WHY ONLY THE FINAL THOUGHT? HIERARCHICAL AGGREGATION FOR MULTIMODAL RETRIEVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal large language models (MLLMs) have emerged as powerful backbones for learning embeddings for multimodal retrieval. Recent methods incorporate chain-of-thought (CoT) reasoning into embedding generation and further employ adaptive routing mechanisms to selectively invoke reasoning and control its computational cost. However, different queries may benefit from embeddings formed with different reasoning-token budgets and at different model depths, whereas existing methods typically rely on a single final embedding. To address this limitation, we introduce a dual-aggregator framework that exploits these embeddings, adaptively combining them for each query and aggregating their matching evidence to rank candidates. Experiments on MMEB-V2 validate the benefits of integrating information across reasoning-token budgets and model depths, with our framework achieving strong overall retrieval performance with less than 1% additional parameters over the backbone. Further analytical experiments demonstrate that our framework adapts its aggregation to individual queries, effectively integrating information across budgets and depths to improve retrieval performance.

1 INTRODUCTION

Multimodal large language models (MLLMs) have made substantial progress in understanding information across modalities Bai et al. (2025); Radford et al. (2021); Chen et al. (2026). These capabilities have enabled MLLMs to serve as unified encoders for multimodal retrieval, mapping heterogeneous inputs into a shared embedding space. While MLLM-based embedding models already exhibit strong semantic understanding, recent work shows that incorporating chain-of-thought (CoT) reasoning can further improve retrieval on complex queries Cui et al. (2026).

As illustrated in Figure 1(a), *Direct Embedding* maps multimodal inputs to embeddings in a single forward pass, without explicit CoT generation. *Explicit CoT-driven Embedding* augments embedding generation with textual reasoning traces produced either by an external reasoner or by the embedding model itself (Cui et al., 2026; Lan et al., 2026; Jiang et al., 2026; Wang et al., 2026). To reduce the overhead of explicit text generation, *Latent CoT-driven Embedding* performs reasoning through a sequence of continuous latent states He et al. (2026). Alternatively, *Selective-CoT Embedding* adaptively determines whether to invoke reasoning for a given query, avoiding unnecessary CoT generation Wang et al. (2026); Zhang et al. (2026).

However, this binary decision offers only a coarse form of adaptation, as queries that benefit from reasoning may still require different reasoning intensities. Beyond reasoning intensity, probing studies show that intermediate MLLM layers can preserve global semantic information more effectively than the topmost layers, which may emphasize local details (Tao et al., 2024). These findings motivate exploring retrieval representations at different decoder depths. Yet retrieval typically relies on a single final representation, leaving such information underutilized. This raises a key question: *how can retrieval make better use of representations formed throughout the reasoning process?*

To address this question, we introduce *DUET*, a dual-aggregator framework that adaptively draws on query embeddings formed with different reasoning budgets and at different model depths. We obtain these embeddings by varying the number of continuous latent reasoning tokens and extracting representations at multiple layers. At each depth, a *Query Aggregator* learns query-dependent weights

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

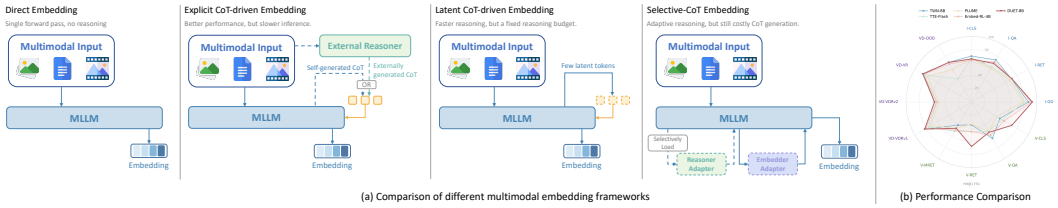

Figure 1: Comparison of multimodal embedding paradigms and retrieval performance. (a) Direct Embedding produces an embedding without CoT. Explicit CoT-driven Embedding incorporates textual reasoning generated either by the embedding model itself or by an external reasoner. Latent CoT-driven Embedding performs reasoning through continuous latent states, while Selective-CoT Embedding adaptively invokes reasoning when needed. (b) Performance comparison of DUET-8B and baselines across 12 MMEB-V2 task groups spanning image, video, and visual-document modalities. Scores are reported as Hit@1 (%) for Image and Video, and NDCG@5 for VisDoc tasks.

to combine embeddings across reasoning budgets. A *Candidate Aggregator* then ranks the retrieved candidates by integrating matching evidence across embeddings and modeling interactions among competing candidates. Our contributions are threefold.

- We establish **budget–depth representation diversity** as a resource for multimodal retrieval. DUET constructs a set of query representations across reasoning-token budgets and decoder depths, exploiting complementary retrieval evidence beyond the final reasoning output.
- We introduce a **dual-aggregator framework** that learns both how to combine query representations and how to use them for candidate ranking. Query-dependent aggregation captures varying reasoning preferences, while candidate-aware aggregation preserves and integrates matching evidence across representations.
- We achieve **state-of-the-art overall performance among comparable methods on MMEB-V2** (Meng et al. (2025)), with less than 1% additional parameters over the backbone. Controlled ablations support the value of budget and depth diversity and both aggregators, while attention analyses and representation interventions reveal how DUET uses different reasoning representations.

2 Related Work

2.1 Universal Multimodal Embedding

Universal multimodal embedding requires a shared representation space that accommodates heterogeneous inputs and diverse retrieval objectives. Beyond image–text alignment in CLIP (Radford et al. (2021)) and SigLIP (Zhai et al. (2023)), VLM2Vec (Jiang et al. (2024)) learns instruction-conditioned embeddings, while UniME (Gu et al. (2025)) combines knowledge distillation with instruction tuning using hard negatives. Research on training supervision spans multimodal data synthesis in GME (Zhang et al. (2024)) and MegaPairs (Zhou et al. (2025)), modality-aware negative mining in MM-Embed (Lin et al. (2025)), and hardness-aware contrastive learning in LLaVE (Lan et al. (2025)). UniME-V2 (Gu et al. (2026)) further uses MLLMs as judges to refine negative selection and provide soft semantic supervision. These efforts improve training coverage and the ability to distinguish relevant candidates from negatives. At the pipeline level, LamRA (Liu et al. (2025b)) and UniME-V2 also explore learned reranking to improve candidate selection after retrieval. Meanwhile, VLM2Vec-V2 (Meng et al. (2025)) extends modality support to videos and visual documents and introduces MMEB-V2 for broader evaluation. Collectively, these studies advance task adaptation, training supervision, and retrieval design, while their embedding models generally represent inputs through direct encoding. Reasoning-enhanced embedding methods complement these advances by introducing intermediate reasoning before producing retrieval representations.

2.2 Reasoning-Enhanced Multimodal Retrieval

Explicit reasoning helps multimodal embedding models interpret complex retrieval queries. Think-Then-Embed (TTE) (Cui et al. (2026)) augments inputs with generated rationales before embedding,

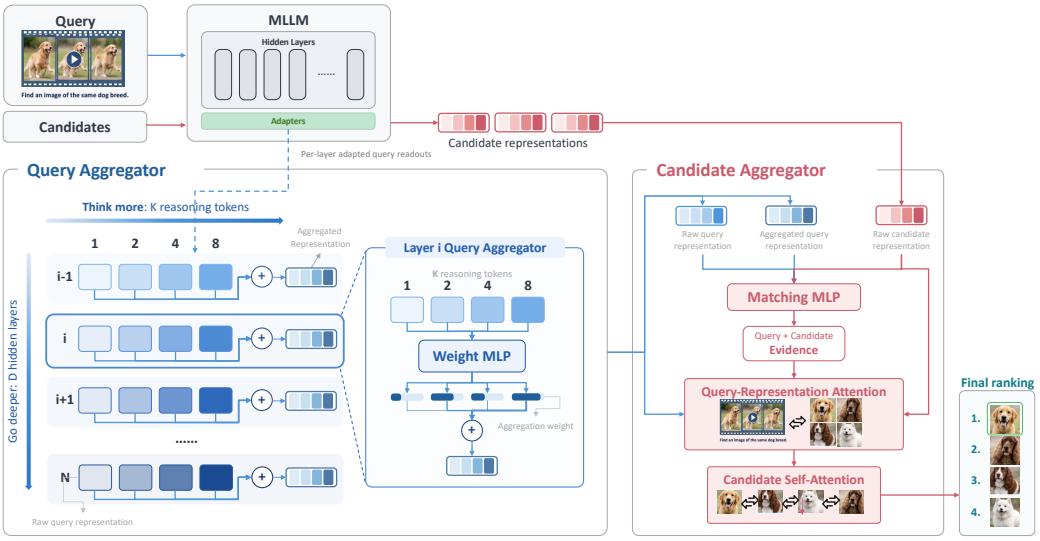

Figure 2: Overview of the proposed framework. We (1) extract multi-exit query representations across reasoning budgets and model depths, (2) adaptively combine them using the *Query Aggregator*, and (3) aggregate query–candidate matching information and compare candidates through two attention modules in the *Candidate Aggregator*. Together, these modules enable the model to adaptively integrate information across reasoning budgets and depths for multimodal retrieval.

while RGE (Liu et al. (2025a)) and UME-R1 (Lan et al. (2026)) integrate reasoning and embedding generation within a single model. Embed-RL (Jiang et al. (2026)) further aligns reasoning with retrieval through embedder-guided reinforcement learning. To avoid unnecessary CoT generation, TRACE (Hao et al. (2026)), MMEmb-R1 (Wang et al. (2026)), and Think When Needed (Zhang et al. (2026)) learn when to invoke reasoning through difficulty-aware training, reinforcement learning, or explicit routing mechanisms. Another line of work reduces the cost of reasoning itself. PLUME (He et al. (2026)) replaces verbal reasoning traces with continuous latent states, LaME (Wu et al. (2026)) learns compact latent reasoning tokens, and DME (Chen et al. (2026)) organizes latent reasoning around retrieval-specific evidence. TTE-Flash (Cheng et al. (2026)) similarly uses latent think tokens and explores adapting the reasoning budget to each input. However, committing to a single reasoning budget may leave useful information from other reasoning levels underexplored. MARS (Jung et al. (2026)) constructs retrieval representations by integrating information from multiple model depths. However, its aggregation weights are shared across inputs, limiting their ability to reflect query-specific differences in the importance of different layers. Despite these advances, how to adaptively use representations across both reasoning budgets and model depths remains underexplored. Addressing this question is important for aligning retrieval representations with the varying reasoning demands of individual queries.

3 METHOD

The DUET framework consists of three stages, namely *Multi-exit Representation Learning*, *Query Aggregation*, and *Candidate Aggregation*. Section 3.1 first introduces the contrastive learning objective and then describes how we obtain retrieval-capable query representations from multiple reasoning budgets and model depths. Sections 3.2 and 3.3 introduce the Query Aggregator and Candidate Aggregator, respectively, detailing how these representations are combined for each query and used to refine candidate rankings. An overview of the framework is shown in Figure 2. At inference time, the Query Aggregator combines embeddings from different reasoning budgets at each model depth to form multiple query representations. These representations are used to retrieve a pool of potentially relevant candidates. The Candidate Aggregator then integrates matching evidence across query representations and models interactions among candidates to produce the final ranking.

3.1 MULTI-EXIT REPRESENTATION LEARNING

We train a multimodal embedding model through contrastive learning to align queries with their relevant candidates in a shared representation space. Given a mini-batch of B query-candidate pairs $\{(q_i, c_i)\}_{i=1}^{B}$, let h_{q_i} and h_{c_i} denote their normalized embeddings. The remaining candidates in the mini-batch serve as in-batch negatives for q_i. We optimize the model with the InfoNCE objective:

$$\mathcal{L}_{\text{NCE}} = -\frac{1}{B}\sum_{i=1}^{B}\log\frac{\exp(\text{sim}(h_{q_i}, h_{c_i})/\tau)}{\sum_{j=1}^{B}\exp(\text{sim}(h_{q_i}, h_{c_j})/\tau)}. \tag{1}$$

where $\text{sim}(\cdot,\cdot)$ denotes cosine similarity and τ is the temperature. This objective establishes the shared embedding space before multi-exit training. We then freeze the backbone and use its normalized candidate embeddings v_c as fixed retrieval targets.

To construct query representations with different reasoning budgets, we append k learnable continuous reasoning tokens, followed by an embedding readout token, to the multimodal query:

$$[\text{prefix}(q), e_1, \ldots, e_k, e_{\text{emb}}]. \tag{2}$$

These continuous reasoning tokens aggregate query information through causal attention to the query prefix and preceding reasoning tokens. Their number k controls the reasoning budget. To encourage the readout token to capture the information aggregated by these tokens, we mask its direct attention to the query prefix. The readout token thus forms the retrieval representation from the query information encoded in the reasoning tokens.

Let $\mathcal{K}$ and $\mathcal{D}$ denote the sets of reasoning budgets and decoder depths, and let $z_{k,d}(q)$ denote the normalized readout representation at budget k and depth d. With the backbone frozen, we optimize the continuous reasoning tokens, readout embedding, and lightweight residual adapters acting on reasoning tokens at each layer. Training combines multi-positive retrieval supervision over query-specific candidate sets, cross-exit consistency, and selective CoT distillation:

$$\mathcal{L}_{\text{exit}} = \mathcal{L}_{\text{full}} + 0.5\,\mathcal{L}_{\text{sub}} + 0.25\,\mathcal{L}_{\text{cons}} + 0.5\,\mathcal{L}_{\text{CoT}}. \tag{3}$$

Here, $\mathcal{L}_{\text{full}}$ supervises the largest-budget, final-depth exit, while $\mathcal{L}_{\text{sub}}$ averages retrieval losses over all exits. $\mathcal{L}_{\text{cons}}$ transfers the full exit's candidate preferences to other exits, and $\mathcal{L}_{\text{CoT}}$ distills representations from a CoT-conditioned teacher when it achieves a higher retrieval margin. The student uses the original query without annotated reasoning text, so explicit CoT supervision is needed only during training. Detailed definitions are provided in Appendix A.3.1.

3.2 QUERY-DEPENDENT REPRESENTATION AGGREGATION

The reasoning budget needed for effective retrieval varies across queries. A larger budget does not necessarily yield a better representation, and representations formed with smaller budgets can still provide useful matching information. Selecting a single exit discards these alternatives, while uniform averaging treats them as equally reliable. We therefore introduce a **Query Aggregator** that learns query-dependent weights to combine budget-specific representations at each selected depth into an aggregated query representation.

For each representation $z_{k,d}(q)$, we extract a compact feature vector through a shared projection:

$$p_{k,d}(q) = \text{SiLU}(W_P\text{LN}(z_{k,d}(q)) + b_P), \tag{4}$$

where $W_P \in \mathbb{R}^{r\times H}$ and $b_P \in \mathbb{R}^r$ are learnable parameters, with $r < H$. This projection compresses the high-dimensional query representations into compact features, reducing the input dimensionality and the parameter cost of the weight prediction network.

To identify the source of each representation, we use learnable budget and depth embeddings e_k^K and e_d^D from two embedding tables trained jointly with the MLP. We also compute the dot product and L2 distance between the current representation $z_{k,d_j}(q)$ and the aggregated query representation from the previous selected depth, $m_{d_{j-1}}(q)$, to characterize their alignment. These comparison features are concatenated with $p_{k,d_j}(q)$, the budget and depth embeddings, and query metadata describing

input lengths and modalities to form $x_{k,d_j}(q)$. A shared lightweight MLP maps each feature vector to a weight logit. We normalize the logits over reasoning budgets at each depth:

$$\alpha_{k,d}(q) = \mathrm{softmax}_k(\mathrm{MLP}_Q(x_{k,d}(q))). \tag{5}$$

The resulting weights combine the original query representations into an aggregated retrieval view:

$$m_d(q) = \mathrm{norm}\left(\sum_{k\in\mathcal{K}} \alpha_{k,d}(q) z_{k,d}(q)\right). \tag{6}$$

The weights depend only on the query and apply to all candidates. During training, we keep the MLLM and residual adapters fixed and optimize only the Query Aggregator.

Each aggregated query representation receives multi-positive contrastive supervision over a training candidate set $\mathcal{B}(q)$, which includes positive candidates and hard negatives as described in Appendix A.3.1. For the query-dependent fused representation $m_d(q)$ at each depth d, we optimize the following multi-positive retrieval objective:

$$\mathcal{L}_{Q,\mathrm{retr}} = -\mathbb{E}_q\left[\frac{1}{|\mathcal{D}|}\sum_{d\in\mathcal{D}} \log \frac{\sum_{c\in\mathcal{P}(q)\cap\mathcal{B}(q)} \exp(\cos(m_d(q), v_c)/\tau)}{\sum_{c\in\mathcal{B}(q)} \exp(\cos(m_d(q), v_c)/\tau)}\right]. \tag{7}$$

Here, $\mathcal{P}(q)$ denotes the set of candidates relevant to q, and τ is the temperature.

We additionally train an auxiliary binary head whose target is positive when retrieval is correct at the current depth or remains incorrect at all subsequent selected depths. The overall objective is

$$\mathcal{L}_Q = \mathcal{L}_{Q,\mathrm{retr}} + \lambda_{\mathrm{aux}}\mathcal{L}_{Q,\mathrm{aux}}, \tag{8}$$

where $\mathcal{L}_{Q,\mathrm{aux}}$ is the binary cross-entropy loss; all selected depths are retained at inference.

The Query Aggregator thus adapts the contribution of each reasoning budget to the current query while preserving retrieval views at different depths. Further details on training and inference are provided in Appendix A.3.2.

3.3 Candidate-Aware Evidence Aggregation

Each query representation retrieves and ranks candidates independently, which can lead to conflicting rankings. We merge the retrieved candidates into a shared pool, but this alone does not resolve these disagreements. We therefore introduce a **Candidate Aggregator** that first combines matching evidence across query representations for each candidate, then compares candidates within the pool to produce the final ranking.

Let $\{u_a(q)\}_{a=1}^{A}$ denote the A query representations, including the raw exit representations, Query Aggregator outputs, and base query representation $z_0(q)$. We retrieve candidates using each representation, then merge the results and remove duplicates to form a shared candidate pool $\mathcal{C}(q)$, which is the union of the top-5 candidates retrieved by each of the 21 query representations and by the normalized average of the base and final-exit representations. For every candidate $c \in \mathcal{C}(q)$, we construct matching evidence against all A query representations.

Specifically, we project the query representation $u_a(q)$ and candidate embedding v_c into a shared low-dimensional space and pass their elementwise product through an MLP to obtain a matching feature vector. This vector captures interactions between their projected features, providing a learned description of query–candidate compatibility. A second MLP encodes the candidate's retrieval score and rank under $u_a(q)$. The score measures the similarity between $u_a(q)$ and v_c, while the rank indicates the candidate's position when candidates are ordered by this similarity score. We add these two feature vectors together with a learned source embedding e_a, which identifies the query representation associated with the evidence. The combined matching vector is

$$t_{c,a} = \mathrm{MLP}_{\mathrm{match}}(P_Q(u_a(q)) \odot P_C(v_c)) + \mathrm{MLP}_{\mathrm{retr}}(r_{c,a}) + e_a. \tag{9}$$

Here, P_Q and P_C are learnable projections, and $\odot$ denotes elementwise multiplication. $r_{c,a}$ contains the retrieval score and rank of candidate c under $u_a(q)$, while e_a is a learnable embedding identifying the query representation.

For each candidate c, *Query-Representation Attention* aggregates its matching vectors $\{t_{c,a}\}_{a=1}^{A}$ into a summary h_c at a learnable `CLS` token. *Candidate Self-Attention* then compares candidate summaries, producing updated representations $\widetilde{h}_c$. We omit candidate positional embeddings so that scores do not depend on input order. The final score combines a base similarity score $a_c = \cos(z_0(q), v_c) + \cos(z_{k_{\max},d_{\max}}(q), v_c)$ with a learned correction:

$$S_c = \beta \left(a_c - \max_{c' \in \mathcal{C}(q)} a_{c'} \right) + w^\top \widetilde{h}_c + b, \qquad c \in \mathcal{C}(q). \tag{10}$$

where β scales the base score, and w, b are learnable scoring parameters.

During training, we freeze all preceding modules. If a retrieved pool contains no relevant candidate, we exclude the query from the Candidate Aggregator training loss without inserting additional positive candidates. We optimize the Candidate Aggregator with a multi-positive listwise loss:

$$\ell_C(q) = -\log \frac{\sum_{c \in \mathcal{P}(q) \cap \mathcal{C}(q)} \exp(S_c)}{\sum_{c \in \mathcal{C}(q)} \exp(S_c)}. \tag{11}$$

At inference, candidates in the retrieved pool are ranked in descending order of their final scores S_c. Further details on training and inference are provided in Appendix A.3.3.

4 Experiments

4.1 Experimental Setup

Implementation Details. We initialize DUET from Qwen3-VL-4B-Instruct and Qwen3-VL-8B-Instruct Bai et al. (2025), using retrieval pairs from 25 image, video, and visual-document datasets from Zhang et al. (2026). We first perform full-parameter contrastive fine-tuning for one epoch without CoT annotations, then freeze the backbone and sequentially train the multi-exit modules with retrieval supervision and selective distillation from CoT-conditioned representations, followed by sequential training of the Query and Candidate Aggregators. For both models, reasoning budgets $\mathcal{K} = \{1, 2, 4, 8\}$ and decoder depths $\mathcal{D} = \{18, 24, 30, 36\}$ yield 16 raw exits. Multi-exit training runs for one epoch with batch size 32 and learning rates of 5×10^{-5} for continuous reasoning tokens and 2.5×10^{-5} for adapters. The Query and Candidate Aggregators are trained for up to eight epochs, with learning rates of 3×10^{-4} and 2×10^{-4} and batch sizes of 64 and 24, respectively. The additional trainable components contain 25.15M parameters for 4B and 40.03M for 8B, each accounting for less than 1% of its backbone. Further training configurations, visual preprocessing, and data details are provided in Appendix A.1.

Benchmark and Baselines. We evaluate on MMEB-V2 (Meng et al. (2025)), which comprises 78 tasks spanning images, videos, and visual documents. We report Hit@1 for image and video tasks and NDCG@5 for visual document retrieval. We compare our results against **direct embedding baselines** , including ColPali (Faysse et al. (2025)), VLM2Vec (Jiang et al. (2024)), GME (Zhang et al. (2024)), LamRA (Liu et al. (2025b)), CAFe (Yu et al. (2025)), and VLM2Vec-V2 (Meng et al. (2025)), as well as **reasoning-based embedding baselines** , including explicit CoT incorporated methods like TTE (Cui et al. (2026)), UME-R1 (Lan et al. (2026)), and TWN (Zhang et al. (2026)), and latent reasoning methods like PLUME (He et al. (2026)) and TTE-Flash (Cheng et al. (2026)). To reduce the influence of training data scale, we omit Qwen3-VL-Embedding (Li et al. (2026)) and DME (Chen et al. (2026)), whose training relies on substantially larger proprietary datasets.

4.2 Main Results

We evaluate DUET-4B and DUET-8B on the MMEB-V2 benchmark across 78 datasets, with results reported in Table 1. DUET-8B achieves the highest overall score of 69.2 among the compared methods, while DUET-4B obtains a competitive score of 67.0. DUET performs particularly well on video tasks: the 4B and 8B variants achieve overall scores of 53.5 and 54.8, respectively, both exceeding the strongest baseline score of 53.0. Detailed per-task performance is reported in Appendix A.4. These results demonstrate the effectiveness of DUET across diverse multimodal retrieval settings. The exit-selection oracle reveals substantial headroom for selecting among raw exits, motivating the exploitation of complementary information across reasoning budgets and depths. We additionally

Table 1: Comparison of performance between baselines and our method on MMEB-V2. CLS: classification, QA: question answering, RET: retrieval, GD: grounding, MRET: moment retrieval, VDR: ViDoRe, VR: VisRAG, OOD: out-of-domain. The highest and second-highest values are highlighted in bold and underline. The exit-selection oracle selects, for each query, the best-performing output among the 16 raw exits, measured by Hit@1 for Image/Video and NDCG@5 for VisDoc.

Model	Image					Video					VisDoc					All
	CLS	QA	RET	GD	Overall	CLS	QA	RET	MRET	Overall	VDRv1	VDRv2	VR	OOD	Overall	
of Datasets	10	10	12	4	36	5	5	5	3	18	10	4	6	4	24	78
Direct Embedding Baseline Models																
ColPali-V1.3-3B	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	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
GME-7B	57.7	34.7	71.2	59.3	56.0	37.4	50.4	28.4	37.0	38.4	**89.4**	55.6	85.0	44.4	75.2	57.8
LamRA-2-7B	59.2	26.5	70.0	62.7	54.1	39.3	42.6	24.3	34.6	35.2	22.0	11.5	37.4	21.0	23.9	40.4
LamRA-2.5-7B	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-2B	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
VLM2Vec-7B	62.7	56.9	69.4	82.2	65.5	39.1	30.0	29.0	38.9	33.7	56.9	9.4	59.1	38.1	46.4	52.3
VLM2Vec-V2	62.9	56.3	69.5	77.3	64.9	39.3	34.3	28.8	36.8	34.6	75.5	45.1	79.6	39.6	65.4	58.0
CAFe-7B	63.6	61.7	69.1	87.6	67.6	35.8	58.7	34.4	39.5	42.4	70.7	49.6	79.5	38.1	63.9	60.6
Reasoning-based Embedding Baseline Models																
UME-R1-2B	64.8	62.8	67.6	77.2	66.6	44.3	51.0	32.9	39.7	42.2	72.4	46.2	79.2	37.2	63.9	60.1
UME-R1-7B	67.1	69.2	71.9	84.9	71.3	48.6	60.7	38.2	39.3	47.5	75.7	50.5	83.7	37.6	67.1	64.5
TTE_s-2B	67.9	66.6	70.2	84.1	70.1	47.3	49.1	33.2	32.1	41.3	77.5	53.2	83.2	41.1	68.8	63.1
TTE_s-7B	69.7	72.4	**74.0**	90.6	**74.2**	49.1	60.6	36.4	37.2	46.8	84.1	**62.7**	**91.9**	47.6	76.4	68.6
Embed-RL-2B	62.8	67.9	68.6	90.4	69.2	57.0	55.9	45.1	49.4	52.1	79.9	52.0	84.6	65.7	74.1	66.8
Embed-RL-4B	63.7	70.5	71.3	91.4	70.1	**57.6**	58.4	45.1	49.5	53.0	80.2	53.4	84.9	67.1	74.7	68.1
PLUME	66.5	59.2	67.6	79.7	66.3	45.0	52.3	33.5	46.7	44.1	72.1	49.8	78.1	57.4	67.5	61.6
TWN-4B	68.6	71.7	68.0	87.0	71.3	45.8	63.4	33.6	36.6	45.7	81.5	53.6	84.8	67.3	75.3	66.6
TWN-8B	**70.2**	**74.3**	70.8	87.1	73.4	50.1	64.0	34.8	40.8	48.2	82.5	56.7	86.2	**70.0**	**77.0**	68.7
TTE-Flash	67.1	61.7	70.7	81.3	68.3	54.4	51.5	45.4	**51.5**	50.6	75.6	52.7	83.8	41.1	68.1	64.1
Ours																
DUET-4B	65.7	65.9	69.4	89.8	69.7	51.6	66.3	45.7	48.6	53.5	78.4	52.1	82.7	67.0	73.2	67.0
DUET-4B (*oracle*)	73.9	71.9	75.7	95.1	76.3	62.5	80.6	56.1	66.5	66.4	84.0	65.6	87.1	75.6	80.3	75.3
DUET-8B	65.8	68.0	71.2	**93.0**	71.2	53.4	**67.9**	**46.3**	49.4	**54.8**	82.5	56.4	85.6	69.8	76.8	**69.2**
DUET-8B (*oracle*)	72.0	75.0	79.6	95.8	77.3	62.3	81.5	56.4	66.6	66.7	86.3	67.3	90.8	77.9	82.9	76.6

quantify DUET's inference overhead relative to the contrastively fine-tuned backbone under matched evaluation settings in Appendix A.2.1.

4.3 Ablation Studies

We conduct additional studies to evaluate the key components and aggregation strategy of DUET through architectural ablations, attention analysis, and interventions on its input representations.

4.3.1 Ablation of the Dual-Aggregator Architecture

We ablate the two aggregators and their input representations. Additional aggregation and routing baselines are reported in Appendix A.2.2. In the full model, the Candidate Aggregator receives 21 query representations, including 16 raw exits from $K \in \{1, 2, 4, 8\}$ and $D \in \{18, 24, 30, 36\}$, one base representation at $K = 0, D = 36$, and four Query Aggregator outputs, one at each selected depth. It combines their query–candidate matching evidence to rank the retrieved candidates. All architecture variants that retain a Candidate Aggregator use the same training data, training protocol, and candidate pools. We organize the variants into three groups according to which part of the framework is modified.

Ablating the Query Aggregator. We retain the Candidate Aggregator and vary only how query representations are constructed, testing the contribution of learned, query-dependent budget aggregation.

1. **w/o Query Aggregator.** We remove the four Query Aggregator outputs and retrain the Candidate Aggregator from scratch on the remaining 17 representations. This tests the contribution of the learned aggregated representations.
2. **Global Weighting.** At each selected depth, we replace query-dependent budget weights with learned weights shared across all queries, retaining four depth-specific aggregated representations.
3. **One-Budget.** At each selected depth, we select the budget-specific representation with the highest Query Aggregator weight, yielding four selected representations in place of the four weighted aggregates.

Table 2: Ablation of DUET's multi-view aggregation and candidate ranking design on MMEB-V2. $\mathrm{Aggr}_{\mathrm{Q}}$ and $\mathrm{Aggr}_{\mathrm{C}}$ denote the Query Aggregator and Candidate Aggregator, respectively.

Variant	Image	Video	VisDoc	Overall
FULL	71.2	54.8	76.8	69.2
Ablating the Query Aggregator				
w/o $\mathrm{Aggr}_{\mathrm{Q}}$	70.7	53.2	74.9	67.9
Global Weighting	70.7	53.0	73.7	67.6
One-Budget	71.0	52.6	75.6	68.2
Ablating the Candidate Aggregator				
w/o $\mathrm{Aggr}_{\mathrm{C}}$	68.5	48.2	71.1	64.6
RRF	57.5	45.4	62.3	56.2
Budget and Depth Diversity				
Endpoint Repr.	70.0	52.7	72.9	66.9
Budget Diversity	70.0	52.8	73.4	67.1
Depth Diversity	70.2	53.0	73.3	67.2

Table 3: Representation ablations of DUET-8B and DUET-4B on MMEB-V2. Each variant removes one group of inputs to the Candidate Aggregator at inference time. *None* represents the full DUET model.

Ablated input	Image	Video	VisDoc	All
DUET-4B				
None	69.7	53.5	73.2	67.0
Shallow	68.1	52.9	70.4	65.3
Middle	68.1	52.9	70.6	65.3
Final	68.1	52.6	70.6	65.3
Aggregated	67.5	52.1	69.9	64.7
DUET-8B				
None	71.2	54.8	76.8	69.2
Shallow	70.7	52.9	73.7	67.5
Middle	70.9	52.9	73.6	67.6
Final	70.7	52.9	73.5	67.4
Aggregated	70.9	52.6	73.3	67.4

Ablating the Candidate Aggregator. We retain all 21 query representations and replace only the candidate-ranking mechanism, testing whether simple score or rank fusion can replace learned candidate-aware ranking.

1. **w/o Candidate Aggregator.** We retain all 21 query representations and rank candidates across the full eligible gallery by summing their cosine similarities with equal weights.
2. **RRF.** We rerank the shared candidate pool using Reciprocal Rank Fusion (Cormack et al. (2009)) over the full-gallery rankings from all 21 query representations, with a fusion constant of 60.

Budget and Depth Diversity. We retain the base representation, omit Query Aggregator outputs, and retrain the Candidate Aggregator with the same training protocol and fixed candidate pools. All representation-dependent evidence features are recomputed using only the retained representations, while the candidate pool and base similarity score are held fixed across variants:

1. **Endpoint Representations.** We use only the base representation $z_0(q)$ and the largest-budget, final-depth representation $z_{8,36}(q)$, testing whether these two endpoints suffice for candidate-aware ranking.
2. **Budget Diversity.** We use $z_0(q)$ and $\{z_{k,36}(q)\}_{k\in\mathcal{K}}$, retaining all four reasoning budgets at the final depth to examine budget diversity without intermediate-layer representations.
3. **Depth Diversity.** We use $z_0(q)$ and $\{z_{8,d}(q)\}_{d\in\mathcal{D}}$, retaining all four depths at the largest reasoning budget to examine depth diversity without smaller-budget representations.

Table 2 reports DUET-8B ablations on MMEB-V2, with corresponding DUET-4B results in Table 7 showing consistent overall trends. Removing the Query Aggregator or replacing its query-dependent fusion with shared weights or single-budget selection lowers overall performance, supporting adaptive budget aggregation. Replacing the Candidate Aggregator with score summation or RRF causes larger overall drops, highlighting the value of learned candidate-aware ranking. Under the same Candidate Aggregator architecture, Budget Diversity and Depth Diversity both improve over Endpoint Representations. Retaining both dimensions in the w/o Query Aggregator variant yields further gains, supporting the complementary value of representations across budgets and depths. The full model further improves performance by incorporating learned query aggregates and achieves the highest scores across all three modalities at 8B.

4.3.2 Ablation of the Multi-Exit Representation

We analyze DUET's attention allocation across query representations, then assess its dependence on these inputs through ablation. Figure 3 shows attention weights for top-ranked candidates in DUET-8B; Figure 4 presents the DUET-4B results. Base denotes the original embedding ($K = 0, D = 36$);

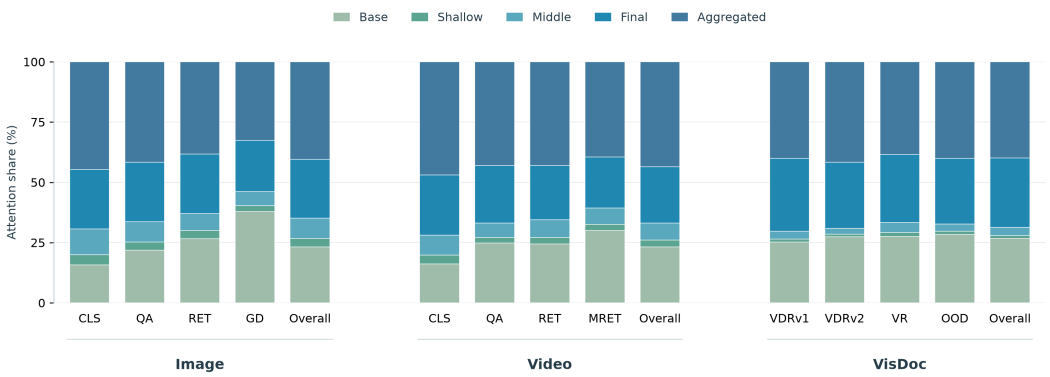

Figure 3: Query-view attention distributions in the Candidate Aggregator across MMEB-V2 tasks for DUET-8B. Colors indicate query representation groups.

Shallow groups raw representations at depths 18 and 24, Middle at depth 30, and Final at depth 36, each across all four budgets; Aggregated comprises the four Query Aggregator outputs. We renormalize the weights and average them equally across datasets within each task. They reflect attention allocation, not independent causal contributions. In DUET-8B, grounding and moment retrieval emphasize the base representation, suggesting reliance on localization cues already captured by the backbone. Aggregated representations receive substantial attention across modalities and model sizes, indicating their use alongside raw representations. Appendix A.2.3 reports attention weights for individual exits.

We mask each representation group at the Candidate Aggregator's input in DUET-4B and DUET-8B without retraining, keeping the candidate pool and remaining representations fixed. Query Aggregator outputs are not recomputed. Unlike retrained architecture variants, these interventions assess the trained model's dependence on direct inputs. Table 3 shows that masking any group reduces performance across all modalities and both model sizes. Removing aggregated representations causes the largest overall drop for DUET-4B and a joint-largest drop for DUET-8B. These results support the complementary value of raw exits and learned aggregates for ranking.

5 Discussion

Several limitations suggest directions for future research. First, DUET uses a fixed set of reasoning budgets and decoder depths. Extending the framework to a denser budget–depth grid, including all decoder layers, could uncover additional useful representations. Second, a substantial gap remains to the exit-selection oracle, which selects the best-performing raw exit for each query. Finally, the two aggregators are currently trained in separate stages. Future work could explore joint optimization, allowing candidate-level ranking supervision to directly guide query-dependent representation aggregation and improve coordination between the two modules.

6 Conclusion

In this work, we introduce DUET, a dual-aggregator framework that combines query representations formed with different reasoning-token budgets and at different decoder depths for multimodal retrieval. Its Query Aggregator learns query-dependent combinations, while its Candidate Aggregator integrates matching evidence for final ranking. Experiments on MMEB-V2 demonstrate the effectiveness of this design across both model scales, outperforming averaging, merging, and routing alternatives overall with less than 1% of additional parameters. Controlled ablations support the complementary value of budget and depth diversity and the contributions of both aggregators. These findings highlight the value of jointly exploiting representations across the reasoning process, showing how complementary information from multiple stages can contribute to retrieval beyond the final representation alone.

AI USE STATEMENT

We used large language models (LLMs) solely to improve the clarity and fluency of the manuscript. They were not involved in formulating the research motivation, designing the framework, or implementing the method.

IMPACT STATEMENT

This work aims to improve universal multimodal embeddings through adaptive aggregation of representations and matching evidence. The proposed framework may benefit multimodal search and information access. We do not anticipate societal consequences specific to this approach beyond established considerations for responsible deployment of multimodal models, including data privacy and biases inherited from training data.

ETHICS STATEMENT

Our code and methodology are intended to improve universal multimodal embeddings and are not designed to facilitate discrimination, privacy violations, or other harmful uses. Nevertheless, we acknowledge that the pretrained MLLMs and multimodal datasets used in this work may contain inherent biases, which could propagate to the learned representations and affect candidate rankings in downstream applications.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide comprehensive details regarding our proposed framework in both the main text and the appendix.

Methodology. A thorough description of our method is presented in Section 3, including the multi-exit representation learning (Section 3.1), query aggregator design (Section 3.2), candidate aggregator design (Section 3.3), and training details (Appendix A.1). We provide figures (Figure 1 and Figure 2) to enhance readability.

Datasets. We employ open-source datasets (Zhang et al. (2026)) for training and utilize open-source benchmarks for evaluation (Meng et al. (2025)). All datasets and benchmarks referenced in the paper are publicly accessible and can be downloaded from the Hugging Face platform [1].

Experiment Details. Detailed information on method implementation is provided in Section 4.1 and Appendix A.1. This includes computation details, hyper-parameter configurations, and multimodal embedding evaluation configurations.

Additionally, we provide the training and evaluation code in the supplementary materials. Should the paper be accepted, we commit to making the full source code for our approach publicly available.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Haonan Chen, Chu Li, Zhicheng Wang, Yuanwei Liu, Yuanjiang Wang, Shaohua Jiang, and Zhicheng Dou. Douyin multimodal embedding model technical report. *arXiv preprint arXiv:2608.02148*, 2026.

Jianpeng Cheng, Xian Wu, Jiangfan Zhang, Wentao Bao, Chaitanya Ahuja, Shlok Kumar Mishra, Hanchao Yu, Yang Gao, Fan Xia, Qi Guo, et al. Tte-flash: Accelerating reasoning-based multimodal representations via think-then-embed tokens. *arXiv preprint arXiv:2605.16638*, 2026.

[1] https://huggingface.co/datasets

Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pp. 758–759, 2009.

Xuanming Cui, Jianpeng Cheng, Hong-you Chen, Satya Narayan Shukla, Abhijeet Awasthi, Xichen Pan, Chaitanya Ahuja, Shlok Mishra, Taipeng Tian, Qi Guo, et al. Think then embed: Generative context improves multimodal embedding. In *International Conference on Learning Representations*, volume 2026, pp. 2690–2709, 2026.

Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. Colpali: Efficient document retrieval with vision language models. In *International Conference on Learning Representations*, volume 2025, pp. 61424–61449, 2025.

Tiancheng Gu, Kaicheng Yang, Ziyong Feng, Xingjun Wang, Yanzhao Zhang, Dingkun Long, Yingda Chen, Weidong Cai, and Jiankang Deng. Breaking the modality barrier: Universal embedding learning with multimodal llms. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 2860–2869, 2025.

Tiancheng Gu, Kaicheng Yang, Kaichen Zhang, Xiang An, Ziyong Feng, Yueyi Zhang, Weidong Cai, Jiankang Deng, and Lidong Bing. Unime-v2: Mllm-as-a-judge for universal multimodal embedding learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 21378–21386, 2026.

Xiangzhao Hao, Shijie Wang, Tianyu Yang, Tianyue Wang, Haiyun Guo, and Jinqiao Wang. Trace: Task-adaptive reasoning and representation learning for universal multimodal retrieval. *arXiv preprint arXiv:2603.02929*, 2026.

Chenwei He, Xiangzhao Hao, Tianyu Yang, Yuxiang Ma, Yuheng Jia, Lingxiang Wu, Chaoyang Zhao, Haiyun Guo, and Jinqiao Wang. Plume: Latent reasoning based universal multimodal embedding. *arXiv preprint arXiv:2604.02073*, 2026.

Haonan Jiang, Yuji Wang, Yongjie Zhu, Xin Lu, Wenyu Qin, Meng Wang, Pengfei Wan, and Yansong Tang. Embed-rl: Reinforcement learning for reasoning-driven multimodal embeddings. In *European Conference on Computer Vision*, pp. 39–56. Springer, 2026.

Ziyan Jiang, Rui Meng, Xinyi Yang, Semih Yavuz, Yingbo Zhou, and Wenhu Chen. Vlm2vec: Training vision-language models for massive multimodal embedding tasks. *arXiv preprint arXiv:2410.05160*, 2024.

Uicheol Jung, Juyoung Hong, Geuntaek Lim, and Yukyung Choi. Mars: What retrieval signals are hidden in multimodal large language models for text-video retrieval? *arXiv preprint arXiv:2609.02565*, 2026.

Zhibin Lan, Liqiang Niu, Fandong Meng, Jie Zhou, and Jinsong Su. Llave: Large language and vision embedding models with hardness-weighted contrastive learning. In *EMNLP (Findings)*, pp. 13721–13735, 2025.

Zhibin Lan, Liqiang Niu, Fandong Meng, Jie Zhou, and Jinsong Su. Ume-r1: Exploring reasoning-driven generative multimodal embeddings. In *International Conference on Learning Representations*, volume 2026, pp. 115779–115806, 2026.

Mingxin Li, Yanzhao Zhang, Dingkun Long, Keqin Chen, Sibo Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, et al. Qwen3-vl-embedding and qwen3-vl-reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. *arXiv preprint arXiv: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*, volume 2025, pp. 44215–44234, 2025.

Chunxu Liu, Jiyuan Yang, Ruopeng Gao, Yuhan Zhu, Feng Zhu, Rui Zhao, and Limin Wang. Reasoning guided embeddings: Leveraging mllm reasoning for improved multimodal retrieval. *arXiv preprint arXiv:2511.16150*, 2025a.

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

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

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Mingxu Tao, Quzhe Huang, Kun Xu, Liwei Chen, Yansong Feng, and Dongyan Zhao. Probing multimodal large language models for global and local semantic representations. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 13050–13056, 2024.

Yuchi Wang, Dingkang Yang, Haiyang Yu, Weikang Bian, Jiefeng Long, Xiao Liang, Chao Feng, and Hongsheng Li. Mmemb-r1: Reasoning-enhanced multimodal embedding with pair-aware selection and adaptive control. *arXiv preprint arXiv:2604.06156*, 2026.

Peixi Wu, Biao Yang, Feipeng Ma, Bosong Chai, Bo Lin, Wei Yuan, Fan Yang, Tingting Gao, Hebei Li, and Xiaoyan Sun. Lame: Learning to think in latent space for multimodal embedding via information bottleneck. *arXiv preprint arXiv:2606.13061*, 2026.

Hao Yu, Zhuokai Zhao, Shen Yan, Lukasz Korycki, Jianyu Wang, Baosheng He, Jiayi Liu, Lizhu Zhang, Xiangjun Fan, and Hanchao Yu. Cafe: Unifying representation and generation with contrastive-autoregressive finetuning. In *2025 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 6345–6356. IEEE, 2025.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11941–11952. IEEE, 2023.

Longxiang Zhang, Weilong Dai, Guanghao Zhang, Hao Jiang, and Pipei Huang. Think when needed: Adaptive reasoning-driven multimodal embeddings with a dual-lora architecture. *arXiv preprint arXiv:2605.14448*, 2026.

Xin Zhang, Yanzhao Zhang, Wen Xie, Mingxin Li, Ziqi Dai, Dingkun Long, Pengjun Xie, Meishan Zhang, Wenjie Li, and Min Zhang. Gme: Improving universal multimodal retrieval by multimodal llms. *arXiv preprint arXiv:2412.16855*, 2024.

Junjie Zhou, Yongping Xiong, Zheng Liu, Ze Liu, Shitao Xiao, Yueze Wang, Bo Zhao, Chen Jason Zhang, and Defu Lian. MegaPairs: Massive data synthesis for universal multimodal retrieval. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19076–19095, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.935. URL https://aclanthology.org/2025.acl-long.935/.

A Appendix

A.1 Implementation Details

A.1.1 Multimodal Vision Processing

We resize images while preserving their aspect ratios and aligning both spatial dimensions to multiples of 32. The pixel budget ranges from 131,072 to 786,432, corresponding to approximately 128–768 visual tokens per image. Spatial patches of 16×16 pixels are merged in 2×2 groups, so each visual token represents a 32×32 region of the resized image. For video evaluation, we retain up to 64 frames for each video, uniformly subsampling longer sequences at evenly spaced indices, and set the FPS parameter to 2.0. Each frame uses a minimum pixel budget of 131,072 and a nominal maximum of 307,200, with the maximum adjusted according to the frame count and a nominal video budget of 2,457,600 pixels. For 64-frame inputs, the minimum-resolution safeguard overrides the budget-derived limit and sets the per-frame ceiling to approximately 137,625 pixels. Under these settings, merging adjacent frame pairs temporally yields approximately 4,300 visual tokens per video, with the exact count depending on the resized dimensions. Query and candidate sequences share a maximum length of 8,192 tokens, consistent with training.

A.1.2 Training Configurations for All Stages

We initialize DUET-8B and DUET-4B from Qwen3-VL-8B-Instruct and Qwen3-VL-4B-Instruct, respectively, and perform first-stage contrastive training for one epoch. Both models are trained with DeepSpeed ZeRO-3 and bf16 precision on a single node with eight PPU-ZW810E accelerators, each with 96 GB of memory. The effective batch size is 128, using eight examples per device and two gradient accumulation steps. We set the initial learning rate to 1×10^{-6}, apply cosine decay with a 3% warmup ratio, and use a weight decay of 0.01. Both aggregators use AdamW, dropout of 0.1, gradient clipping at 1.0, and a maximum of eight training epochs. The Query Aggregator uses a learning rate of 3×10^{-4}, weight decay of 10^{-3}, dropout of 0.1, and batch size of 64. The Candidate Aggregator uses two attention modules with a model dimension of 128, four attention heads, and a feed-forward hidden dimension of 256. It is trained with a learning rate of 2×10^{-4}, weight decay of 10^{-2}, dropout of 0.1, and batch size of 24.

A.1.3 Additional Training Parameters

We first perform full-parameter contrastive fine-tuning to establish a shared query-candidate embedding space. In the subsequent stages, we freeze the backbone, including the vision encoder, and train the continuous reasoning tokens, readout embedding, residual adapters, Query Aggregator, and Candidate Aggregator in their respective stages. Table 4 reports the parameter counts of these components for DUET-4B and DUET-8B. Together, they account for less than 1% of the parameters in their respective backbones, allowing DUET to learn multi-exit representations and adaptive aggregation through lightweight adaptation of the original model.

A.1.4 Training Data Composition

We adopt the multimodal CoT training data from TWN Zhang et al. (2026), covering image, video, and visual document tasks. The image portion comprises 20 training tasks from the MMEB training splits, spanning classification (ImageNet-1K, N24News, HatefulMemes, VOC2007, SUN397), visual question answering (OK-VQA, A-OKVQA, DocVQA, InfographicsVQA, ChartQA, Visual7W), retrieval (VisDial, CIRR, VisualNews, MSCOCO, NIGHTS, WebQA), and visual grounding (RefCOCO). VisualNews and MSCOCO each contribute both text-to-image and image-to-text retrieval tasks. The video portion uses LLaVA-Hound instruction data for video question answering and bidirectional retrieval between videos and captions. Visual document retrieval data comes from ViDoRe, VisRAG in-domain, and VisRAG synthetic datasets. Together, these sources comprise approximately 1.67M query–target pairs, with the task-specific composition detailed in Table 5.

TWN annotates both sides of each training pair using Qwen3.5-35B-A3B through a two-step process that generates reasoning followed by a concise retrieval-oriented summary. Prompts are tailored to the task and modality, allowing brief or empty reasoning for simple inputs. The same model evaluates annotation quality using task-specific criteria. Classification and question-answering annotations

Table 4: Additional trainable parameter counts for DUET-4B and DUET-8B after full-parameter contrastive fine-tuning. M denotes millions of parameters. The backbone remains frozen during these subsequent training stages.

Component	DUET-4B	DUET-8B
Readout adapters	23.80M	38.08M
Query Aggregator	0.37M	0.57M
Candidate Aggregator	0.97M	1.37M
Overall	25.15M	40.03M

Table 5: Training data composition with CoT annotations. Modality indicates the input modality pattern (T: text, I: image, V: video, D: document).

Dataset	**#Pairs**	**#Valid query CoTs**	**#Valid candidate CoTs**	**Modality**
Image-based (MMEB-train, 20 datasets)				
ImageNet-1K	99,984	90,314	18	$T+I \rightarrow T$
N24News	48,979	48,959	0	$T+I \rightarrow T$
HatefulMemes	8,500	8,151	0	$T+I \rightarrow T$
VOC2007	7,844	3,815	0	$T+I \rightarrow T$
SUN397	19,835	18,267	0	$T+I \rightarrow T$
OK-VQA	8,988	8,789	0	$T+I \rightarrow T$
A-OKVQA	17,016	16,800	0	$T+I \rightarrow T$
DocVQA	39,198	39,186	53	$T+I \rightarrow T$
InfographicsVQA	23,732	23,724	5	$T+I \rightarrow T$
ChartQA	28,155	28,146	1	$T+I \rightarrow T$
Visual7W	69,765	63,287	2	$T+I \rightarrow T$
VisDial	123,113	102,078	113,997	$T \rightarrow T+I$
CIRR	26,107	24,814	15,358	$T+I \rightarrow T+I$
VisualNews-t2i	99,847	71,281	89,059	$T \rightarrow T+I$
VisualNews-i2t	99,913	83,949	18,194	$T+I \rightarrow T$
MSCOCO-t2i	99,976	6,297	76,928	$T \rightarrow T+I$
MSCOCO-i2t	112,831	68,225	202	$T+I \rightarrow T$
NIGHTS	15,940	12,294	7,715	$T+I \rightarrow T+I$
WebQA	17,146	13,719	15,189	$T \rightarrow T+I$
MSCOCO	99,348	94,094	55,042	$T+I \rightarrow T+I$
Video-based (LLaVA-Hound)				
LLaVA-Hound-caption	90,132	88,735	85,873	$V \rightarrow T$
LLaVA-Hound-VQA	196,124	191,531	19,647	$V+T \rightarrow T$
Document-based (ViDoRe and VisRAG)				
ViDoRe	116,626	115,022	18,195	$T \rightarrow D$
VisRAG-InDomain	122,414	115,536	122,188	$T \rightarrow D$
VisRAG-Synthetic	83,464	78,463	82,903	$T \rightarrow D$
Image	1,066,217	826,189	391,763	—
Video	286,256	280,266	105,520	—
Document	322,504	309,021	223,286	—
Total	**1,674,977**	**1,415,476**	**720,569**	—

undergo reasoning checks and answer verification, while retrieval and grounding annotations are checked for hallucinations and relevance without requiring an exact answer. Validation is skipped for simple target-side inputs, and empty reasoning traces are treated as clean. A pair is marked clean when all evaluated sides pass validation, yielding an overall clean rate of 89.7%.

During *Contrastive Initialization*, we use all training pairs without incorporating their CoT annotations. For *Multi-exit Representation Learning*, we use the full CoT-annotated dataset to learn query representations across reasoning budgets and model depths. This training schedule first establishes a shared retrieval foundation and then incorporates reasoning supervision to support retrieval at multiple exits.

Table 6: End-to-end evaluation time on all MMEB-V2 datasets with initially empty feature caches, including encoding and scoring. Plain denotes the contrastively fine-tuned backbone, and Full denotes the complete DUET pipeline. Times are reported in minutes.

Model	**Modality**	**Datasets**	**Queries**	**Plain** (min)	**Full** (min)	**Full/Plain**
DUET-4B	Image	36	36,000	22.8	51.9	2.28×
	Video	18	30,523	129.3	376.0	2.91×
	VisDoc	24	12,590	18.8	38.4	2.04×
DUET-8B	Image	36	36,000	26.7	79.4	2.98×
	Video	18	30,523	133.1	420.7	3.16×
	VisDoc	24	12,590	25.1	40.2	1.60×

A.2 Additional Studies

A.2.1 Inference Cost Analysis

We measure the inference cost of DUET-4B and DUET-8B relative to their contrastively fine-tuned backbones, denoted as Plain. Plain uses a single query representation for direct retrieval, while Full executes the complete DUET pipeline, including multi-exit representation extraction, query aggregation, candidate retrieval, and candidate aggregation.

Measurement setup. We evaluate all 78 datasets in MMEB-V2 on a dedicated node with eight PPU-ZW810E accelerators. Full and Plain use the same evaluation queries, candidate galleries, preprocessing, and encoding batch sizes: 32 for image and visual-document tasks and 8 for video tasks. We report end-to-end wall-clock time with feature caches initially empty, including encoding and scoring. Scoring is performed serially on rank 0, and accelerator operations are synchronized at stage boundaries for timing. These measurements characterize evaluation runtime under the stated setup rather than single-query serving latency. DUET shares visual encoding across budgets but runs the decoder separately for each of the four reasoning budgets and the base representation, extracting all four depth-specific representations within each budget pass. Times are summed over all datasets within each modality, and Full/Plain ratios are computed from the unrounded totals.

Results. Table 6 reports the end-to-end evaluation cost across all three modalities. Full requires 2.04–2.91× the runtime of Plain at 4B and 1.60–3.16× at 8B. DUET uses this additional computation to extract and integrate complementary representations across reasoning budgets and decoder depths, while sharing visual encoding across budgets and collecting intermediate-depth representations within each decoder pass. PLUME (He et al. (2026)) likewise incurs additional inference cost, reporting approximately 1.9× latency over its direct-embedding baseline. While the different timing protocols prevent a direct runtime comparison, DUET achieves stronger overall retrieval performance than PLUME and higher overall scores than the alternative aggregation strategies at both model sizes. These gains make the additional computation an acceptable trade-off for quality-focused retrieval.

A.2.2 Additional Architecture Ablations

We extend the main-text ablations to DUET-4B and compare both model sizes with alternative aggregation and routing strategies.

DUET-4B component and representation ablations. We repeat the main-text ablations at 4B using the same variant definitions and evaluation protocol. Table 7 reports the results. Removing the Query Aggregator or replacing its learned weights with global weighting or single-budget selection lowers overall performance. Replacing the Candidate Aggregator with similarity summation or RRF causes larger declines. Budget Diversity and Depth Diversity both improve over Endpoint Representations, while retaining both dimensions in the w/o Query Aggregator variant yields further gains. Full DUET achieves the highest overall score, demonstrating the effectiveness of combining adaptive query aggregation with candidate-aware ranking.

Alternative aggregation and routing strategies. We further compare DUET with simpler ways of selecting and combining its query representations at both model sizes. These comparisons complement the controlled budget-depth ablations in Table 2. Endpoint Averaging, Mean-16, Router,

Table 7: Ablation of DUET-4B's multi-view aggregation and candidate ranking design on MMEB-V2. $\mathrm{Aggr}_{\mathrm{Q}}$ and $\mathrm{Aggr}_{\mathrm{C}}$ denote the Query Aggregator and Candidate Aggregator, respectively.

Variant	Image	Video	VisDoc	Overall
FULL	69.7	53.5	73.2	67.0
Ablating the Query Aggregator				
w/o $\mathrm{Aggr}_{\mathrm{Q}}$	68.3	52.7	71.6	65.7
Global Weighting	67.5	52.8	70.2	64.9
One-Budget	68.1	53.9	66.0	64.2
Ablating the Candidate Aggregator				
w/o $\mathrm{Aggr}_{\mathrm{C}}$	65.5	49.8	70.2	63.3
RRF	56.7	43.6	61.7	55.2
Budget and Depth Diversity				
Endpoint Repr.	67.9	52.6	70.3	65.1
Budget Diversity	68.2	52.9	70.5	65.4
Depth Diversity	68.0	52.9	70.6	65.3

and ARM produce a single embedding for direct retrieval, whereas Uniform Query retains the Candidate Aggregator. We consider the following variants:

1. **Endpoint Averaging.** We average and normalize the base representation $z_0(q)$ and the largest-budget, final-depth representation $z_{8,36}(q)$ for direct retrieval. Unlike Endpoint Representation in Table 2 and Table 7, this variant does not use the Candidate Aggregator.
2. **Uniform Averaging (Mean-16).** We uniformly average and normalize all 16 raw exit representations into a single query embedding. The base representation and Query Aggregator outputs are excluded.
3. **Router.** An MLP, matched to DUET's total trainable parameter count, takes the base query representation and selects a single query representation for direct retrieval, following the design of Zhang et al. (2026), without subsequent candidate aggregation.
4. **Adaptive Representation Merging (ARM).** An MLP, matched to DUET's total trainable parameter count, predicts query-dependent softmax weights over the query representations. Their weighted combination is normalized and used for direct retrieval, without subsequent candidate aggregation.
5. **Uniform Query.** At each selected depth, we replace the learned budget weights with equal weights and normalize the resulting average. We retain the 16 raw exits and the base representation, and retrain the Candidate Aggregator using these representations and the four uniform aggregates, following the same training protocol and retaining the same candidate pools.

As shown in Table 8, DUET achieves higher overall scores than all compared alternatives at both model sizes. Endpoint Averaging and Mean-16 indicate that uniform fusion alone does not fully exploit the available representations, while Router and ARM show that selecting or merging them into a single embedding remains less effective than the full framework. Uniform Query also underperforms despite retaining the Candidate Aggregator, supporting the benefit of learned, query-dependent budget weighting. Together, these results support combining adaptive query aggregation with candidate-aware evidence integration for multimodal retrieval.

A.2.3 Exit Attention Analysis

Figure 4 shows DUET-4B's attention allocation across query representation groups. Grounding and moment retrieval place greater emphasis on the base representation than other tasks within their respective modalities, suggesting greater reliance on localization cues already captured by the backbone. Compared with DUET-8B, DUET-4B assigns more attention to earlier raw representations, while aggregated representations receive substantial attention across modalities. These patterns

Table 8: Comparison of alternative aggregation and routing strategies on MMEB-V2. We report Hit@1 for Image and Video and NDCG@5 for VisDoc.

Method	Image	Video	VisDoc	All
DUET-4B				
Full	69.7	53.5	73.2	67.0
Endpoint Averaging	67.1	51.8	69.5	64.3
Uniform Averaging (Mean-16)	63.7	48.7	69.1	61.9
Router	66.7	52.8	71.5	65.0
ARM	66.8	53.8	71.5	65.3
Uniform Query	67.7	51.9	71.4	65.2
DUET-8B				
Full	71.2	54.8	76.8	69.2
Endpoint Averaging	69.1	51.9	72.6	66.2
Uniform Averaging (Mean-16)	66.3	47.3	71.5	63.5
Router	68.5	52.3	74.8	66.7
ARM	69.0	52.5	75.2	67.1
Uniform Query	70.7	52.8	73.5	67.5

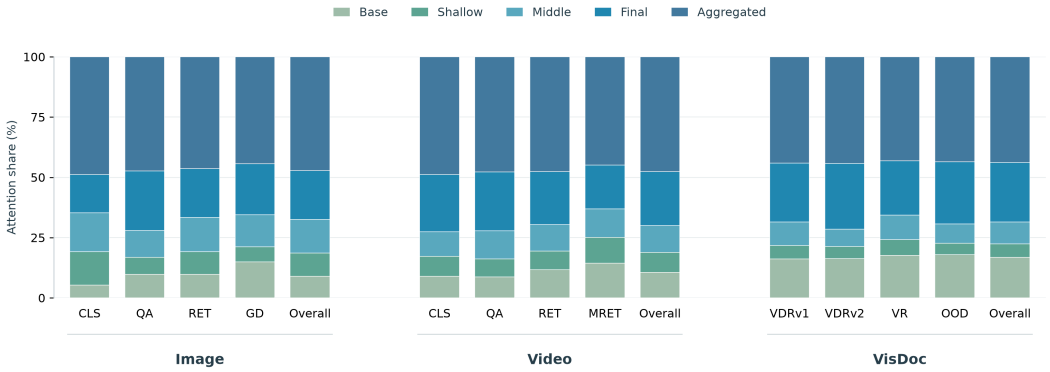

Figure 4: Query-view attention distributions in the Candidate Aggregator across MMEB-V2 tasks for DUET-4B. Colors indicate query representation groups.

suggest that the smaller model draws on both intermediate representations and learned combinations, supporting DUET's use of information from multiple reasoning stages.

We further visualize the Candidate Aggregator's mean attention to query representations for top-ranked candidates across reasoning budgets, depths, and modalities in Figure 5. DUET-8B assigns more attention to final-depth aggregated representations than DUET-4B (28.6–35.6% versus 23.4–27.6%), while DUET-4B places greater emphasis on intermediate-depth aggregated representations. Budget preferences also differ across model scales. DUET-8B's preference for $K = 4$, together with its greater attention to final-depth representations, suggests that useful reasoning information may emerge with fewer tokens and be further refined through deeper processing. The preferred balance between reasoning budget and decoder depth therefore appears to vary with model scale. These findings reinforce DUET's central premise that combining representations from different reasoning stages enables more effective use of multimodal reasoning for candidate ranking.

A.2.4 Exit Accuracy on MMEB-V2

We evaluate the 16 raw exits and four aggregated exits of DUET-4B and DUET-8B, averaging scores over MMEB-V2 datasets within each modality. The results are shown in Figure 6. Shallow raw exits generally perform worse, but the Query Aggregator improves their performance by combining representations across reasoning budgets. At greater depths, raw exits can outperform their aggregated counterparts. The gains at shallow depths demonstrate the value of query aggregation, while the strong performance of deeper raw exits motivates retaining both types of representations. Together,

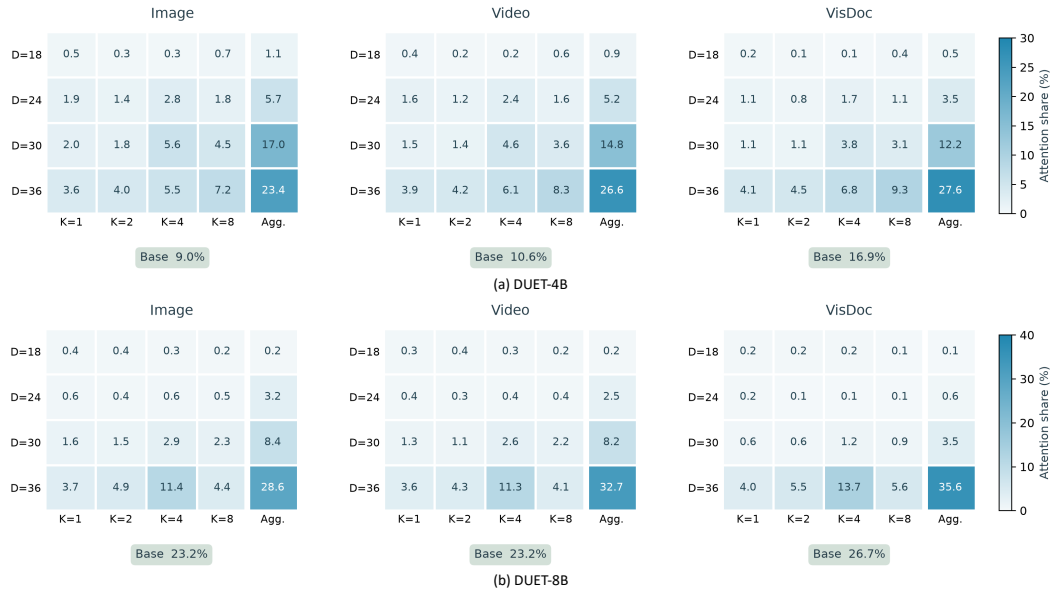

Figure 5: Exit-wise attention in the Candidate Aggregator across model scales and modalities. Agg. denotes Query Aggregator outputs; base attention is reported separately.

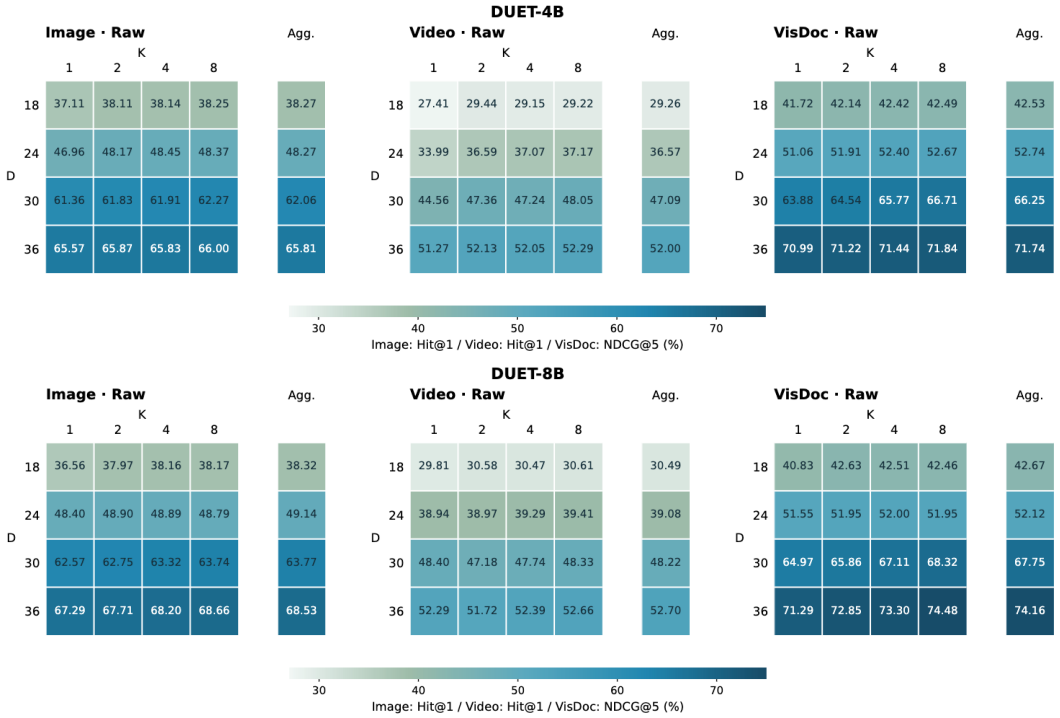

Figure 6: Exit-wise performance of DUET-4B and DUET-8B on MMEB-V2. We evaluate raw and aggregated exits using Hit@1 for Image and Video and NDCG@5 for VisDoc, averaging scores over datasets within each modality.

they provide diverse matching evidence for the Candidate Aggregator to exploit in final ranking. Across both model sizes, the full DUET model outperforms every individual exit, showing that standalone exit performance does not fully capture the value of these representations when used

together. These results support the effectiveness of dual aggregation in translating evidence across reasoning budgets and model depths into stronger overall retrieval performance.

A.3 Detailed Formulation and Optimization of the Dual Aggregators

We detail the computation and optimization of multi-exit representation learning and the two aggregators. We first present the multi-exit training objectives, then derive how retrieval supervision learns the Query Aggregator's budget weights and how ranking supervision trains the Candidate Aggregator. Training proceeds sequentially, with the MLLM backbone fixed throughout and the parameters learned in each stage kept fixed during subsequent stages. We follow the notation introduced in the main text. For compactness, we write $z_{k,d_j}(q_i)$ and $m_{d_j}(q_i)$ as $z_{i,k,j}$ and $m_{i,j}$, respectively, where d_j is the j-th selected decoder depth in ascending order.

A.3.1 Multi-Exit Representation Learning

Starting from the contrastive model, we keep the backbone fixed and optimize the continuous reasoning tokens, residual adapters, and readout embedding. The adapters use a bottleneck dimension of 128, and the trainable readout embedding is initialized from the <emb> embedding learned during contrastive initialization. We train all 16 exits in $\mathcal{E} = \mathcal{K} \times \mathcal{D}$, with $\mathcal{K} = \{1, 2, 4, 8\}$ and $\mathcal{D} = \{18, 24, 30, 36\}$. The exit with the largest budget and deepest selected layer, denoted by $e_F = (8, 36)$, additionally provides supervision to other exits.

For each query q_i, we construct a training candidate set $\mathcal{B}_i$ containing at most 32 candidates. Single-positive examples use one relevant candidate and 31 hard negatives. We denote the positive and negative subsets by $\mathcal{B}_i^+ = \mathcal{B}_i \cap \mathcal{P}_i$ and $\mathcal{B}_i^- = \mathcal{B}_i \setminus \mathcal{P}_i$, respectively. At exit $e = (k, d)$, the similarity between the query and candidate c is

$$s_{i,c}^e = z_{k,d}(q_i)^\top v_c. \tag{12}$$

Each exit is trained to increase the total probability assigned to relevant candidates:

$$\ell_{\mathrm{NCE}}^{(i,e)} = -\log \frac{\sum_{c \in \mathcal{B}_i^+} \exp(s_{i,c}^e/\tau)}{\sum_{c \in \mathcal{B}_i} \exp(s_{i,c}^e/\tau)}, \qquad \tau = 0.02. \tag{13}$$

We apply a dedicated retrieval loss to e_F and average the same objective over all exits:

$$\begin{aligned} \mathcal{L}_{\mathrm{full}} &= \frac{1}{B} \sum_{i=1}^{B} \ell_{\mathrm{NCE}}^{(i,e_F)}, \\ \mathcal{L}_{\mathrm{sub}} &= \frac{1}{B|\mathcal{E}|} \sum_{i=1}^{B} \sum_{e \in \mathcal{E}} \ell_{\mathrm{NCE}}^{(i,e)}. \end{aligned} \tag{14}$$

Under the all-exit configuration, $\mathcal{L}_{\mathrm{sub}}$ includes all 16 exits, including e_F. Beyond direct relevance supervision, we encourage other exits to reproduce the reference exit's relative preferences over candidates. Let p_i^F and p_i^e denote their softmax distributions over the same training candidate set. We minimize

$$\mathcal{L}_{\mathrm{cons}} = \mathbb{E}_{i,e}\left[D_{\mathrm{KL}}\left(\mathrm{sg}(p_i^F) \,\|\, p_i^e \right) \right], \tag{15}$$

where the expectation averages over training queries and exits receiving consistency supervision, and sg denotes stop-gradient. The reference distribution is detached within this term, while the reference exit continues to receive direct retrieval supervision. This objective transfers relative matching information across budgets and depths in addition to the relevance labels provided by the contrastive loss.

For queries with annotated CoT traces, we further introduce selective representation distillation. An additional forward pass appends the annotated reasoning to the query in the format <think>...</think> and produces a teacher representation z_i^T without gradient tracking. The student representation is the full exit $z_i^F = z_{8,36}(q_i)$, computed without the reasoning text.

We apply distillation only when the teacher provides better separation between positive and negative candidates. For a normalized representation z, we define

$$\Delta_i(z) = \max_{c \in \mathcal{B}_i^+} z^\top v_c - \max_{c \in \mathcal{B}_i^-} z^\top v_c. \tag{16}$$

The margin compares the highest-scoring positive with the hardest negative, excluding all other positives from the negative set. For queries with CoT annotations and both positive and negative candidates, the distillation gate is

$$g_i = \mathbf{1}\left[\Delta_i(z_i^T) > \Delta_i(z_i^F)\right]. \tag{17}$$

We set $g_i = 0$ for queries ineligible for this comparison. The gated cosine-distance objective is

$$\mathcal{L}_{\mathrm{CoT}} = \frac{\sum_{i=1}^{B} g_i \left[1 - \cos\left(z_i^F, \mathrm{sg}(z_i^T)\right)\right]}{\max\left(\sum_{i=1}^{B} g_i, 1\right)}. \tag{18}$$

Both the gate and teacher representation are treated as constants during differentiation. Normalizing by the number of selected queries prevents the loss from being diluted when fewer examples pass the gate; the term is zero when none are selected. Whereas distribution consistency transfers relative candidate preferences across exits, CoT distillation aligns the student representation with a reasoning-conditioned target that achieves a higher retrieval margin.

The complete training objective is

$$\mathcal{L}_{\mathrm{exit}} = \mathcal{L}_{\mathrm{full}} + 0.5\,\mathcal{L}_{\mathrm{sub}} + 0.25\,\mathcal{L}_{\mathrm{cons}} + 0.5\,\mathcal{L}_{\mathrm{CoT}}. \tag{19}$$

These terms jointly establish retrieval capability at individual exits, transfer matching information across computation settings, and selectively incorporate supervision from explicit reasoning.

A.3.2 Query-Dependent Representation Aggregation

We elaborate the construction of query-dependent budget weights and derive their gradients under retrieval supervision. We also specify the auxiliary depth-prediction objective used during training below.

The weight-prediction network receives a compact projection of each raw representation:

$$p_{i,k,j} = \mathrm{SiLU}\left(W_p\,\mathrm{LN}(z_{i,k,j}) + b_p\right) \in \mathbb{R}^{128}. \tag{20}$$

The projection is shared across all budgets and selected depths. Its output is concatenated with the budget and depth embeddings, each of dimension 16, and query metadata μ_i. The metadata contains the prefix token count, visual token count, and binary indicators of image and video presence. Both counts are transformed by $\log(1+x)$ and standardized using training-set statistics. To condition the weights on the preceding aggregation result, we additionally compute

$$\gamma_{i,k,j} = z_{i,k,j}^{\top} m_{i,j-1}, \qquad \Delta_{i,k,j} = \|z_{i,k,j} - m_{i,j-1}\|_2. \tag{21}$$

These features describe the alignment and distance between the current representation and the preceding aggregated representation. The inner product equals cosine similarity when both representations have unit norm. At the first selected depth, we use $m_{i,0} = 0$, giving $\gamma_{i,k,1} = 0$ and $\Delta_{i,k,1} = 1$.

The resulting input and budget weights are

$$\begin{aligned} x_{i,k,j} &= [p_{i,k,j}; e_k^K; e_j^D; \mu_i; \gamma_{i,k,j}; \Delta_{i,k,j}], \\ g_{i,k,j} &= \mathrm{MLP}_{\mathrm{Q}}(x_{i,k,j}), \\ \alpha_{i,k,j} &= \frac{\exp(g_{i,k,j})}{\sum_{k' \in \mathcal{K}} \exp(g_{i,k',j})}. \end{aligned} \tag{22}$$

The shared MLP maps the input to a scalar through a hidden layer, with SiLU activation and dropout. The weights combine the original representations:

$$\bar{m}_{i,j} = \sum_{k \in \mathcal{K}} \alpha_{i,k,j} z_{i,k,j}, \qquad m_{i,j} = \mathrm{norm}(\bar{m}_{i,j}). \tag{23}$$

Because the gate depends only on query-side information, the same aggregated representation is used to match all candidates.

We supervise each aggregated representation using a labeled candidate panel $\mathcal{B}_i$, which contains at most 32 candidates and at least one positive. Let $\mathcal{P}_i^Q = \mathcal{P}_i \cap \mathcal{B}_i$ denote the positives in this panel. The similarity and corresponding candidate probability at depth d_j are

$$s_{i,j,c} = m_{i,j}^\top v_c, \qquad \pi_{i,j,c}^Q = \frac{\exp(s_{i,j,c}/\tau_Q)}{\sum_{c' \in \mathcal{B}_i} \exp(s_{i,j,c'}/\tau_Q)}, \tag{24}$$

where $\tau_Q = 0.02$. We maximize the total probability assigned to the positive set. Its negative log-likelihood is

$$\begin{aligned} \ell_{i,j}^Q &= -\log \sum_{c \in \mathcal{P}_i^Q} \pi_{i,j,c}^Q \\ &= -\log \frac{\sum_{c \in \mathcal{P}_i^Q} \exp(s_{i,j,c}/\tau_Q)}{\sum_{c \in \mathcal{B}_i} \exp(s_{i,j,c}/\tau_Q)} \\ &= \mathrm{LSE}_{c \in \mathcal{B}_i}\left(\frac{s_{i,j,c}}{\tau_Q}\right) - \mathrm{LSE}_{c \in \mathcal{P}_i^Q}\left(\frac{s_{i,j,c}}{\tau_Q}\right), \end{aligned} \tag{25}$$

where LSE denotes log-sum-exp. This formulation reduces to softmax cross-entropy for a single positive and accommodates multiple positives without requiring equal probability for each. We use the log-sum-exp expression for numerical stability and average the losses over queries and selected depths:

$$\mathcal{L}_{Q,\mathrm{retr}} = \frac{1}{BJ} \sum_{i=1}^{B} \sum_{j=1}^{J} \ell_{i,j}^Q, \qquad J = |\mathcal{D}|. \tag{26}$$

To derive the gradient, we define the probability distribution restricted to positive candidates:

$$\rho_{i,j,c}^Q = \mathbf{1}[c \in \mathcal{P}_i^Q] \, \frac{\exp(s_{i,j,c}/\tau_Q)}{\sum_{c' \in \mathcal{P}_i^Q} \exp(s_{i,j,c'}/\tau_Q)}. \tag{27}$$

Differentiating the loss gives

$$\frac{\partial \ell_{i,j}^Q}{\partial s_{i,j,c}} = \frac{1}{\tau_Q}\left(\pi_{i,j,c}^Q - \rho_{i,j,c}^Q\right), \tag{28}$$

and therefore

$$\nabla_{m_{i,j}} \ell_{i,j}^Q = \frac{1}{\tau_Q}\left(\sum_{c \in \mathcal{B}_i} \pi_{i,j,c}^Q v_c - \sum_{c \in \mathcal{P}_i^Q} \rho_{i,j,c}^Q v_c\right). \tag{29}$$

The retrieval gradient thus adjusts the aggregated representation according to the difference between the panel-wide candidate distribution and the positive-only distribution.

We next trace this gradient to the budget logits. Fixing i, j and omitting these indices for brevity, differentiation through the budget softmax yields

$$\frac{\partial \bar{m}}{\partial g_k} = \sum_{\ell \in \mathcal{K}} z_\ell \frac{\partial \alpha_\ell}{\partial g_k} = \alpha_k (z_k - \bar{m}). \tag{30}$$

When $\|\bar{m}\|_2$ exceeds the numerical stability floor ϵ, the normalization Jacobian is

$$\frac{\partial m}{\partial \bar{m}} = \frac{1}{\|\bar{m}\|_2}\left(I_H - m m^\top\right). \tag{31}$$

Applying the chain rule gives

$$\frac{\partial \ell^Q}{\partial g_k} = \frac{\alpha_k}{\|\bar{m}\|_2} (z_k - \bar{m})^\top \left(I_H - m m^\top\right) \nabla_m \ell^Q. \tag{32}$$

This expression shows how retrieval supervision learns the budget weights without explicit budget-selection labels. Increasing a budget's logit is favored when its representation moves the normalized combination in a direction that reduces the retrieval loss. The expression describes the current depth's local contribution; deeper losses also propagate through the dependence of subsequent gates on preceding aggregated representations.

The auxiliary objective provides additional supervision about whether deeper processing can correct the current retrieval result. We define

$$\chi_{i,j} = \mathbf{1}\left[\arg\max_{c\in\mathcal{B}_i} s_{i,j,c} \in \mathcal{P}_i^Q\right]. \tag{33}$$

For each nonfinal depth $j < J$, the auxiliary label is

$$y_{i,j} = \begin{cases} 0, & \chi_{i,j} = 0 \text{ and } \exists\, j' > j : \chi_{i,j'} = 1, \\ 1, & \text{otherwise.} \end{cases} \tag{34}$$

Thus, a zero label identifies an incorrect result that can be corrected at a deeper selected depth. A label of one indicates that the current result is already correct or that no deeper result is correct.

The auxiliary head receives the weighted projected representation, depth embedding, query metadata, and summaries of the raw representations' retrieval results. Specifically, let $\widehat{e}_{i,k,j} \in \mathbb{R}^3$ contain the standardized highest similarity, second-highest similarity, and their difference on $\mathcal{B}_i$. The auxiliary input is

$$\xi_{i,j} = \left[\sum_k \alpha_{i,k,j} p_{i,k,j}; e_j^D; \mu_i; \sum_k \alpha_{i,k,j}\widehat{e}_{i,k,j}; \mathcal{H}(\alpha_{i,:,j}); \max_k \alpha_{i,k,j}\right], \tag{35}$$

where $\mathcal{H}(\alpha) = -\sum_k \alpha_k \log \alpha_k$. A separate MLP maps this 153-dimensional input to a logit $b_{i,j}^{\text{aux}}$. Writing $p_{i,j}^{\text{aux}} = \sigma(b_{i,j}^{\text{aux}})$, we obtain

$$\mathcal{L}_{Q,\text{aux}} = -\frac{1}{B(J-1)} \sum_{i=1}^{B}\sum_{j=1}^{J-1} \left[y_{i,j} \log p_{i,j}^{\text{aux}} + (1 - y_{i,j}) \log(1 - p_{i,j}^{\text{aux}})\right]. \tag{36}$$

The labels are constructed from the current forward pass and treated as constants during differentiation. The prediction branch propagates gradients through the shared projection and aggregation weights. Retrieval statistics enter only this auxiliary branch and do not affect the gate's candidate-independent input.

The complete objective is

$$\mathcal{L}_Q = \mathcal{L}_{Q,\text{retr}} + \lambda_{\text{aux}}\mathcal{L}_{Q,\text{aux}}, \qquad \lambda_{\text{aux}} = 0.25. \tag{37}$$

The auxiliary head is used only during training. All selected depths remain available for retrieval and subsequent candidate aggregation at inference time.

A.3.3 Candidate-aware Evidence Aggregation

We detail how the Candidate Aggregator converts matching evidence into ranking scores and derive its training objective over the retrieved candidate pool. The Query Aggregator is fixed throughout this stage.

For each query, we collect the 16 raw representations, four aggregated representations, and base representation into $\mathcal{U}_i = (u_{i,1}, \ldots, u_{i,A})$, with $A = 21$. Each representation retrieves its five highest-scoring candidates. We also use

$$u_i^{\text{aux}} = \text{norm}(z_{i,0} + z_{i,8,J}) \tag{38}$$

to provide an additional retrieval list. Denoting the Top-5 candidates retrieved by representation u from gallery $\mathcal{G}_i$ as $\text{Top}_5(u, \mathcal{G}_i)$, the deduplicated pool is

$$\mathcal{C}_i = \left(\bigcup_{a=1}^{A} \text{Top}_5(u_{i,a}, \mathcal{G}_i)\right) \cup \text{Top}_5(u_i^{\text{aux}}, \mathcal{G}_i). \tag{39}$$

The auxiliary representation expands the candidate pool but is not included among the 21 representations used for matching.

For every $c \in \mathcal{C}_i$, we compute its interaction with each query representation:

$$t_{i,c,a} = M_{\text{sem}}\left(P_Q(u_{i,a}) \odot P_C(v_c)\right) + M_{\text{retr}}(r_{i,c,a}) + e_a. \tag{40}$$

The projections P_Q and P_C map query and candidate embeddings to 128 dimensions, and M_{sem} transforms their elementwise product into a matching feature. The 11-dimensional vector $r_{i,c,a}$ records similarity, retrieval membership and rank, relative scores, and support across representations and depths. Its projection has the same dimension as the semantic matching feature. The learned embedding e_a identifies which raw, aggregated, or base representation provides the evidence.

Matching features are computed for every query-representation–candidate pair in the pool. In particular, we evaluate the exact similarity $u_{i,a}^\top v_c$ even when representation a did not retrieve candidate c. Retrieval membership is recorded separately, so a missing entry in a Top-5 list is not treated as zero similarity.

Query-Representation Attention first summarizes the evidence associated with each candidate. Candidate Self-Attention then compares these summaries within the pool:

$$\begin{aligned} h_{i,c} &= \mathrm{Attn}_{\mathrm{Q}}\left([t_{\mathrm{CLS}}, t_{i,c,1}, \ldots, t_{i,c,A}]\right)_{\mathrm{CLS}}, \\ [\widetilde{h}_{i,c}]_{c\in\mathcal{C}_i} &= \mathrm{Attn}_{\mathrm{C}}\left([h_{i,c}]_{c\in\mathcal{C}_i}\right). \end{aligned} \tag{41}$$

Candidate padding is masked, and no candidate positional embeddings are added. Consequently, permuting the candidates permutes their output representations correspondingly.

A shared linear layer predicts a candidate-specific score correction,

$$\delta_{i,c} = w^\top \widetilde{h}_{i,c} + b. \tag{42}$$

Combining this correction with the base retrieval score gives

$$\begin{aligned} a_{i,c} &= z_{i,0}^\top v_c + z_{i,8,J}^\top v_c, \\ S_{i,c} &= \beta\left(a_{i,c} - \max_{c'\in\mathcal{C}_i} a_{i,c'}\right) + \delta_{i,c}, \qquad \beta = 16. \end{aligned} \tag{43}$$

Subtracting the pool maximum adds the same constant to every candidate logit and therefore preserves both ranking and softmax probabilities. The scoring layer is zero-initialized, so training starts from the base retrieval ranking and learns corrections to it.

The ranking objective maximizes the total probability assigned to relevant candidates in the retrieved pool. Let $\mathcal{P}_i^C = \mathcal{P}_i \cap \mathcal{C}_i$. For queries with $\mathcal{P}_i^C \neq \varnothing$, we define

$$\pi_{i,c}^C = \frac{\exp(S_{i,c})}{\sum_{c'\in\mathcal{C}_i}\exp(S_{i,c'})}. \tag{44}$$

Following the positive-set likelihood used for the Query Aggregator, the per-query loss is

$$\begin{aligned} \ell_i^C &= -\log \sum_{c\in\mathcal{P}_i^C} \pi_{i,c}^C \\ &= \mathrm{LSE}_{c\in\mathcal{C}_i}(S_{i,c}) - \mathrm{LSE}_{c\in\mathcal{P}_i^C}(S_{i,c}). \end{aligned} \tag{45}$$

The final scores are used directly as logits, without an additional temperature.

We weight queries inversely to their training dataset sizes. For T datasets containing $N_1, \ldots, N_T$ queries, let $N = \sum_{t=1}^T N_t$. A query from dataset $t(i)$ receives weight

$$\omega_i = \frac{N}{T\, N_{t(i)}}. \tag{46}$$

Let $\mathcal{I}_+ = \{i : \mathcal{P}_i^C \neq \varnothing\}$ denote the valid query indices in the current minibatch. The objective is their weighted mean:

$$\mathcal{L}_C = \frac{\sum_{i\in\mathcal{I}_+}\omega_i \ell_i^C}{\sum_{i\in\mathcal{I}_+}\omega_i}, \qquad \mathcal{I}_+ \neq \varnothing. \tag{47}$$

To express the gradient, we define the positive-restricted distribution

$$\rho_{i,c}^C = \mathbf{1}[c\in\mathcal{P}_i^C]\,\frac{\exp(S_{i,c})}{\sum_{c'\in\mathcal{P}_i^C}\exp(S_{i,c'})}. \tag{48}$$

Because the base retrieval score is fixed and the learned correction enters additively,

$$\frac{\partial \mathcal{L}_C}{\partial \delta_{i,c}} = \frac{\omega_i}{\sum_{i' \in \mathcal{I}_+} \omega_{i'}} \left(\pi^C_{i,c} - \rho^C_{i,c} \right). \tag{49}$$

This gradient trains the matching networks and both attention modules through the final scoring layer. Their attention weights therefore emerge from ranking supervision, without additional attention-level targets. The loss does not backpropagate through the fixed Query Aggregator or the discrete retrieval and deduplication operations.

A.4 Detailed MMEB-V2 Evaluation Results

Table 9: Detailed evaluation results on MMEB-V2. CLS: classification, QA: question answering, RET: retrieval, GD: grounding, MRET: moment retrieval, VDR: ViDoRe, VR: VisRAG, OOD: out-of-domain. The highest and second-highest values in each row are highlighted in bold and underline, respectively. Rankings use unrounded values; all results are displayed to one decimal place.

	VLM2Vec-V2	UME-R1-7B	TTE_o-7B	Embed-RL-4B	PLUME	TWN-4B	TWN-8B	DUET-4B	DUET-8B
Avg-All (78 tasks)	58.0	64.5	68.6	68.1	61.6	66.6	68.7	67.0	**69.2**
Avg-Image (36 tasks, Hit@1)	64.9	71.3	**74.2**	70.1	66.3	71.3	73.4	69.7	71.2
Avg-Video (18 tasks, Hit@1)	34.6	47.5	46.8	53.0	44.1	45.7	48.2	53.5	**54.8**
Avg-VisDoc (24 tasks, NDCG@5)	65.4	67.1	76.4	74.7	67.5	75.3	**77.0**	73.2	76.8
I-CLS (10)	62.9	67.1	69.7	63.7	66.5	68.6	**70.2**	65.7	65.8
I-QA (10)	56.4	69.2	72.4	70.5	59.2	71.7	**74.3**	65.9	68.0
I-RET (12)	69.6	71.9	**74.0**	71.3	67.6	68.0	70.8	69.4	71.2
I-VG (4)	77.1	84.9	90.6	91.4	79.7	87.0	87.1	89.8	**93.0**
V-CLS (5)	39.3	48.6	49.1	**57.6**	45.0	45.8	50.1	51.6	53.4
V-QA (5)	34.3	60.7	60.6	58.4	52.3	63.4	64.0	66.3	**67.9**
V-RET (5)	28.8	38.2	36.4	45.1	33.5	33.6	34.8	45.7	**46.3**
V-MR (3)	36.8	39.3	37.2	**49.5**	46.7	36.6	40.8	48.6	49.4
VD-Vidore-V1 (10)	75.5	75.7	**84.1**	80.2	72.1	81.5	82.5	78.4	82.5
VD-Vidore-V2 (4)	45.1	50.5	**62.7**	53.4	49.8	53.6	56.7	52.1	56.4
VD-VisRAG (6)	79.6	83.7	**91.9**	84.9	78.1	84.8	86.2	82.7	85.6
VD-OOD (4)	39.6	37.6	47.6	67.1	57.4	67.3	**70.0**	67.0	69.8
ImageNet-1K	80.8	80.4	**84.3**	79.5	74.1	81.0	81.7	74.2	75.9
N24News	73.0	82.3	**83.1**	48.3	81.1	75.3	75.6	67.0	79.6
HatefulMemes	55.8	**79.0**	67.4	66.2	75.5	72.0	74.0	68.7	61.3
VOC2007	84.9	90.8	86.6	79.5	86.1	86.0	89.4	93.3	**94.2**
SUN397	70.9	80.3	78.9	79.2	76.9	80.9	**81.4**	77.0	78.8
Place365	36.1	46.8	44.6	43.1	42.4	45.1	**47.2**	43.2	45.8
ImageNet-A	47.6	53.9	**60.4**	58.1	50.8	59.8	59.9	53.5	46.3
ImageNet-R	89.3	90.1	90.5	88.2	87.5	90.1	**91.7**	86.9	86.5
ObjectNet	65.1	42.3	72.6	75.4	61.5	76.6	**77.4**	74.0	72.6
Country211	25.8	25.0	**29.0**	19.4	25.0	18.9	23.7	19.5	17.0
OK-VQA	51.7	71.7	**74.7**	67.3	60.5	70.6	71.1	70.9	68.8
A-OKVQA	44.0	58.7	**66.1**	59.3	49.9	62.6	63.9	59.7	59.9
DocVQA	90.1	93.8	**95.6**	94.3	89.9	94.7	94.7	92.3	92.1
InfographicsVQA	59.1	79.2	77.5	77.5	59.6	80.0	**82.3**	77.4	73.1
ChartQA	48.1	75.1	70.9	80.9	49.8	81.7	**84.4**	70.0	69.4
Visual7W	52.8	55.2	57.9	55.3	47.6	57.8	**62.9**	55.1	59.5
ScienceQA	38.1	53.7	60.0	**61.6**	42.9	56.5	61.2	53.7	57.2
VizWiz	43.3	51.6	53.8	56.2	46.5	55.2	**57.8**	55.1	57.0
GQA	65.4	69.3	**80.9**	68.5	69.1	75.4	78.0	38.9	55.6
TextVQA	71.6	83.5	87.0	84.3	78.9	82.4	86.7	85.5	**87.5**
VisDial	82.7	80.7	84.4	**84.9**	72.6	80.8	84.0	81.1	82.8
CIRR	57.3	55.3	**65.1**	61.2	54.6	47.5	47.9	54.0	58.7
VisualNews_t2i	74.7	76.8	**78.5**	73.7	71.3	72.0	75.1	72.4	73.7
VisualNews_i2t	78.3	**82.0**	81.3	73.9	72.7	77.3	79.8	74.5	73.8
MSCOCO_t2i	75.9	78.3	77.9	78.9	74.1	77.6	**79.2**	76.5	78.8
MSCOCO_i2t	71.1	71.4	73.1	76.3	69.8	75.6	**77.3**	73.4	74.5
NIGHTS	68.4	68.1	69.8	66.4	68.0	67.4	**71.4**	67.4	68.4
WebQA	90.6	**90.9**	90.8	90.5	89.1	88.5	90.3	88.3	90.2
FashionIQ	19.6	23.4	29.7	**31.9**	20.3	9.5	18.5	23.9	27.7
Wiki-SS-NQ	67.6	72.5	70.5	69.6	68.6	70.9	72.0	70.7	**74.0**
OVEN	64.8	71.4	**72.7**	60.7	68.4	63.7	66.8	65.5	64.9
EDIS	84.2	92.0	**93.9**	87.4	81.8	85.8	87.3	85.2	86.6
MSCOCO	67.6	72.7	74.1	**93.6**	66.9	78.2	78.1	83.7	90.9
RefCOCO	87.0	91.4	**97.7**	95.9	86.5	92.5	93.8	93.7	95.6
RefCOCO-Matching	86.3	91.1	**96.3**	88.0	88.4	91.1	91.5	92.6	92.4
Visual7W-Pointing	69.0	84.2	**94.3**	87.9	74.9	86.0	85.0	89.0	93.0
K700	38.0	42.8	55.0	**56.8**	42.2	52.2	55.4	46.7	54.0
SmthSmthV2	42.8	50.4	44.9	**59.5**	44.8	33.0	36.8	50.6	56.8
HMDB51	40.9	58.3	51.7	**60.1**	51.2	45.8	50.7	54.3	51.3
UCF101	60.0	70.0	64.2	**78.5**	66.5	70.9	76.1	73.3	73.7
Breakfast	14.8	21.5	29.7	**33.0**	20.1	27.0	31.5	**33.0**	31.3
MVBench	33.7	58.2	59.5	55.9	47.4	**62.6**	**62.6**	57.7	59.0
Video-MME	30.7	47.3	53.1	50.5	40.0	55.0	55.3	59.1	**62.8**
NExTQA	20.9	69.6	70.1	58.2	57.3	67.1	67.4	72.4	**73.4**
EgoSchema	34.0	52.4	55.6	52.8	47.8	54.2	54.6	62.2	**63.4**
ActivityNetQA	52.3	76.0	64.6	74.4	69.2	77.9	80.1	80.3	**80.7**
DiDeMo	30.4	40.0	34.9	46.8	32.7	31.8	33.5	**49.9**	48.1
MSR-VTT	28.3	38.9	37.6	46.2	36.2	37.1	38.2	**46.8**	44.2
MSVD	48.1	60.8	58.5	65.8	56.1	54.8	56.5	64.0	**67.9**
VATEX	26.5	32.6	31.0	43.4	28.2	29.0	29.8	43.0	**46.0**
YouCook2	10.6	18.5	19.9	23.3	14.5	15.3	16.0	24.6	**25.3**
QVHighlight	49.4	54.9	51.0	73.6	57.1	45.0	50.2	75.8	**76.6**
Charades-STA	20.2	21.9	18.9	25.0	19.4	23.2	25.3	27.4	**27.4**
MomentSeeker	40.8	41.1	41.5	49.9	**63.5**	41.6	46.9	42.7	44.1
ViDoRe_arxivqa	80.6	73.6	84.6	88.7	72.6	88.4	89.1	83.4	**89.2**
ViDoRe_docvqa	44.9	41.1	46.0	47.5	36.2	48.5	**50.2**	43.3	47.0
ViDoRe_infovqa	83.7	80.8	88.7	86.9	79.0	88.2	89.8	86.8	**90.0**
ViDoRe_tabfquad	89.2	90.2	**94.7**	**94.7**	88.8	91.2	92.6	91.9	92.9
ViDoRe_tatdqa	43.8	46.7	**59.4**	54.8	36.6	52.7	54.4	48.3	55.5
ViDoRe_shiftproject	60.8	65.0	**81.6**	69.0	64.8	75.1	76.3	72.0	76.0
ViDoRe_artificial_intelligence	88.5	89.5	**98.1**	91.6	83.8	97.4	97.2	89.0	94.0
ViDoRe_energy	86.5	85.7	**93.5**	88.1	82.6	88.8	89.4	89.0	93.0
ViDoRe_government_reports	85.0	89.8	**96.7**	90.7	83.2	89.2	90.5	89.0	92.0
ViDoRe_healthcare_industry	92.2	94.3	**97.9**	90.4	91.1	95.6	95.5	91.0	95.0
ViDoRe_esg_reports_human_labeled_v2	45.6	50.4	**69.4**	59.8	52.1	54.8	55.3	46.2	54.9
ViDoRe_biomedical_lectures_v2_multilingual	44.3	50.7	**60.8**	50.1	48.2	54.1	59.7	49.2	54.9
ViDoRe_economics_reports_v2_multilingual	43.0	57.8	60.4	53.9	49.6	57.9	59.7	63.4	**65.4**
ViDoRe_esg_reports_v2_multilingual	46.6	43.2	**60.3**	49.7	49.0	47.6	52.1	49.6	50.3
VisRAG_ArxivQA	76.9	80.5	**94.5**	86.9	71.6	85.4	87.0	80.6	84.6
VisRAG_ChartQA	83.7	85.0	**91.2**	88.5	80.8	84.0	84.0	81.4	85.6
VisRAG_MP-DocVQA	88.1	83.4	**90.1**	79.3	74.9	84.4	87.3	84.8	84.1
VisRAG_SlideVQA	84.1	91.5	**95.6**	92.6	88.9	92.6	94.1	92.0	92.7
VisRAG_InfoVQA	82.3	89.2	**93.0**	89.6	85.7	90.8	91.9	89.1	92.6
VisRAG_PlotQA	75.9	72.7	**86.9**	72.4	66.2	71.5	72.9	68.3	74.2
ViDoSeek-page	29.1	21.3	35.0	84.4	80.6	81.3	**86.6**	60.1	67.3
ViDoSeek-doc	79.0	75.3	84.4	82.4	76.9	82.8	84.6	98.3	**98.4**
MMLongBench-page	15.8	12.3	20.7	51.0	39.9	53.4	**55.5**	42.6	45.7
MMLongBench-doc	63.0	41.3	50.4	50.7	32.0	51.8	53.3	67.1	**67.9**