Skip-then-Select: UNDERSTANDING VISUAL TOKEN DYNAMICS IN UNIVERSAL MULTIMODAL EMBEDDINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

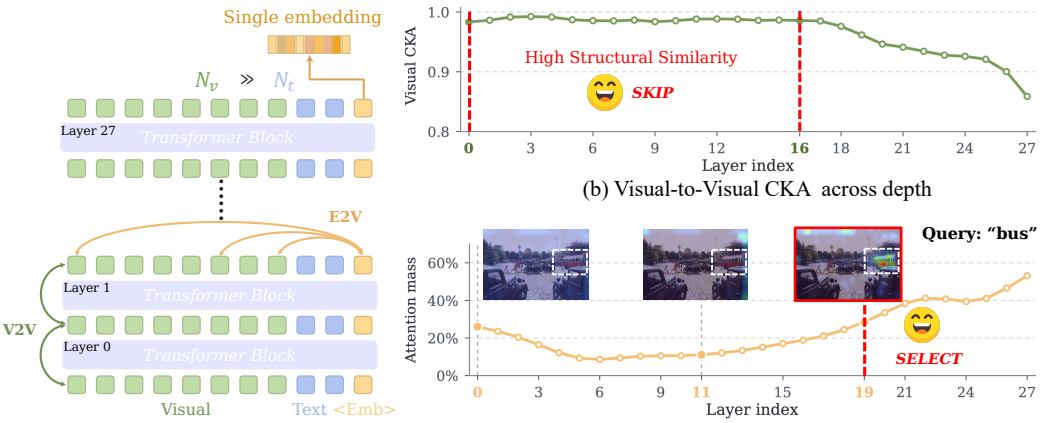
Universal multimodal embedding models harness the capabilities of multimodal large language models to encode diverse inputs into a shared representation space, but generating a single embedding requires costly processing of long visual sequences through the entire language backbone. How visual information evolves across layers and contributes to the final embedding has not been thoroughly investigated in UMEs. We therefore jointly analyze cross-layer visual-state evolution and within-layer embedding-to-visual interaction. This analysis reveals two patterns: visual representations retain similar relational structure across shallow-to-middle layers, while embedding-to-visual attention follows a U-shaped trajectory in total mass and increasingly focuses on task-relevant regions in deep layers. Building on these findings, we propose Visual *Skip-then-Select* (ViS^2), which reduces both the depth and sequence length of visual computation. The Skip module preserves early multimodal interaction and routes shallow visual features through a lightweight projector to bypass middle-layer processing. The Select module reintroduces the full visual sequence for interaction with text and embedding tokens in deeper layers, then prunes visual tokens using embedding-to-visual attention. We train the modified encoder in two stages, first aligning intermediate representations with the original model and then distilling its final embeddings. Evaluated on 36 MMEB datasets, ViS^2 retains 95.6% of the original performance while reducing FLOPs by 75.0% and achieving a $2.27\times$ end-to-end speedup, averaged across two UME models. Our models and code will be released.

1 INTRODUCTION

Multimodal embedding aims to project inputs from different modalities into a shared vector space. Conventional approaches primarily achieve this via modality-specific encoders Radford et al. (2021); Zhai et al. (2023); Tschannen et al. (2025). More recently, universal multimodal embedding (UME) has emerged to represent heterogeneous inputs and their combinations within a single model Jiang et al. (2025); Lin et al. (2025). Multimodal large language models (MLLMs) provide a promising backbone for UME, combining multimodal understanding with instruction-following capabilities Liu et al. (2025); Gu et al. (2025a). These models use the final hidden state of an $\langle\mathrm{Emb}\rangle$ token as the output embedding (Figure 1a) Jiang et al. (2024); Zhang et al. (2025b). Yet obtaining this compact representation often requires processing every visual token through all language-model layers Tang et al. (2026); Hönicke et al. (2026). Since visual tokens can far outnumber text tokens, their full-depth processing is costly for real-world applications, which demand high throughput and low latency Khattab & Zaharia (2020); Karpukhin et al. (2020).

To reduce this overhead, existing MLLM acceleration methods prune or merge redundant visual tokens Chen et al. (2024); Zhang et al. (2025c); Yang et al. (2025) or sparsify their computation across layers Liu et al. (2026); Choraria et al. (2026); Bulat et al. (2026). UME efforts have also explored visual token compression (Li et al., 2026c) and compression-based training (Li et al., 2026a). However, these MLLM acceleration methods primarily target generation, while the UME compression approaches above offer limited insight into how $\langle\mathrm{Emb}\rangle$ aggregates visual information across layers. This leads us to ask: *when can full visual processing be bypassed while retaining visual states for aggregation, and when can visual tokens be pruned while preserving retrieval quality?*

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



(a) Full-depth embedding extraction

(b) Visual-to-Visual CKA across depth

(c) <Emb>-to-Visual attention across depth

Figure 1: **Full-depth processing and layer-wise changes in visual and ⟨Emb⟩ representations.** (a) Visual tokens ($N_v \gg N_t$) traverse the full language model before a single embedding is extracted. Visual-to-Visual (V2V) denotes the relationship between visual states across layers, whereas ⟨Emb⟩-to-Visual (E2V) denotes the interaction between ⟨Emb⟩ and visual tokens within a layer. (b) V2V centered kernel alignment (CKA) between visual states at neighboring layers. High CKA indicates similar representational structure. (c) E2V attention mass follows a U-shaped trajectory across depth, while attention becomes concentrated on task-relevant regions (e.g., the bus) in deep layers.

To answer these questions, we examine how visual representations change across layers and how the ⟨Emb⟩ token interacts with them within each layer. These two perspectives reveal complementary patterns, as shown in Figure 1(b,c): (1) V2V structural similarity remains high across shallow-to-middle layers; (2) E2V attention weakens in middle layers, then strengthens and focuses on task-relevant regions in deeper layers. These observations suggest that visual tokens may not require full processing at every layer.

Building on these findings, we propose Visual *Skip-then-Select* (ViS^2), a hybrid framework combining a lightweight visual highway with late token pruning. After early multimodal interaction, visual states bypass the middle LLM layers via a lightweight highway. In deeper layers, the projected visual states re-enter the backbone for renewed multimodal fusion. Attention from ⟨Emb⟩ then guides the selection of visual tokens for the remaining layers. We train ViS^2 using two-stage training from the original embedder. Our contributions are threefold:

- We introduce a joint perspective and analysis on visual states evolution and embedding-to-visual interaction to guide layer-wise visual computation in MLLM-based UMEs.
- We propose ViS^2, a framework that effectively reduces both the depth and sequence length of visual processing in MLLM-based UMEs.
- Across two MLLM-based embedders and 36 datasets, ViS^2 achieves the highest overall score and lowest inference latency among the evaluated compression methods.

2 Related Work

Universal multimodal embedding. Multimodal representation learning has evolved from aligning modality-specific encoders, as in CLIP and SigLIP (Radford et al., 2021; Zhai et al., 2023), toward using MLLMs to encode heterogeneous inputs and task instructions into a shared embedding space. E5-V demonstrates multimodal transfer through prompting and text-only training, while VLM2Vec introduces MMEB and contrastive adaptation across classification, retrieval, visual question answering, and grounding (Jiang et al., 2024; 2025). Subsequent approaches improve discriminative representations through modality-aware hard negatives, fused-modal training data, and embedding distillation (Lin et al., 2025; Zhang et al., 2025b; Gu et al., 2025a; Cao et al., 2026). The scope also extends beyond natural images: VLM2Vec-V2 and Qwen3-VL-Embedding support videos and visual documents, while WAVE incorporates audio into unified multimodal re-

trieval (Meng et al., 2025; Li et al., 2026b; Tang et al., 2026). These developments establish MLLMs as versatile backbones. Our work focuses on the computation required to form the final embedding representations, examining how visual states contribute across the language-model layers.

Visual token compression and sparse computation in MLLMs. Existing methods reduce visual computation by shortening token sequences or changing their processing across depth. LLaVA-PruMerge and VisionZip reduce the number of visual tokens before they enter the language model (Shang et al., 2025; Yang et al., 2025). Within the backbone, FastV prunes after early layers, SparseVLM uses text-guided token scoring and recycling, and PyramidDrop progressively reduces visual-token budgets (Chen et al., 2024; Zhang et al., 2025c; Xing et al., 2025). Selection criteria also extend beyond attention: DivPrune selects tokens to maximize diversity (Alvar et al., 2025). Depth-wise approaches modify where visual tokens participate in computation. DeepInsert introduces visual tokens directly into intermediate layers, bypassing early processing (Choraria et al., 2026). ViCA lets visual tokens bypass self-attention and feed-forward updates, retaining sparse text-to-vision cross-attention; VISOR additionally allocates visual self-attention dynamically according to input complexity (Liu et al., 2026; Bulat et al., 2026). Motivated by changes in token relevance across depth, Reroute allows deferred tokens to bypass a stage and compete for re-entry at subsequent routing decisions (Yang et al., 2026). These methods primarily target generation, leaving the computational requirements of embedding formation insufficiently explored. We develop an embedding-specific acceleration framework guided by our findings on visual-state evolution and $\langle\mathrm{Emb}\rangle$-to-Visual interaction.

Acceleration of universal multimodal embedding models. Efforts to improve efficiency in MLLM-based embedding address both training and inference. CoMa pretrains models to aggregate semantic information into compression tokens before contrastive learning, improving the data efficiency of embedding adaptation (Li et al., 2026a). For inference, the Matryoshka Multimodal Embedder supports multiple visual-token granularities for text–image interleaved retrieval (Zhang et al., 2025a), while Magic-MM-Embedding compresses visual features before the language backbone and combines this architecture with progressive multimodal and contrastive training (Li et al., 2026c). MoEMB scales embedding capacity through mixture-of-experts backbones and studies adaptive computation through token pruning, expert skipping, and layer skipping, including their behavior across depth (Cui et al., 2026). Despite these efforts, how to coordinate visual-state preservation and token selection across layers during embedding formation remains underexplored.

3 METHOD

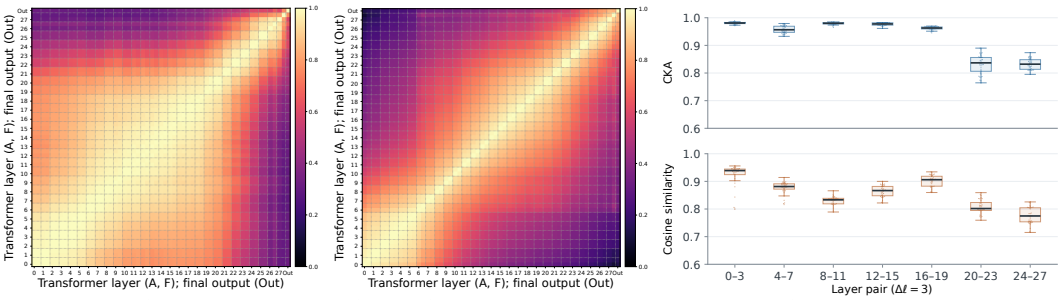
We consider an MLLM-based UME encoder that maps an input and its task instruction to a single vector in a shared embedding space. The language-model input consists of N_v visual tokens, N_t text tokens including the task instruction, and a final embedding token $\langle\mathrm{Emb}\rangle$. Denoting their input representations by $\boldsymbol{x}_j$, $\boldsymbol{u}_j$, and $\boldsymbol{e}$, respectively, we write the sequence as

$$\boldsymbol{x} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_v}; \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{N_t}; \boldsymbol{e}]. \tag{1}$$

The original encoder processes all tokens through L language-model layers, indexed by $\ell = 0, \ldots, L-1$. We denote the states before layer ℓ by $\boldsymbol{H}^{(\ell)} = [\boldsymbol{H}_v^{(\ell)}; \boldsymbol{H}_t^{(\ell)}; \boldsymbol{h}_e^{(\ell)}]$, where the three components correspond to visual, text, and embedding tokens. Thus, $\boldsymbol{H}^{(0)} = \boldsymbol{x}$, layer ℓ maps $\boldsymbol{H}^{(\ell)}$ to $\boldsymbol{H}^{(\ell+1)}$, and $\boldsymbol{h}_e^{(L)}$ is the output embedding. Our objective is to reduce language-backbone computation while preserving retrieval quality. We first analyze how visual representations and $\langle\mathrm{Emb}\rangle$-to-Visual attention evolve across layers (Section 3.1), then introduce Visual *Skip-then-Select* (Section 3.2) and its two-stage training scheme (Section 3.3).

3.1 LAYER-WISE ANALYSIS OF REPRESENTATIONS AND ATTENTION

We conduct a layer-wise analysis using samples from 36 sub-datasets of MMEB Jiang et al. (2025). We examine two complementary aspects: representation dynamics, measured by cross-layer similarity of visual and $\langle\mathrm{Emb}\rangle$ states, and $\langle\mathrm{Emb}\rangle$-to-Visual attention, characterized by its intensity and spatial distribution. Additional examples are provided in Appendix A.2. Together, these analyses yield three findings.



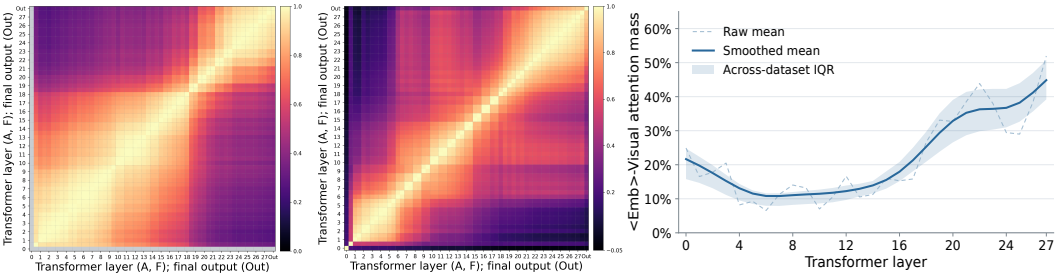
(a) Visual token CKA (b) Visual token cosine similarity (c) Visual similarity across datasets

Figure 2: **Cross-layer similarity of visual hidden states.** (a) Linear CKA between visual hidden states. (b) Mean cosine similarity between corresponding visual tokens. Each heatmap axis contains 57 states: the attention (A) and FFN (F) inputs at layers 0–27, followed by the final output (Out). (c) Across-dataset distributions for selected pairs of attention-input states with a fixed layer gap of three. Dots denote datasets; boxes show the interquartile range and central lines indicate medians.

Finding 1: Visual-to-Visual (V2V) structural similarity remains high across shallow-to-middle layers. We use linear centered kernel alignment (CKA) Kornblith et al. (2019) to measure structural similarity and cosine similarity to measure directional alignment between corresponding tokens (definitions in Appendix A.1). Figure 2(a) shows high CKA across shallow-to-middle layers, while cosine similarity generally decreases with layer separation (b), suggesting that relational structure is largely preserved despite changes in token directions. Panel (c) shows consistently high CKA with little cross-dataset variation for the selected shallow-to-middle layer pairs, supporting the generality of this finding. This motivates a lightweight projector to carry visual features across skipped layers.

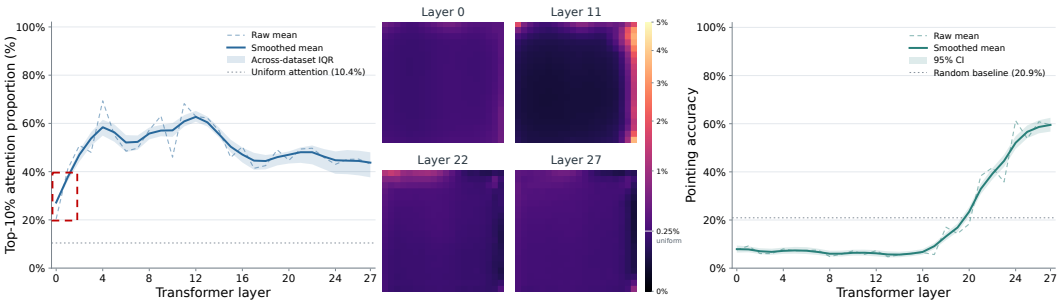
Finding 2: Embedding representations evolve in stages, while $\langle\mathrm{Emb}\rangle$-to-Visual (E2V) attention follows a U-shaped trajectory. Figure 3(a, b) reveals stage-dependent changes in $\langle\mathrm{Emb}\rangle$ representations, including a marked directional change in layer 0 and reduced similarity between middle and deep layers. Panel (c) shows a U-shaped trajectory in E2V attention intensity: the total attention allocated to visual tokens is relatively high at layer 0, falls in the shallow-to-middle layers, and rises again toward the final layers. Together, these patterns motivate retaining full multimodal interaction at the beginning and end of the network while reducing visual computation in the middle.

Finding 3: Visual attention exhibits a Broad–Anchor–Focus pattern across depth. At layer 0, E2V attention is relatively diffuse (*Broad*), as indicated by the small share allocated to the top 10% of visual tokens in Figure 4(a). This share increases in subsequent layers, while the mean spatial



(a) <Emb> token CKA (b) <Emb> token cosine similarity (c) Attention mass of <Emb>-to-visual across layers

Figure 3: **Cross-layer similarity of $\langle\mathrm{Emb}\rangle$ and variation in $\langle\mathrm{Emb}\rangle$-to-Visual attention intensity across layers.** (a) Linear CKA between $\langle\mathrm{Emb}\rangle$ hidden states. Gray entries are undefined because the initial $\langle\mathrm{Emb}\rangle$ representation is constant across inputs. (b) Mean cosine similarity between corresponding $\langle\mathrm{Emb}\rangle$ representations. (c) Attention intensity from $\langle\mathrm{Emb}\rangle$ to visual tokens, averaged over heads and summed over visual tokens. Dashed and solid curves show the raw and Gaussian-smoothed means; shading denotes the interquartile range of smoothed dataset means.



(a) Attention proportion on top-10% (b) Attention spatial distribution (c) Top-1 attention token hit rate

Figure 4: **Spatial allocation of ⟨Emb⟩-to-Visual attention across layers.** (a) Layer-wise changes in the proportion of total E2V attention allocated to the top $\lceil 0.1N_v \rceil$ most-attended visual tokens. (b) Mean normalized spatial attention on a shared 20×20 grid. Uniform attention is 0.25% per cell. (c) Top-1 attention hit rate on RefCOCO queries. A hit occurs when the most-attended visual token's center lies inside the referred bounding box. Dashed curves show raw means and solid curves show Gaussian smoothing.

map at layer 11 in panel (b) shows concentration near image boundaries (*Anchor*), suggesting a shared positional bias consistent with visual attention sinks (Kang et al., 2025). The top-1 attention hit rate in panel (c) remains low through the middle layers but rises sharply in deeper layers (*Focus*), indicating that the most-attended visual tokens increasingly fall within task-relevant regions.

3.2 Skip-then-Select Architecture

These findings motivate a design that preserves visual information for late aggregation while reducing intermediate visual computation. We therefore propose Visual *Skip-then-Select* (ViS^2), a hybrid architecture combining a visual skip connection with deep-layer token selection (Figure 5). The architecture comprises two stages: *Skip* preserves early multimodal interaction and routes the resulting visual features directly to deeper layers through a lightweight projector; *Select* restores fusion with the full visual sequence, then uses E2V attention to select tokens for the remaining layers. Together, these stages reduce visual computation along two axes: depth through layer skipping, and sequence length through token selection.

Skip: early interaction and visual bypass (layers 0 to $i-1$). We retain the first LLM layer with all visual, text, and ⟨Emb⟩ tokens. This preserves the early interaction identified in Findings 2 and 3, allowing text and embedding states to incorporate visual information before visual computation is skipped. The subsequent middle layers process only text and embedding tokens:

$$[\boldsymbol{H}_t^{(\ell+1)}, \boldsymbol{h}_e^{(\ell+1)}] = \text{Layer}_\ell\big([\boldsymbol{H}_t^{(\ell)}, \boldsymbol{h}_e^{(\ell)}]\big), \qquad \ell = 1, \ldots, i-1. \tag{2}$$

In parallel, a visual skip connection takes the visual features after early fusion and passes them through a lightweight projector $\mathcal{P}$ for re-entry at layer i:

$$\boldsymbol{V}_\mathcal{P} = \mathcal{P}(\boldsymbol{H}_v^{(1)}). \tag{3}$$

This lightweight path allows hundreds or thousands of visual tokens to bypass almost half of the LLM layers, reducing the sequence length in layers 1 to $i-1$ from $N_v + N_t + 1$ to $N_t + 1$.

Select: post-fusion processing and token selection (layers i to $L-1$). At layer i, the full projected visual sequence rejoins the text and embedding states. We restore multimodal fusion through layer s, spanning $s - i + 1$ layers:

$$\boldsymbol{H}^{(i)} = [\boldsymbol{V}_\mathcal{P}; \boldsymbol{H}_t^{(i)}; \boldsymbol{h}_e^{(i)}], \tag{4}$$

$$\boldsymbol{H}^{(\ell+1)} = \text{Layer}_\ell\big(\boldsymbol{H}^{(\ell)}\big), \qquad \ell = i, \ldots, s. \tag{5}$$

After restoring fusion, we use E2V attention to select visual tokens for the remaining layers. Specifically, at layer s, we score each visual token by the head-averaged attention it receives from ⟨Emb⟩

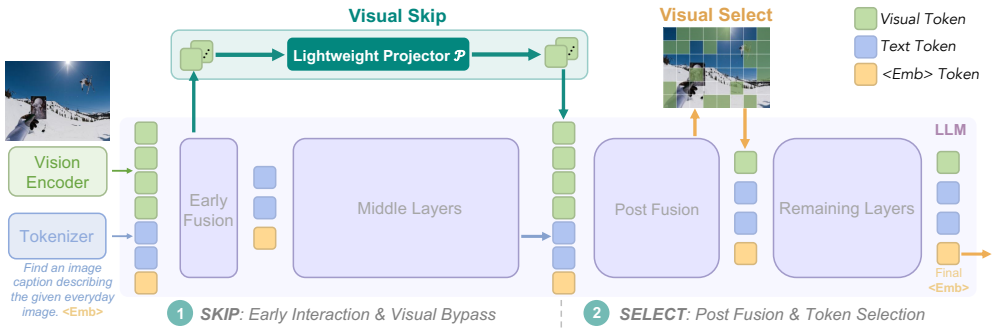


Figure 5: **Overview of Skip-then-Select.** (1) **Skip:** after early multimodal interaction, only text and ⟨Emb⟩ tokens traverse the middle LLM layers, while a lightweight projector $\mathcal{P}$ routes visual features after early fusion directly to deeper layers. (2) **Select:** the full projected visual sequence rejoins the text and embedding tokens for post-fusion processing. E2V attention then guides token selection, and the remaining layers process the reduced sequence to produce the final embedding.

and select the N_{keep} highest-scoring tokens:

$$a_j = \frac{1}{N_h} \sum_{h=1}^{N_h} A_{\text{Emb},j}^{(s,h)}, \qquad \mathcal{I} = \text{TopK}\left(\{a_j\}_{j=1}^{N_v}, N_{\text{keep}}\right), \tag{6}$$

where N_h is the number of attention heads and TopK returns the indices of the N_{keep} largest scores. The remaining layers, from $s+1$ to $L-1$, operate on the reduced sequence:

$$\boldsymbol{H}_{\text{sel}}^{(s+1)} = [\boldsymbol{H}_v^{(s+1)}[\mathcal{I}]; \boldsymbol{H}_t^{(s+1)}; \boldsymbol{h}_e^{(s+1)}], \tag{7}$$

$$\boldsymbol{H}_{\text{sel}}^{(\ell+1)} = \text{Layer}_\ell\left(\boldsymbol{H}_{\text{sel}}^{(\ell)}\right), \qquad \ell = s+1, \ldots, L-1. \tag{8}$$

These layers process $N_{\text{keep}} + N_t + 1$ tokens and produce the final ⟨Emb⟩ representation.

3.3 Two-Stage Training

We use the original encoder as a frozen teacher to align intermediate states and then distill final embeddings.

Stage 1: intermediate-state alignment. We train the projector and post-fusion layers with MSE and cosine losses only on the full output of layer s, before token selection:

$$\mathcal{L}_{\text{hid}} = \text{MSE}\left(\boldsymbol{H}^{(s+1)}, \hat{\boldsymbol{H}}^{(s+1)}\right) + 1 - \text{Cos}\left(\boldsymbol{H}^{(s+1)}, \hat{\boldsymbol{H}}^{(s+1)}\right), \tag{9}$$

where hats denote teacher states and Cos averages cosine similarity over corresponding tokens.

Stage 2: final-embedding distillation. We then train the full student path to align its final embedding $\boldsymbol{z} = \boldsymbol{h}_e^{(L)}$ with the teacher's $\hat{\boldsymbol{z}} = \hat{\boldsymbol{h}}_e^{(L)}$:

$$\mathcal{L}_{\text{emb}} = \text{MSE}(\boldsymbol{z}, \hat{\boldsymbol{z}}) + 1 - \text{Cos}(\boldsymbol{z}, \hat{\boldsymbol{z}}) + \lambda \mathcal{L}_{\text{KL}}. \tag{10}$$

For each reference role $r \in \{\text{q}, \text{c}\}$ (query or candidate), we compute student and teacher cosine similarities to the same frozen teacher references. Softmax with temperature $\tau = 0.05$ gives distributions $p^{(r)}$ and $\hat{p}^{(r)}$, whose forward KL divergences are averaged:

$$\mathcal{L}_{\text{KL}} = \frac{1}{2} \sum_{r \in \{\text{q},\text{c}\}} D_{\text{KL}}\left(\hat{p}^{(r)} \| p^{(r)}\right). \tag{11}$$

Losses are averaged over training examples, with λ controlling the KL weight. Appendix A.3 details reference sampling and loss computation.

4 EXPERIMENTS

4.1 SETUP

Implementation Details. We evaluate our method on two UME models with different MLLM backbones: UniME-v2 Gu et al. (2025b) and Ops-MM-embedding-v1 OpenSearch-AI (2025), built on LLaVA-OneVision-8B Li et al. (2024) and Qwen2-VL-7B Wang et al. (2024), respectively. We adapt the language backbone using LoRA with rank $r = 16$ and scaling parameter $\alpha = 32$, and train all parameters of the lightweight projector. For a fair comparison, we fine-tune all competing methods using a similar Stage 2 training procedure (Section 3.3). For PDrop, we progressively prune visual tokens at layers 2, 6, and 16, retaining 50%, 25%, and 12.5% of the original visual tokens, respectively. This schedule is chosen to bring its FLOPs closer to those of the other competing methods, which retain 20% of the original visual tokens. We measure FLOPs on MMEB evaluation samples and report them in TFLOPs. End-to-end latency in Table 1 is measured on an NVIDIA A100 GPU with a batch size of 8 and reported in milliseconds per sample by dividing the batch latency by 8. Further implementation details are provided in Appendix A.3.

Evaluation Benchmarks and Metrics. We evaluate on MMEB Jiang et al. (2025), covering 36 datasets across classification, VQA, retrieval, and visual grounding. Following the official splits and candidate sets, we report Precision@1 (%), averaged within each task category and across all 36 datasets (Avg.), with equal weight per dataset.

Table 1: **Performance and efficiency on MMEB.** We compare the original encoders, existing visual-token compression methods, and our method on two UME backbones. Task scores are category-level mean Precision@1 (%); Avg. is the unweighted mean across all 36 datasets. Rel. Avg. reports the percentage of overall performance retained relative to the corresponding original encoder. Darker and lighter blue cells indicate the best and second-best values among compressed models within each backbone, respectively; ties share the same highlight.

Method \ Evaluation	MMEB				Avg↑	Rel. Avg. (%)↑	FLOPs (T)↓	Latency (ms)↓
	Cls	VQA	Ret	Grd				
UniME-v2(Gu et al. (2025b))								
Original	65.74	63.36	72.72	89.68	70.06	100	20.67	110.84
FastV(Chen et al. (2024))	63.45	55.58	68.89	80.23	64.94	92.7	4.94	41.73
VisionZip(Yang et al. (2025))	64.80	58.91	70.68	83.75	67.22	95.9	4.29	37.69
DivPrune(Alvar et al. (2025))	62.52	57.88	69.83	84.03	66.06	94.3	4.30	61.26
DART(Wen et al. (2025))	64.41	58.06	69.15	82.55	66.24	94.5	5.55	45.47
PDrop(Xing et al. (2025))	62.04	57.89	65.46	77.83	63.78	91.0	6.66	48.95
Ours	64.42	61.13	70.53	88.60	68.23	97.4	4.67	37.29
Ops-MM-embedding-v1(OpenSearch-AI (2025))								
Original	70.42	68.64	72.38	84.53	72.14	100	5.29	55.18
FastV(Chen et al. (2024))	67.44	58.38	67.93	64.08	64.71	89.7	1.57	35.70
VisionZip(Yang et al. (2025))	65.85	58.30	66.07	72.75	64.59	89.5	1.42	36.61
DivPrune(Alvar et al. (2025))	66.62	61.43	69.28	69.15	66.34	91.9	1.42	38.23
DART(Wen et al. (2025))	68.63	56.71	68.09	68.93	65.17	90.3	1.70	37.68
PDrop(Xing et al. (2025))	66.74	58.78	67.51	64.00	64.48	89.4	1.98	44.39
Ours	66.86	61.86	68.94	80.18	67.64	93.8	1.45	35.07

4.2 MAIN RESULTS

Comparison with Other Methods. We compare ViS^2 with FastV (Chen et al., 2024), VisionZip (Yang et al., 2025), DivPrune (Alvar et al., 2025), DART (Wen et al., 2025), and PDrop (Xing et al., 2025) on both UME backbones, with results reported in Table 1. On UniME-v2, our method retains 97.4% of the original performance, with a 77.4% reduction in FLOPs and a 2.97× speedup. Its average score of 68.23 exceeds VisionZip's by 1.01 points, with slightly higher

Table 2: Effectiveness of the projector and Stage 2. Stage 1 aligns intermediate states; Stage 2 distills final embeddings. All variants retain 20% of the original visual tokens at inference. Bold marks the best score per backbone.

Setting	Cls	VQA	Ret	Grd	Avg
UniME-v2					
Visual Oracle	62.35	58.06	66.71	74.13	63.92
Direct Reuse	54.06	43.46	54.43	63.60	52.30
Stage 1 Only	63.80	60.34	66.09	82.20	65.65
Stage 1 + Stage 2	**64.40**	**61.36**	**70.23**	**88.25**	**68.15**
Ops-MM-embedding-v1					
Visual Oracle	**67.34**	48.84	67.63	65.80	62.13
Direct Reuse	60.65	33.89	58.14	58.88	52.18
Stage 1 Only	66.26	61.06	66.28	74.93	65.78
Stage 1 + Stage 2	66.94	**61.89**	**68.81**	**79.13**	**67.51**

Table 3: Projector architectures.

Architecture	RelMSE↓	Cos.↑
0A+11F	0.3234	0.8218
1A+9F	0.2645	0.8565
2A+7F	0.2477	0.8663
4A+4F	**0.2451**	**0.8678**
5A+2F	0.2500	0.8650
6A+0F	0.2612	0.8586

Table 4: Comparison of attention selection strategies.

Scoring	MMEB↑	Latency↓
All text (Eager)	67.48	13.053
Last Emb (Flash)	67.83	3.620

FLOPs (4.67 vs. 4.29 TFLOPs) but comparable, slightly lower latency (37.29 vs. 37.69 ms). On Ops-MM-embedding-v1, it obtains an average score of 67.64, retaining 93.8% of the original performance while reducing FLOPs by 72.6% and providing a $1.57\times$ speedup. It exceeds DivPrune, the strongest competitor by overall score, by 1.30 points at a similar computational cost (1.45 vs. 1.42 TFLOPs).

Gains in Visual Grounding. The largest gains over the best competing method in each task category occur in visual grounding: our method reaches 88.60 on UniME-v2 and 80.18 on Ops-MM-embedding-v1, exceeding the strongest competing methods on grounding by 4.57 and 7.43 points, respectively. This pattern is consistent with our motivation to preserve the full visual sequence for fusion in deeper layers before selecting tokens.

4.3 Ablation and Analysis

Ablation of the Skip Module. Table 2 evaluates the projector's adaptation of visual features across skipped layers and the additional benefit of Stage 2 training. All variants retain 20% of the original visual tokens after selection at inference; their skipped middle layers process only text and embedding tokens. *Direct Reuse* reinserts the visual states output by the complete layer 0 without cross-layer mapping or additional trainable parameters, whereas *Visual Oracle* injects the original encoder's visual states at the target re-entry layer. *Stage 1 Only* uses intermediate-state alignment, while *Stage 1 + Stage 2* adds final-embedding distillation (Section 3.3).

Stage 1 Only improves the overall score over Direct Reuse by 13.35 points on UniME-v2 and 13.60 points on Ops-MM-embedding-v1, demonstrating the benefit of learned adaptation along the visual bypass. It also exceeds Visual Oracle by 1.73 and 3.65 points, respectively. This gain suggests that Stage 1 benefits from both intermediate-state alignment and the adaptation of post-fusion layers to strengthen $\langle\mathrm{Emb}\rangle$-to-Visual interactions. Visual Oracle injects the teacher's visual states without this joint adaptation. Stage 2 further improves the overall scores by 2.50 and 1.73 points, respectively. These improvements occur across all four task categories, with the largest gains in grounding (+6.05 and +4.20 points). Thus, the projector enables effective visual-state adaptation across skipped layers, while Stage 2 further improves the final embeddings under the same inference token budget.

Projector Architecture. Table 3 compares different ratios of causal attention (A) to SiLU FFN (F) layers under similar latency budgets, including a deeper FFN-only alternative. All configurations use an internal dimension of 2048; one full block is AF, and the 5A+2F ordering is AAFAAAF. The 4A+4F design reduces relative MSE by 24.2% over 0A+11F, highlighting the value of token interactions beyond increased token-wise depth. Across the tested configurations, 4A+4F achieves the lowest relative MSE and highest cosine similarity, supporting our choice of four complete attention–FFN blocks.

Ablation of the Select Module. In Table 4, E2V selection improves mean MMEB across both backbones from 67.48 to 67.83 over all-text scoring. FlashAttention with last-token attention recomputation reduces attention execution and scoring latency from 13.053 to 3.620 ms per batch at $B = 8$, a $3.61\times$ speedup over eager attention.

In Table 5, E2V alone retains 20% of visual tokens; auxiliary variants select 15% by E2V and 5% from the remainder, measured against the original token count. Norm-based selection lowers mean MMEB; uniform and average sampling change it by only -0.01 and $+0.07$ points. We therefore use E2V attention alone by default.

Table 5: Auxiliary selection strategies. "+" uses 15% E2V-selected tokens and 5% auxiliary tokens, versus 20% E2V alone.

Selection strategy	**MMEB↑**
E2V attention	67.83
+ top-k key ℓ_1	67.51
+ top-k hidden state ℓ_1	67.72
+ uniform sampling	67.82
+ average sampling	67.90

Qualitative Analysis. Figure 6 compares token selection on a natural image from Visual7W-Pointing and two DocVQA samples sharing the same document but different queries. For the natural image, ViS^2 better preserves the largest carrot specified by the query. For DocVQA, its retained coverage shifts between the university's location ("Nashville") and the recipient's name ("Prof. W. J. Darby"). VisionZip and DivPrune select identical tokens across the two document queries, consistent with their reliance on visual signals. FastV and DART incorporate textual signals but show weaker query alignment, consistent with limited text–visual interaction before early selection. These examples support selecting tokens after renewed multimodal fusion in deeper layers to better preserve query-relevant regions. Additional qualitative comparisons across object localization, chart understanding, and scene-text recognition are provided in Appendix A.4.3.

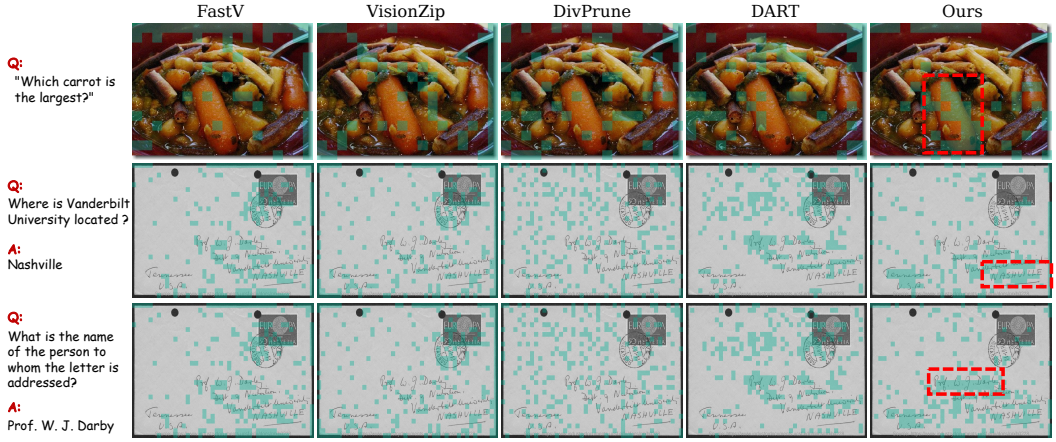


Figure 6: **Query-dependent visual-token selection.** Top: a natural-image query; bottom two rows: different queries for the same DocVQA image. Columns compare methods; red dashed boxes highlight query-relevant regions in the ViS^2 results.

5 Conclusion

Our work demonstrates that preserving visual information for embedding formation need not require full visual computation at every layer. Joint analysis of visual-state evolution and embedding-to-visual interaction reveals similar visual representational structure across shallow-to-middle layers and increasingly task-aligned visual attention in deeper layers. These findings motivate Visual *Skip-then-Select* (ViS^2), which preserves early interaction, bypasses middle-layer visual processing, and restores multimodal fusion before token selection. Across two UME backbones and 36 MMEB datasets, ViS^2 reduces FLOPs by 75.0% while retaining 95.6% of the original performance on average. More broadly, understanding how visual representations evolve and interact with the embedding token offers a perspective on the computational needs of UMEs. We hope our work will inspire the efficient architecture design of MLLM-based UME models in the future.

REFERENCES

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 9392–9401, 2025.

Adrian Bulat, Alberto Baldrati, Ioannis Maniadis Metaxas, Yassine Ouali, and Georgios Tzimiropoulos. VISion On Request: Enhanced VLLM efficiency with sparse, dynamically selected, vision-language interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2603.23495.

Jiawei Cao, Junyi Feng, Jiashen Hua, Ziheng Huang, Bing Deng, Kaijie Wu, Chaochen Gu, and Jieping Ye. Illuminating visual identity in universal multimodal embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8737–8748, 2026.

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

Moulik Choraria, Xinbo Wu, Akhil Bhimaraju, Nitesh Sekhar, Yue Wu, Xu Zhang, Prateek Singhal, and Lav R Varshney. Deepinsert: Early layer bypass for efficient and performant multimodal understanding. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7059–7079, 2026. doi: 10.18653/v1/2026.eacl-long.332. URL https://aclanthology.org/2026.eacl-long.332/.

Xuanming Cui, Shlok Kumar Mishra, Wentao Bao, Aashu Singh, Zihao Wang, Xiangjun Fan, Jun Xiao, Ser-Nam Lim, and Jianpeng Cheng. MoEMB: Scaling universal multimodal embeddings with efficient mixture-of-experts models. *arXiv preprint arXiv:2609.08663*, 2026. URL https://arxiv.org/abs/2609.08663.

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, 2025a.

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. *arXiv preprint arXiv:2510.13515*, 2025b.

Florian Hönicke, Michael Günther, Andreas Koukounas, Mohammad Kalim Akram, Scott Martens, Saba Sturua, and Han Xiao. jina-embeddings-v5-omni: Geometry-preserving embeddings via locked aligned towers. *arXiv preprint arXiv:2605.08384*, 2026.

Ting Jiang, Minghui Song, Zihan Zhang, Haizhen Huang, Weiwei Deng, Feng Sun, Qi Zhang, Deqing Wang, and Fuzhen Zhuang. E5-V: Universal embeddings with multimodal large language models. *arXiv preprint arXiv:2407.12580*, 2024. URL https://arxiv.org/abs/2407.12580.

Ziyan Jiang, Rui Meng, Xinyi Yang, Semih Yavuz, Yingbo Zhou, and Wenhu Chen. VLM2Vec: Training vision-language models for massive multimodal embedding tasks. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=zW1kuJQZ2G.

Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. See what you are told: Visual attention sink in large multimodal models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/da8a39bc39ae1c89dd6ebb1e3bcbb3f3-Abstract-Conference.html.

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

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

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pp. 3519–3529. PMLR, 2019.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

Da Li, Yuxiao Luo, Keping Bi, Jiafeng Guo, Wei Yuan, Biao Yang, Yan Wang, Fan Yang, Tingting Gao, and Guorui Zhou. Compressing then matching: An efficient pre-training paradigm for multimodal embedding. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3707–3718, 2026a. doi: 10.18653/v1/2026.acl-long.169. URL https://aclanthology.org/2026.acl-long.169/.

Mingxin Li, Yanzhao Zhang, Dingkun Long, Keqin Chen, Sibo Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Qwen3-VL-Embedding and Qwen3-VL-Reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. *arXiv preprint arXiv:2601.04720*, 2026b. URL https://arxiv.org/abs/2601.04720.

Qi Li, Yanzhe Zhao, Yongxin Zhou, Yameng Wang, Yandong Yang, Yuanjia Zhou, Jue Wang, Zuojian Wang, and Jinxiang Liu. Magic-MM-Embedding: Towards visual-token-efficient universal multimodal embedding with MLLMs. *arXiv preprint arXiv:2602.05275*, 2026c. URL https://arxiv.org/abs/2602.05275.

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.

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

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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 4015–4025, 2025.

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. VLM2Vec-V2: Advancing multimodal embedding for videos, images, and visual documents. *arXiv preprint arXiv:2507.04590*, 2025. URL https://arxiv.org/abs/2507.04590.

OpenSearch-AI. Ops-MM-embedding-v1-7B. Hugging Face model repository, 2025. URL https://huggingface.co/OpenSearch-AI/Ops-MM-embedding-v1-7B. Accessed: 2026-09-22.

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.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. LLaVA-PruMerge: Adaptive token reduction for efficient large multimodal models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22857–22867, 2025. URL https://arxiv.org/abs/2403.15388.

Changli Tang, Qinfan Xiao, Ke Mei, Tianyi Wang, Fengyun Rao, and Chao Zhang. WAVE: Learning unified & versatile audio-visual embeddings with multimodal LLM. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=MiV3WXDYJb.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Zichen Wen, Yifeng Gao, Shaobo Wang, Junyuan Zhang, Qintong Zhang, Weijia Li, Conghui He, and Linfeng Zhang. Stop looking for "important tokens" in multimodal language models: Duplication matters more. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 9972–9991, 2025.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. Conical visual concentration for efficient large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/papers/Xing_Conical_Visual_Concentration_for_Efficient_Large_Vision-Language_Models_CVPR_2025_paper.pdf.

Cheng-Yu Yang, Shao-Yuan Lo, and Yu-Lun Liu. Reroute, don't remove: Recoverable visual token routing for vision-language models. *arXiv preprint arXiv:2606.12412*, 2026.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19792–19802, 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.

Xin Zhang, Ziqi Dai, Yongqi Li, Yanzhao Zhang, Dingkun Long, Pengjun Xie, Meishan Zhang, Jun Yu, Wenjie Li, and Min Zhang. Towards text-image interleaved retrieval. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4254–4269. Association for Computational Linguistics, 2025a. doi: 10.18653/v1/2025.acl-long.214. URL https://aclanthology.org/2025.acl-long.214/.

Xin Zhang, Yanzhao Zhang, Wen Xie, Mingxin Li, Ziqi Dai, Dingkun Long, Pengjun Xie, Meishan Zhang, Wenjie Li, and Min Zhang. Bridging modalities: Improving universal multimodal retrieval by multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025b. URL https://arxiv.org/abs/2412.16855.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. In *International Conference on Machine Learning*, pp. 74840–74857. PMLR, 2025c.

A Appendix

A.1 Representation Similarity Metrics

We define the two similarity measures used in Findings 1 and 2 of Section 3.1. For two states a and b, let $X \in \mathbb{R}^{n \times d}$ and $Y \in \mathbb{R}^{n \times d}$ contain the corresponding hidden representations as rows, where n is the number of observations and d is the hidden dimension. For visual representations, each observation is a cached visual token, matched by sample and token position across states; for $\langle \mathrm{Emb} \rangle$, each observation is the embedding token of one cached sample. The states are $A_0, F_0, \ldots, A_{27}, F_{27}, \mathrm{Out}$, where A_ℓ and F_ℓ denote residual-stream states before the attention and FFN input normalization, respectively, and Out denotes the last decoder-block output before terminal RMSNorm.

Centered kernel alignment (CKA). Following Kornblith et al. (2019), we use linear CKA. We first center each feature across observations:

$$C_n = I_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top, \qquad \widetilde{X} = C_n X, \qquad \widetilde{Y} = C_n Y, \tag{12}$$

where I_n is the identity matrix and $\mathbf{1}_n$ is the all-ones vector. Linear CKA is then

$$\mathrm{CKA}(X, Y) = \frac{\|\widetilde{X}^\top \widetilde{Y}\|_F^2}{\|\widetilde{X}^\top \widetilde{X}\|_F \, \|\widetilde{Y}^\top \widetilde{Y}\|_F}, \tag{13}$$

where $\|\cdot\|_F$ is the Frobenius norm. Equivalently, for the centered Gram matrices $K = \widetilde{X}\widetilde{X}^\top$ and $G = \widetilde{Y}\widetilde{Y}^\top$,

$$\mathrm{CKA}(X, Y) = \frac{\langle K, G \rangle_F}{\|K\|_F \|G\|_F}, \qquad \langle K, G \rangle_F = \mathrm{tr}(K^\top G). \tag{14}$$

Thus, CKA compares the pairwise similarity structure among observations. It lies in $[0, 1]$ when the denominator is nonzero, with larger values indicating greater structural similarity. It is invariant to independent orthogonal transformations and nonzero uniform rescaling of either representation, but not to arbitrary invertible linear transformations. Consequently, high CKA does not require corresponding token vectors to point in the same direction.

Cosine similarity. For corresponding nonzero token vectors $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ (the transposes of row i of X and Y), cosine similarity and its reported mean are

$$\cos(\boldsymbol{x}_i, \boldsymbol{y}_i) = \frac{\boldsymbol{x}_i^\top \boldsymbol{y}_i}{\|\boldsymbol{x}_i\|_2 \|\boldsymbol{y}_i\|_2}, \qquad \mathrm{Cos}(X, Y) = \frac{1}{n}\sum_{i=1}^{n} \cos(\boldsymbol{x}_i, \boldsymbol{y}_i). \tag{15}$$

We compute cosine on uncentered states and average the token-wise scores without pooling token vectors first. Each score lies in $[-1, 1]$: 1 denotes the same direction, 0 orthogonality, and -1 opposite directions. Cosine ignores vector magnitude and measures directional alignment in the shared hidden coordinate system; independently rotating one state's representations can change this alignment.

Aggregation and interpretation in the findings. Both metrics are computed separately within each dataset and input role (query or candidate), using all cached tokens of the relevant type. We average the available role-specific scores within each dataset, then average datasets with equal weight for the heatmaps. In Finding 1, high CKA together with decreasing cosine indicates that visual representations retain similar relational structure while their token directions change, motivating a learned projector across layers. These similarities alone do not establish projector accuracy or preservation of retrieval quality. In Finding 2, the same metrics characterize structural and directional changes in the embedding token across depth. Its initial state A_0 is constant across samples, so centering yields a zero matrix and CKA involving A_0 is undefined, including self-comparison. These entries are shown in gray in Figure 3(a), rather than assigned zero; cosine remains defined because the uncentered initial token is nonzero.

A.2 Additional Analysis

Figure 7 extends the visual-state and attention analysis to four datasets on both Ops-MM-embedding-v1 and UniME-v2. The examples show high visual CKA across shallow-to-middle layers and reduced E2V attention in the middle layers, with the strength of the late-layer attention increase varying across datasets and models.

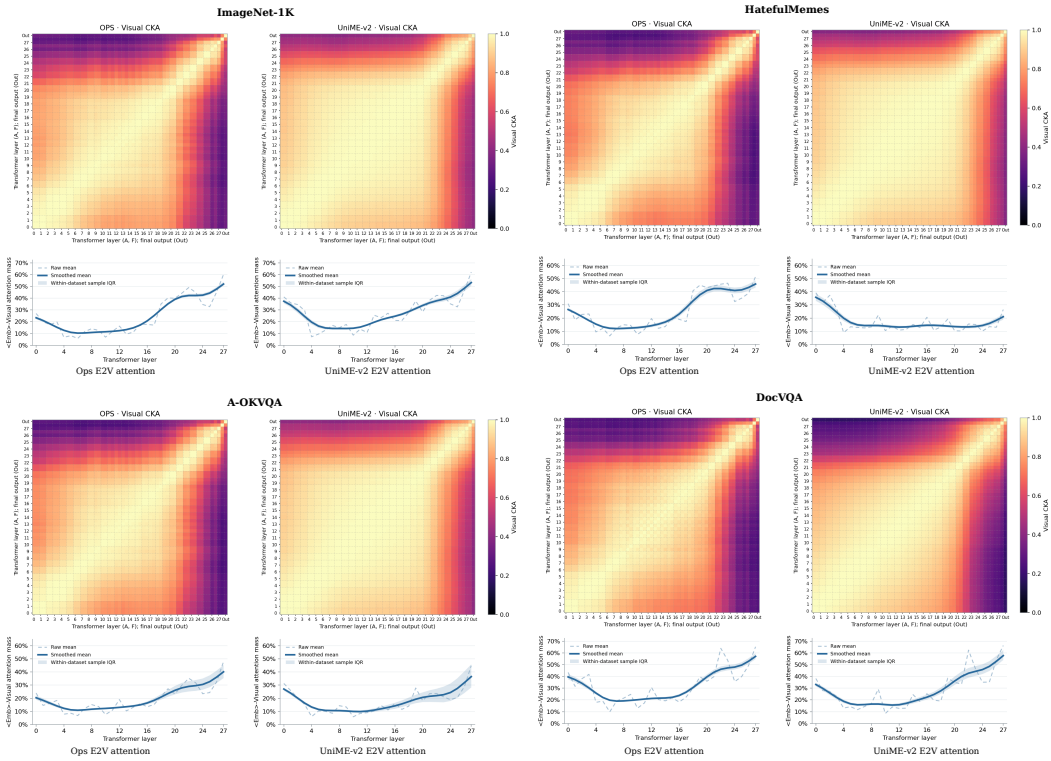


Figure 7: **Visual representation similarity and E2V attention across datasets and models.** Each dataset panel compares Ops-MM-embedding-v1 and UniME-v2 on ImageNet-1K, HatefulMemes, A-OKVQA, or DocVQA. Heatmaps show linear CKA between visual hidden states; curves show head-averaged E2V attention summed over visual tokens. Dashed and solid lines denote raw and smoothed means, respectively; shading indicates the within-dataset sample interquartile range.

A.3 Implementation Details

For Ops-MM-embedding-v1, we use dynamic image resolution, with the number of visual tokens ranging from 4 to 1,280. For UniME-v2, we use a fixed image resolution of 384×384 and a fixed sequence of 1,485 visual tokens.

We follow the two-stage training procedure described in Section 3.3. Stage 1 runs for 4,000 steps with an initial learning rate of 1×10^{-4}, followed by Stage 2 for 3,000 steps with an initial learning rate of 5×10^{-5}. We use a batch size of 64 and a cosine learning-rate schedule in both stages. We train all parameters of the visual projector and adapt the LLM using LoRA with rank $r = 16$ and scaling parameter $\alpha = 32$.

For both models, the projector takes all visual states from the residual stream after the attention sublayer in layer 0. The projected visual tokens are reintroduced at the input to layer 16 for Ops-MM-embedding-v1 and layer 15 for UniME-v2. Table 6 summarizes the architecture and hyperparameters of the visual projector.

KL distillation: reference construction. The KL term transfers the teacher's relative similarities to reference examples, complementing the pointwise MSE and cosine losses. We precompute a

Table 6: Implementation details of the visual projector.

Parameter	Configuration
Input / output dimension	3584 / 3584
Internal hidden dimension	2048
Number of Transformer blocks	4
Block composition	Attention + SwiGLU
Normalization placement	Pre-norm
Normalization	RMSNorm ($\epsilon = 10^{-6}$)
Attention type	Causal multi-head attention
Number of attention heads	32
Attention head dimension	64
FFN intermediate dimension	2944
Fused gate/value dimension	5888
Input projection activation	SiLU
M-RoPE frequency-pair split (T/H/W)	8 / 12 / 12
Total parameters	154.20M

frozen bank of teacher embeddings from training inputs, with separate query and candidate pools. For each training input, we select four disjoint groups of 256 references: nearby queries, random queries, nearby candidates, and random candidates. Nearby references are taken from precomputed neighbor lists ranked by teacher cosine similarity, while random references are sampled uniformly from the corresponding role pool. We exclude the input itself and all known positives from ordinary neighbor and random sampling. One known positive is then included in the nearby group of the opposite role, replacing one neighbor so that this group contains 255 neighbors and one positive. Thus, each input has 1,024 references in total, comprising 512 queries and 512 candidates. Selection is deterministic for each input using seed 1422 and its bank index; student and teacher use exactly the same references. The reference bank contains only training inputs, and the selected references receive soft targets from the teacher rather than hard negative labels.

Similarity distributions. Let $\boldsymbol{z}_i$ and $\hat{\boldsymbol{z}}_i$ denote the student and teacher embeddings of training input i. For role $r \in \{\mathrm{q}, \mathrm{c}\}$, let $\{\hat{\mathbf{g}}_{ij}^{(r)}\}_{j=1}^{M}$ be its selected teacher references, where $M = 512$. We ℓ_2-normalize the embeddings and compute temperature-scaled cosine logits:

$$a_{ij}^{(r)} = \frac{1}{\tau} \frac{\boldsymbol{z}_i^\top \hat{\mathbf{g}}_{ij}^{(r)}}{\|\boldsymbol{z}_i\|_2 \|\hat{\mathbf{g}}_{ij}^{(r)}\|_2}, \qquad \hat{a}_{ij}^{(r)} = \frac{1}{\tau} \frac{\hat{\boldsymbol{z}}_i^\top \hat{\mathbf{g}}_{ij}^{(r)}}{\|\hat{\boldsymbol{z}}_i\|_2 \|\hat{\mathbf{g}}_{ij}^{(r)}\|_2}, \tag{16}$$

with $\tau = 0.05$. We apply softmax separately within each role:

$$p_{ij}^{(r)} = \frac{\exp(a_{ij}^{(r)})}{\sum_{k=1}^{M} \exp(a_{ik}^{(r)})}, \qquad \hat{p}_{ij}^{(r)} = \frac{\exp(\hat{a}_{ij}^{(r)})}{\sum_{k=1}^{M} \exp(\hat{a}_{ik}^{(r)})}. \tag{17}$$

Each distribution therefore sums to one over its 512 references. The nearby and random groups of the same role share one softmax; query and candidate references have separate normalizations.

Loss reduction and optimization. We compute forward KL from the teacher distribution to the student distribution, averaging over both reference roles and the B training inputs:

$$\mathcal{L}_{\mathrm{KL}} = \frac{1}{2B} \sum_{i=1}^{B} \sum_{r \in \{\mathrm{q},\mathrm{c}\}} \sum_{j=1}^{M} \hat{p}_{ij}^{(r)} \left(\log \hat{p}_{ij}^{(r)} - \log p_{ij}^{(r)} \right). \tag{18}$$

This gives the query and candidate reference distributions equal weight, encouraging the student to preserve the teacher's relative similarity patterns within each role. We compute normalization, similarities, and log-softmax in FP32 even when the model forward pass uses BF16. The implementation sums over references and averages over inputs for each role, then averages the two role losses. Teacher embeddings, reference embeddings, and teacher probabilities are detached; gradients flow only through the student embeddings. The training objective adds $\lambda \mathcal{L}_{\mathrm{KL}}$ to the MSE and cosine terms without an additional τ^2 multiplier. The default KL weight is $\lambda = 1$; Appendix A.4.2 compares weights of 1, 5, and 10.

Table 7: **Inference latency across batch sizes.** Amortized time includes partial final batches, whereas batch latency averages only full batches. Repeated Original rows report separate measurement runs. Bold values mark the lowest measured times within each backbone.

Method	Amortized time (ms/input)↓			Batch latency (ms/batch)↓		
	$B = 1$	$B = 4$	$B = 8$	$B = 1$	$B = 4$	$B = 8$
UniME-v2						
Original	127.97	113.73	110.84	127.97	454.56	885.55
Original	128.05	113.99	110.61	128.05	455.61	883.64
FastV	79.82	44.28	41.73	79.82	176.46	332.28
VisionZip	78.80	40.36	37.69	78.80	160.71	299.79
DART	84.24	48.22	45.47	84.24	192.18	362.04
DivPrune	100.95	63.96	61.26	100.95	255.11	488.42
PDrop	85.92	52.33	48.95	85.92	208.66	389.90
Original	129.30	114.16	111.15	129.30	456.30	887.98
Ours	**57.88**	**39.54**	**37.29**	**57.88**	**157.80**	**297.26**
Ops-MM-embedding-v1						
Original	77.38	58.12	55.18	77.38	231.94	440.24
Original	77.47	57.99	54.97	77.47	231.43	438.53
FastV	75.30	40.21	35.70	75.30	160.15	283.74
VisionZip	68.41	41.25	36.61	68.41	164.31	291.03
DART	77.25	41.93	37.68	77.25	167.02	299.59
DivPrune	69.37	42.76	38.23	69.37	170.34	303.96
PDrop	79.95	47.26	44.39	79.95	188.41	353.76
Original	78.20	57.82	54.83	78.20	230.74	437.38
Ours	**66.56**	**38.95**	**35.07**	**66.56**	**155.16**	**278.97**

A.4 Extended Experiments

A.4.1 Inference Latency across Batch Sizes

We compare inference latency at batch sizes of 1, 4, and 8 on Ops-MM-embedding-v1 and UniME-v2. LoRA weights are merged for inference, and the trained visual projector is included in all timings.

Measurement protocol. Each forward pass runs on a single NVIDIA A100-SXM4-80GB GPU in BF16 and evaluation/inference mode, without compilation or CUDA graphs. We use the same 4,306 image-containing inputs for every method and batch size: 1,500 queries and 2,806 candidates, obtained by sampling 5% of the MMEB test inputs within each dataset/role group with seed 422. Text-only inputs are excluded. UniME-v2 uses 384×384 images and a maximum sequence length of 4,096; Ops-MM-embedding-v1 uses dynamic resolution with a maximum of 1,003,520 pixels and a maximum sequence length of 2,048. We measure synchronized wall-clock time after 20 initial warm-up passes, with one additional warm-up and three timed repetitions per batch. Timing starts from inputs already on the GPU and ends at the final normalized embedding, including the vision encoder, multimodal connector, visual projector where applicable, LLM, token selection, pooling, and normalization. Image loading, preprocessing, tokenization, initial host-to-device transfer, model loading, and retrieval are excluded.

Compared configurations. FastV, VisionZip, DART, and DivPrune retain 20% of the visual tokens; PDrop progressively retains 50%, 25%, and 12.5% after blocks 2, 6, and 16, respectively (counting blocks from one). FastV and PDrop use eager attention with scores averaged over valid text queries. These five baselines apply compression to the original model weights without fine-tuning in this latency experiment. Our method restores visual tokens at layer 16 for Ops-MM-embedding-v1 and layer 15 for UniME-v2, then selects 20% using $\langle \mathrm{Emb} \rangle$-to-Visual attention at layers 19 and 18, respectively (zero-based layer indices).

Results. As shown in Table 7, our method achieves the lowest measured amortized time per input and full-batch latency among all compared methods on both backbones at every tested batch size.

A.4.2 Effect of Distillation Losses

We compare five final-embedding distillation objectives on Ops-MM-embedding-v1: cosine alone, cosine with MSE, and cosine with MSE and KL at weights $\lambda \in \{1, 5, 10\}$ (Section 3.3). The cosine and MSE terms each have unit weight when included. All variants start from the same Stage 1 checkpoint and use seed 422, 3,000 training steps, and an effective batch size of 64, with the same training retention-rate schedule (0.1–0.8, $\gamma = 1.5$). We evaluate on all 36 MMEB datasets with a fixed visual-token retention rate of 20%.

Table 8: Comparison of distillation losses. Following Table 2, we report category-level mean Precision@1 (%) and the unweighted mean over all 36 MMEB datasets (Avg.). KL weights are indicated by λ. Bold values mark the best results among the five compared configurations.

Setting	Cls	VQA	Ret	Grd	Avg
Ops-MM-embedding-v1					
Cosine	66.64	61.77	68.58	78.55	67.26
Cosine + MSE	66.68	61.88	68.44	78.78	67.28
Cosine + MSE + KL ($\lambda = 1$)	66.90	61.82	68.66	79.43	67.47
Cosine + MSE + KL ($\lambda = 5$)	67.18	62.05	**68.88**	79.80	67.72
Cosine + MSE + KL ($\lambda = 10$)	**67.26**	**62.24**	68.81	**79.83**	**67.78**

Table 8 shows that adding KL improves the average score from 67.28% to 67.47%. Increasing its weight to $\lambda = 10$ yields the highest average score among these five configurations, reaching 67.78%.

A.4.3 QUALITATIVE RESULTS

Figure 8 provides additional visual-token selection examples across object localization, chart understanding, and scene-text recognition. The paired queries for the same chart and natural image illustrate how our retained visual coverage changes with the requested information.

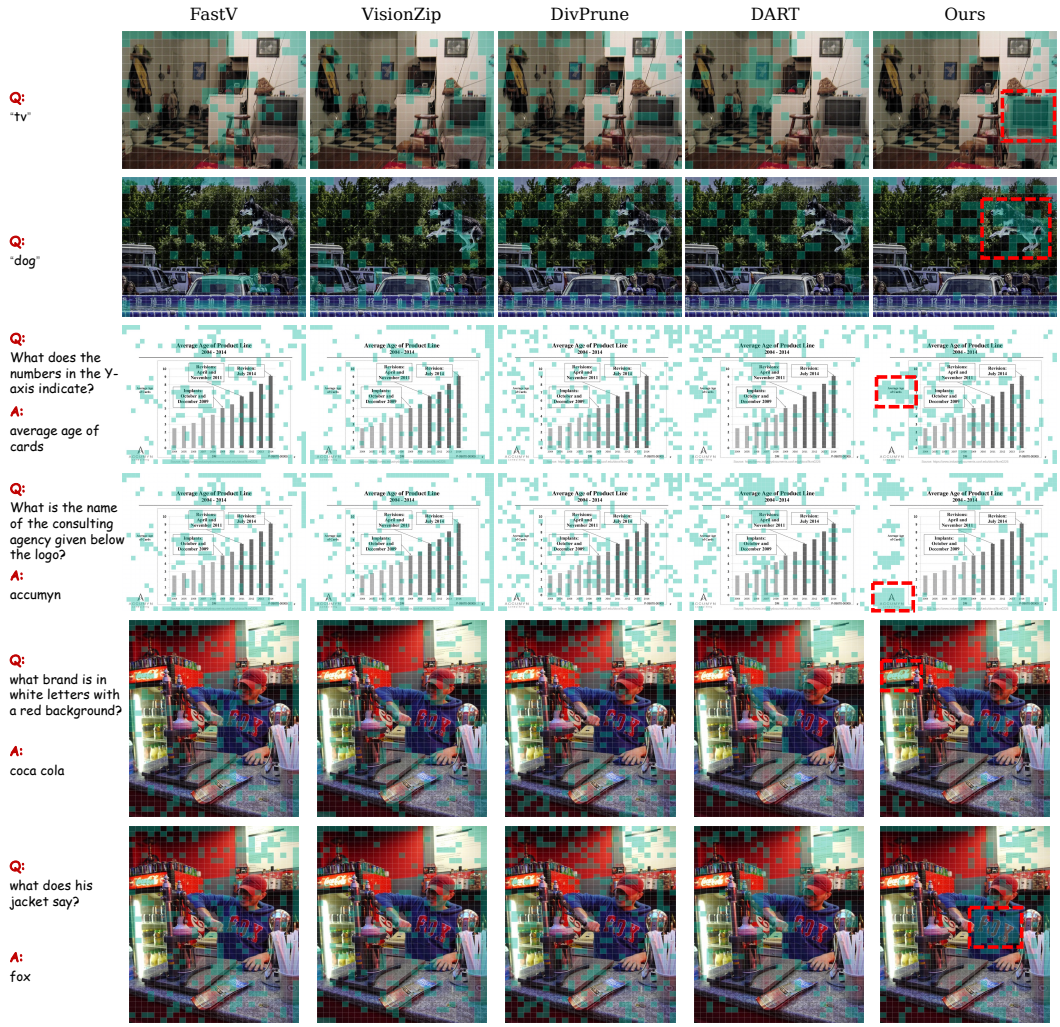


Figure 8: **Additional examples of query-dependent visual-token selection.** Columns compare FastV, VisionZip, DivPrune, DART, and our method. The top two rows show object-localization queries; the middle and bottom pairs each use the same image with different queries. Queries and available answers are shown on the left, and red dashed boxes highlight query-relevant regions in our results.