior of Qwen3-VL-2B-Instruct† after supervised finetuning under different prompt formulations.

Specifically, we perform layer-wise logit probing on models finetuned with the query-first, document-first, and visual-first prompting strategies.

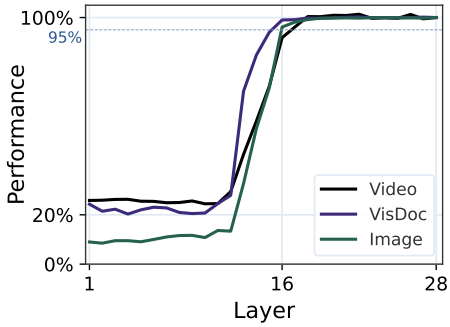

Figure 8: **Qwen3-VL-2B-reranker**^{DF†}.

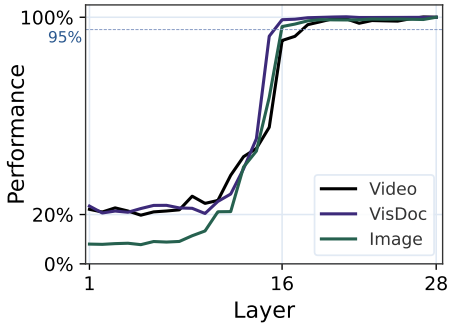

Figure 9: **Qwen3-VL-2B-reranker**^{VF†}.

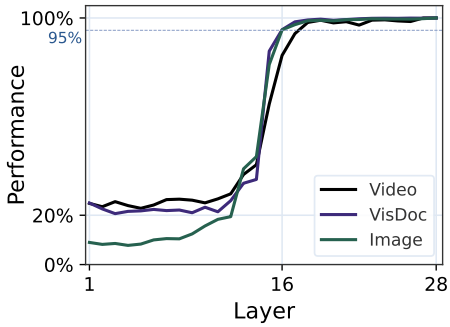

Figure 10: **Qwen3-VL-2B-reranker**^{QF†}.

As shown in Fig. 8, Fig. 9 and Fig. 10 , finetuning with the binary yes/no reranking objective substantially changes the distribution of reranking signals across layers. Compared with the pretrained model, the finetuned models achieve stronger reranking performance at intermediate layers, reaching better performance around layer 16 while also exhibiting more stable behavior in deeper layers. This observation further supports our early-exit analysis, suggesting that reranking-relevant representations emerge relatively early in the transformer stack after task-specific finetuning.

Moreover, the three prompt formulations exhibit highly similar layer-wise trends after finetuning. This suggests that prompt formatting mainly affects efficiency and computation reuse, while having relatively limited influence on where reranking signals emerge across transformer layers.

C Training Configuration

C.1 Token-Pruned Multimodal Reranker Training Set

We construct a token-pruned multimodal reranker training set for learning fine-grained query-document relevance across heterogeneous modalities. Each example is represented as a query-document pair (q, d) with a binary relevance label. Positive candidates are annotated with `label=yes`, while negative candidates are annotated with `label=no`. The negative set contains both hard negatives and randomly sampled negatives, enabling the reranker to learn both subtle semantic distinctions and broad modality-level discrimination.

Dataset Composition. Our training data is constructed from four multimodal sources: MMEB-train (Jiang et al., 2025b) for image-related tasks, ViDoRe and VisRAG (Faysse et al., 2024; Yu et al., 2025) for visual document retrieval, and ShareGPTVideo (Zhang et al., 2024) for video-language retrieval. The resulting dataset contains 736,327 training pairs from 25 source subsets, covering a diverse collection of retrieval and reranking scenarios, including image-to-text, text-to-image, image-to-image, text-to-visual-document, text-to-video, video-to-text, and visual-question-to-text matching. In total, the dataset contains 147,899 positive pairs and 588,428 negative pairs, where the negative pairs consist of 294,375 hard negatives and 294,053 random negatives.

Negative Sampling. We convert each query into multiple point-wise training pairs by pairing every positive instance with four negatives, including two random negatives and two hard negatives. For image and visdoc tasks, the hard negatives are sampled from the top-20 retrieved candidates using the `Qwen3-VL-Embedding-2B`. For video tasks, the hard negatives are sampled from the top-5 retrieved candidates using the `Qwen3-VL-Embedding-2B`.

Data Schema. Each sample follows a unified schema. The fields `subset` and `source_row_idx` identify the original data source and row index. The field `candidate_type` specifies whether the candidate is a positive example, a hard negative, or a random negative. The fields `instruction`,

Subset	Task Type	#Samples
A-OKVQA	Image+Text → Text	8,525
CIRR	Image+Text → Image	13,055
ChartQA	Image+Text → Text	14,145
DocVQA	Image+Text → Text	19,730
HatefulMemes	Image → Text	1,700
ImageNet-1K	Image → Text	50,000
InfographicsVQA	Image+Text → Text	11,970
MSCOCO	Image → Image	50,000
MSCOCO i2t	Image → Text	56,640
MSCOCO t2i	Text → Image	50,000
N24News	Image → Text	24,490
NIGHTS	Image+Text → Image	7,970
OK-VQA	Image+Text → Text	4,500
SUN397	Image → Text	9,925
VOC2007	Image → Text	3,920
VisDial	Text → Image+Text	61,640
Visual7W	Image+Text → Text	34,905
VisualNews i2t	Image → Text	50,000
VisualNews t2i	Text → Image+Text	49,950
WebQA	Text → Image+Text	8,580
ViDoRe	Text → VisDoc	59,095
VisRAG	Text → VisDoc	61,375
ShareGPTVideo t2v	Text → Video	29,971
ShareGPTVideo v2t	Video → Text	29,998
ShareGPTVideo QA	Video+Text → Text	24,243

Table 5: Composition of the token-pruned multimodal reranker training set.

`query_text`, and `document_text` provide the task instruction and textual content on the query and document sides. Visual inputs are stored either as image bytes, through `query_image_bytes` and `document_image_bytes`, or as sampled video-frame paths, through `query_video_frame_paths` and `document_video_frame_paths`. The field `doc_modality` records the document-side modality, such as `text`, `image`, `image+text`, or `video`.

Token-Pruning Metadata. In addition to the original multimodal content, the dataset includes token-pruning metadata. The fields `query_visual_modalities` and `document_visual_modalities` indicate which visual modalities are present on each side. The fields `query_visual_token_counts` and `document_visual_token_counts` record the number of visual tokens for each visual input, while `query_visual_ranked_tokens` and `document_visual_ranked_tokens` store token-importance rankings. The token rankings are computed following subsection 3.2, by summing the attention scores from the last token to all visual tokens across all layers of `Qwen3-VL-2B-Embedding`. The rankings for visual inputs are saved during hard negative mining and therefore require no additional computation. These rankings allow the reranker to perform efficient visual token selection while preserving the original supervision signal of each query-document pair.

C.2 Detailed Configuration

We use `Qwen3-VL-Instruct` as the backbone model and fine-tune only the LLM component with a supervised point-wise reranking objective. All models are trained for one epoch using DeepSpeed ZeRO-2 with bfloat16 precision. The 2B model is trained on 4×NVIDIA A100-SXM4-40GB GPUs, while the 4B and 8B models are trained on 4×NVIDIA H100 GPUs. We use a per-device batch size of 4 with gradient accumulation steps of 1.

Optimization is performed with AdamW using a learning rate of 1×10^{-4}, zero weight decay, cosine learning rate scheduling, and a warmup ratio of 0.03. For parameter-efficient fine-tuning, we adopt LoRA with rank $r = 16$, scaling factor $\alpha = 32$, and dropout rate 0.05. Unless otherwise specified, all experiments use the same training configuration.

D Evaluation

D.1 MMEB-V2

We provide a detailed summary of the evaluation tasks in Tab. 6. The benchmark covers three major multimodal domains: image-centric tasks, video-centric tasks, and visual document retrieval. For each task group, we report the meta-task category, source datasets, retrieval direction, number of queries, and candidate pool size. The retrieval direction is denoted by the input and target modalities, where I, T, V, and D represent image, text, video, and visual document, respectively.

Overall, the evaluation includes a broad range of multimodal retrieval and reranking scenarios, such as image classification, visual question answering, image retrieval, visual grounding, video retrieval, moment retrieval, video classification, video question answering, and visual document retrieval. This diverse task coverage allows us to evaluate whether the proposed reranking framework generalizes across different modality combinations, retrieval directions, and candidate set sizes.

D.2 Baselines

ColPali v1.3 ColPali is a specialized visual document retriever designed for page-level document

Meta-Task	Dataset	Retrieval	#Query	#Cand.
Image Classification	ImageNet-1K, N24News, HatefulMemes, VOC2007, SUN397, Place365, ImageNet-A/R, ObjectNet, Country-211	I→T, I+T→T	1,000	2–1,000
Image VQA	OK-VQA, A-OKVQA, DocVQA, InfographicVQA, ChartQA, Visual7W, ScienceQA, VizWiz, GQA, TextVQA	I+T→T	1,000	1,000
Image Retrieval	VisDial, CIRR, VisualNews, MSCOCO, NIGHTS, WebQA, OVEN, FashionIQ, EDIS, WikiSS-NQ	T→I, I→T, I+T→I/T	1,000	1,000
Visual Grounding	MSCOCO, Visual7W-Pointing, RefCOCO, RefCOCO-Matching	I+T→I/T	1,000	1,000
Video Retrieval	DiDeMo, MSR-VTT, MSVD, VATEX, YouCook2	T→V	670–4,468	670–4,468
Moment Retrieval	QVHighlights, Charades-STA, MomentSeeker	T+V→V	727–1,800	10
Video Classification	Kinetics-700, SSv2, HMDB51, UCF101, Breakfast	V→T	433–1,000	10–700
Video QA	MVBench, Video-MME, NExT-QA, EgoSchema, ActivityNetQA	V+T→T	500–8,564	2–5
Visual Document Retrieval	ViDoRe, ViDoRe-V2, VisRAG, ViDoSeek, MMLongBench-Doc	T→D	52–1,646	70–9,590

Table 6: Evaluation dataset composition. We summarize the evaluated datasets by meta-task, retrieval format, number of queries, and candidate size. I, T, V, and D denote image, text, video, and visual document, respectively.

retrieval. Instead of relying on OCR-based text extraction, it directly embeds document page images with a VLM and produces ColBERT-style multi-vector representations for late-interaction matching (Faysse et al., 2024). In our comparison, ColPali v1.3 serves as a strong visual-document retrieval baseline, particularly for tasks where layout, tables, figures, and page-level visual cues are important.

General Multimodal Embedder (GME) GME is an instruction-aware multimodal dense retriever built on MLLMs. It represents text, images, visual documents, and image-text composed inputs in a unified embedding space, and is trained with contrastive learning over diverse single-modal, cross-modal, and fused-modal retrieval data (Zhang et al., 2025b). We include GME as a general-purpose multimodal embedding baseline, since it supports a broad range of retrieval scenarios beyond visual document retrieval.

VLM2Vec VLM2Vec is a contrastive training framework that converts existing vision-language models into universal multimodal embedding models (Meng et al., 2025). Unlike CLIP-style models that encode images and text independently without task instructions, VLM2Vec can process arbitrary combinations of image and text inputs and produce task-conditioned fixed-dimensional embeddings. We use it as an instruction-aware multimodal embedding baseline trained for diverse embedding tasks such as classification, VQA, multimodal retrieval, and visual grounding.

Qwen3-VL-Reranker-2B Qwen3-VL-Reranker-2B is a multimodal cross-encoder reranker built on Qwen3-VL. Given a query-document pair, where both sides may contain text, images, screenshots, videos, or mixed-modal inputs, it performs fine-grained relevance estimation and outputs a relevance score (Li et al., 2026b). Compared with embedding-based baselines, this model is closer to our setting because it jointly encodes each query-document pair for reranking. We therefore use it as the dense reranker baseline before applying our compression strategy.

D.3 More detailed configurations

Following Qwen3-VL-Reranker (Li et al., 2026b), we constrain image inputs to 4–1800 visual tokens (4096–1,843,200 pixels), and video inputs to at most 64 frames sampled at 1 FPS with a total budget of 7,864,320 pixels; the maximum input length is capped at 10,240 tokens.

During evaluation, we use the raw logit difference between the `yes` and `no` tokens as the reranking score, instead of applying a sigmoid normalization. Since the sigmoid function is monotonic, this does not change the ranking order. In practice, using raw logits avoids numerical saturation when the logit difference is very large or very small, which can otherwise make multiple samples receive indistinguishable scores after sigmoid normalization. To further ensure evaluation robustness, we randomly shuffle the candidate order before reranking, so that the results are not affected by the retrieval model's original candidate ordering.

D.4 Prompt Templates

We use a unified point-wise prompt format for all reranking tasks, where the model is required to judge whether a candidate document satisfies the given query and instruction by generating only `yes` or `no`. Since the reusable visual segment differs across retrieval settings, we adopt two visual-first variants depending on whether the visual input is associated with the query or the document.

Vision-as-query. For tasks where the visual input belongs to the query, e.g., image-to-text or video-to-text retrieval, we place the query before the candidate document. This allows the query-side visual

representations to be computed once and reused across all candidate documents for the same query.

Input Template for Vision-as-query Reranking

```
<|im_start|>system
Judge whether the Document meets the
    requirements based on the Query and the
    Instruct provided.
Note that the answer can only be "yes" or "no
    ".
<|im_end|>

<|im_start|>user
<Instruct>: {Instruction}
<Query>: {Query}
<Document>: {Document}
<|im_end|>
<|im_start|>assistant
```

Figure 11: Prompt template for vision-as-query reranking, where the query-side visual input is placed before the candidate document to enable reuse across candidates.

Vision-as-document. For tasks where the visual input belongs to the document, e.g., text-to-image or image-to-image retrieval, we instead place the document before the query. This makes the document-side visual representations independent of the incoming query, enabling reuse across different queries.

Input Template for Vision-as-document Reranking

```
<|im_start|>system
Judge whether the Document meets the
    requirements based on the Query and the
    Instruct provided.
Note that the answer can only be "yes" or "no
    ".
<|im_end|>

<|im_start|>user
<Instruct>: {Instruction}
<Document>: {Document}
<Query>: {Query}
<|im_end|>
<|im_start|>assistant
```

Figure 12: Prompt template for vision-as-document reranking, where the document-side visual input is placed before the query to enable reuse across queries.

E Ablation on Visual Token Selection

To further validate the effectiveness of our embedder-attention-guided token selection strategy, we compare it with several alternative visual token selection methods. For fair comparison, all methods prune visual tokens before they are fed into the reranker LLM, and therefore reduce the same subsequent LLM-side computation. We only consider image-intrinsic selection criteria that depend on the visual input itself, rather than query-document relevance labels or external supervision. In addition, none of the compared methods introduces extra model parameters or additional forward passes. Under this setting, we evaluate two representative alternatives: similarity-based selection using the embedder representations and attention-based selection using the reranker's vision encoder.

E.1 <Emb> Similarities

Methods. A natural alternative is to select visual tokens according to their similarity to the final embedding token. Specifically, we use the hidden representation of the final <Emb> token produced by the retrieval-stage embedder as a global summary of the input, and compute its cosine similarity with each visual token representation. Visual tokens with higher similarity scores are retained. We evaluate this strategy using representations from different layers, including the 3rd layer, the 14th layer, the last layer, and an all-layer variant that aggregates similarity scores across layers.

Results. As shown in Tab. 7, similarity-based selection provides a simple way to estimate visual token importance, but it is less effective than our attention-based embedder-guided strategy. This suggests that direct representation similarity to the final <Emb> token does not fully capture which visual tokens are most useful for reranking. In contrast, our method uses the attention distribution from the final embedding token, which better reflects how the embedder aggregates visual evidence for retrieval-oriented representation learning.

	Image				
Selection	CLS	QA	RET	GD	**Overall**
Dense	59.1	61.2	51.4	77.3	59.1
Simi(3rd Layer)	56.7	54.6	50.2	65.8	54.9
Simi(14th Layer)	56.9	58.3	50.2	65.4	56.0
Simi(Last Layer)	**58.3**	58.8	50.5	63.5	56.4
Simi(All Layers)	57.6	58.9	50.3	64.5	56.3
Ours	58.1	**60.8**	**51.2**	**72.6**	**58.2**

Table 7: Ablation of similarity-based visual token selection using the embedder <Emb> representation.

E.2 Reranker ViT Attention

We also compare with token selection based on the reranker's own vision encoder. This method ranks

visual tokens by their attention scores inside the ViT, where tokens receiving larger accumulated attention are regarded as more important. Similar to the similarity-based setting, we evaluate attention scores from the 3rd layer, the 14th layer, the last layer, and an all-layer aggregation variant. Since these scores are obtained from the reranker's visual encoding process itself, this baseline does not require an additional model or extra forward computation.

Results. Tab. 8 reports the comparison with ViT-attention-based selection. Although ViT attention captures visual saliency within the reranker's vision encoder, it is not explicitly optimized for retrieval or reranking. As a result, visually salient tokens are not always the most useful tokens for relevance estimation. Our embedder-guided strategy performs better because the selection signal comes from the retrieval-stage embedder, whose attention patterns are more directly aligned with the downstream reranking objective.

	Image				
Selection	CLS	QA	RET	GD	**Overall**
Dense	59.1	61.2	51.4	77.3	59.1
ViT(3rd Layer)	54.7	57.5	48.7	62.3	54.3
ViT(14th Layer)	54.6	58.0	47.4	71.4	55.0
ViT(Last Layer)	57.4	57.1	**51.2**	**73.4**	57.0
ViT(All Layers)	56.3	59.0	48.9	69.3	56.0
Ours	**58.1**	**60.8**	**51.2**	72.6	**58.2**

Table 8: Ablation of visual token selection using attention scores from the reranker's ViT.

F Efficiency Analysis

F.1 Precache Time

Main-text latency results focus on *online reranking latency*, where visual representations are assumed to be precomputed and cached. This setting reflects practical retrieval systems, in which visual features are typically reused across many reranking pairs. Moreover, cache construction can be performed either offline or online depending on the deployment setting, making the preprocessing overhead system-dependent and amortizable across repeated reuse. For completeness, we further report end-to-end latency including visual pre-encoding and cache construction overhead. Although precaching introduces additional preprocessing cost, the overhead can be amortized through repeated reuse during reranking.

Vision-as-query setting. In this setting, the visual query representation is precached once and reused across multiple candidate documents. We therefore measure the end-to-end latency including both cache construction and reranking computation while varying the number of candidates. As shown in Fig. 13, although precaching introduces additional preprocessing overhead, the cost remains relatively small compared with the overall reranking computation. The overhead is further amortized as the number of candidate documents increases, leading to substantially lower end-to-end latency than the vanilla reranking pipeline.

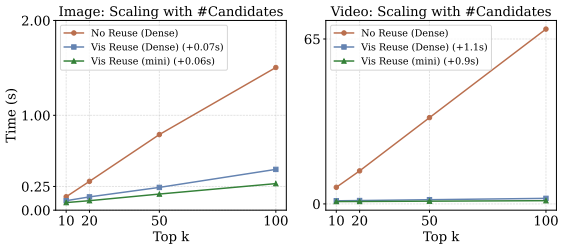

Figure 13: End-to-end latency including visual pre-encoding and cache construction overhead in the vision-as-query setting, measured under different numbers of candidate documents.

Vision-as-document setting. In this setting, document-side visual representations are cached once and reused across many incoming queries. Since the reuse frequency increases with the number of queries, we report latency scaling with respect to the number of queries. As shown in Fig. 14, the upfront cache construction cost becomes increasingly negligible as reuse grows. Consequently, despite including visual pre-encoding and cache construction overhead, the proposed reuse strategy still achieves substantially lower end-to-end latency compared with the vanilla reranking pipeline, particularly in large-scale retrieval scenarios.

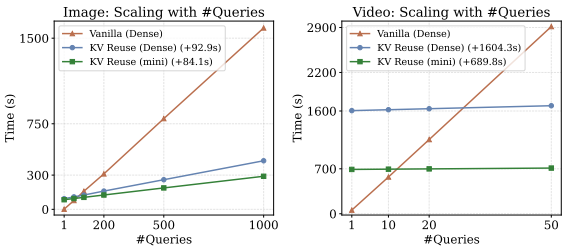

Figure 14: End-to-end latency including visual pre-encoding and cache construction overhead in the vision-as-document setting, measured under different numbers of queries.