ALLOCATE BEFORE YOU EMBED: ADAPTIVE VISUAL INPUT ALLOCATION FOR VIDEO EMBEDDINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large-scale video retrieval requires embedding models to encode long and diverse videos under tight visual-input and inference budgets. Existing methods typically sample a small, fixed set of frames at their original resolution, limiting temporal coverage and ignoring frame importance. Our empirical analysis shows that expanding temporal coverage improves retrieval even under a fixed visual-input budget. Gains are larger when the original per-frame resolution is preserved, highlighting the complementary roles of temporal coverage and spatial fidelity. Motivated by this finding, we propose AllocEmbed, an allocate-then-embed framework that reallocates a fixed visual-input budget across more frames. A lightweight allocator uses low-cost previews to assign frame-wise resolutions before the embedding backbone, preserving more detail where it most benefits retrieval while reducing visual cost elsewhere. We further introduce Retrieval-Driven Policy Optimization (RDPO), which learns the allocator directly from retrieval feedback using a rank-validated similarity gap and a confidence-guided efficiency incentive. Operating entirely before the backbone, AllocEmbed integrates with existing retrieval systems without modifying the embedding model or downstream pipeline. Experiments on the MMEB-V2 V-QA and V-RET tasks and our LongRet benchmark show that AllocEmbed achieves the best overall retrieval performance among the evaluated budget-matched methods and transfers across embedding backbones. Code is included in the supplementary material.

1 INTRODUCTION

Video embeddings provide a common interface for large-scale video retrieval, cross-modal search, and retrieval-augmented video understanding. By compressing a variable-length video into a fixed-dimensional vector, they make it possible to compare video content efficiently at corpus scale. In practice, however, video embedding models operate under limited visual context, and running large models over massive video collections is expensive. Existing systems therefore commonly control cost by uniformly sampling a small, fixed number of frames from each video (Luo et al., 2022; Ma et al., 2022; Meng et al., 2025). This strategy is simple and cost-effective, but it also assumes that different videos, time positions, and visual evidence carry the same value for retrieval.

Figure 1 illustrates our motivation. We first examine how temporal coverage affects video retrieval. Across four embedding backbones, sampling more frames improves retrieval when per-frame resolution is preserved. Under a fixed visual-input budget, denser sampling remains beneficial despite the resulting lower per-frame resolution, but yields smaller gains. Together, these results show that temporal coverage and spatial fidelity provide complementary benefits and motivate a central question: how should a fixed visual-input budget be allocated across a denser frame sequence?

We therefore propose AllocEmbed, an allocate-then-embed framework that redistributes a target visual-input budget across a denser frame sequence. Given low-cost previews of the sampled frames and the task text, a lightweight allocator predicts a continuous resolution scale for each frame and constructs a mixed-resolution video input. All sampled frames remain in their original temporal order, while their resolutions are adjusted to preserve more detail where it most benefits retrieval and reduce visual cost elsewhere. The allocated input is then processed by a frozen embedding

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

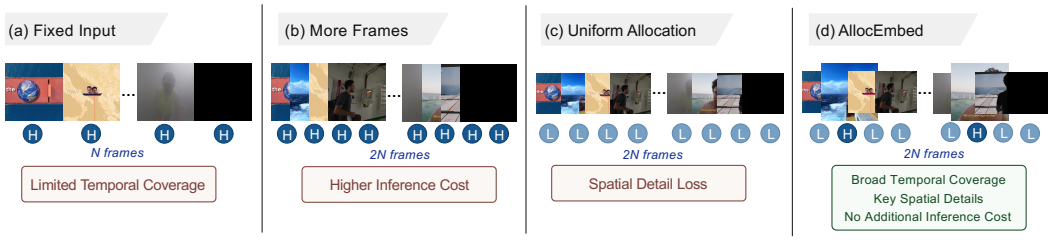


Figure 1: Fixed-frame inputs limit temporal coverage. Adding full-resolution frames increases inference cost, whereas uniform downscaling loses spatial detail. AllocEmbed preserves broad temporal coverage and key spatial detail under the same visual-input budget. H/L denote high/low resolution.

backbone. Because allocation occurs entirely before embedding, AllocEmbed requires no changes to the backbone weights, similarity function, or downstream retrieval pipeline.

Learning such an allocator is challenging because existing datasets provide no frame-level supervision for resolution, and an allocation can be evaluated only after preprocessing, embedding, and retrieval. We address this problem with Retrieval-Driven Policy Optimization (RDPO), which learns directly from end-to-end retrieval outcomes. RDPO samples multiple allocations for the same input and evaluates them against an augmented candidate set with in-batch and global negatives. It then constructs a group-relative learning signal from a rank-validated similarity gap, which grades only allocations that rank the positive target first. A success-gated, confidence-guided efficiency incentive further favors lower-cost allocations without rewarding cheap but incorrect decisions. This formulation trains the allocator through the actual visual preprocessing and tokenization pipeline while keeping the embedding backbone frozen.

We evaluate AllocEmbed on ten MMEB-V2 video QA and retrieval tasks and on LongRet, a new benchmark for long-video retrieval. The allocator is trained with VLM2Vec-V2-2B and evaluated without retraining on three additional embedding backbones, covering model sizes from 2B to 8B parameters. Averaged across the ten MMEB-V2 tasks, AllocEmbed improves Hit@1 by 1.34–1.69 points over the corresponding fixed-frame baselines. Averaged across the four LongRet subsets, the gains increase to 5.69–9.09 points. These aggregate improvements are achieved without exceeding the visual-input cost of each backbone's fixed-frame baseline.

Our contributions are threefold:

I. We introduce **AllocEmbed**, an allocate-then-embed framework motivated by our finding that temporal coverage and spatial fidelity provide complementary benefits for video retrieval. To our knowledge, AllocEmbed is the first framework to study visual input allocation for VLM-based video embedding models.

II. We develop RDPO, a retrieval-driven policy optimization method that learns the allocation policy directly from ranking feedback using a rank-validated similarity gap and a confidence-guided efficiency incentive.

III. We conduct extensive experiments on MMEB-V2 and our newly constructed LongRet benchmark. AllocEmbed improves aggregate retrieval performance under matched visual-input budgets and transfers from its training backbone to three additional embedding backbones without retraining.

2 Related Work

Video and Multimodal Embedding Models. Video embedding models enable cross-modal retrieval by mapping videos and natural-language descriptions into a shared representation space. Early approaches either pretrain dedicated video–text encoders, such as VideoCLIP and Frozen in Time (Xu et al., 2021; Bain et al., 2021), or adapt image–text models to video through frame-level aggregation, as in CLIP4Clip and X-CLIP (Luo et al., 2022; Ma et al., 2022). Recent VLM-based embedding models broaden this scope to heterogeneous queries and targets. VLM2Vec supports diverse image–text tasks, while GME unifies retrieval across text, images, composed image–text inputs, and visual documents (Jiang et al., 2024; Zhang et al., 2024). VLM2Vec-V2 further in-

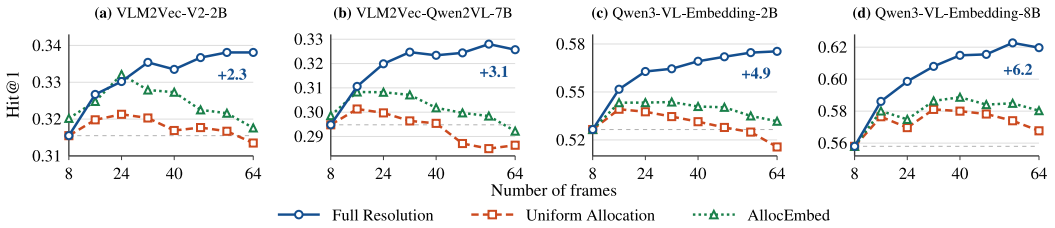


Figure 2: Effect of increasing the sampled frame count on mean Hit@1 across ten MMEB-V2 V-QA and V-RET tasks. Full Resolution keeps the original per-frame resolution as the frame count increases, whereas Uniform Allocation and AllocEmbed use the same visual-input budget as the 8-frame input. For Full Resolution, blue labels report the increase in mean Hit@1 from 8 to 64 frames, measured in percentage points.

corporates videos and visual documents and introduces the MMEB-V2 benchmark (Meng et al., 2025), while Qwen3-VL-Embedding encodes heterogeneous inputs, including videos, in a single model (Li et al., 2026). However, videos are still typically represented by a fixed number of frames at a uniform resolution. The trade-off between temporal coverage and per-frame spatial detail under a constrained visual-input budget therefore remains underexplored.

Efficient Video Processing and Budget Allocation. Efficient video processing typically reduces either the number of input frames or the visual representations processed by the model. Frame-selection methods retain sparse salient or query-relevant frames (Buch et al., 2025), while token merging and pruning compress visual representations within VLMs (Ryoo et al., 2024; Shen et al., 2024; 2026). Other work adapts input resolution to video content for efficient action recognition and query-aware video understanding (Meng et al., 2020; Zhang et al., 2025). Most closely related, ResAdapt learns input-side resolution assignments from generated-response quality without frame-level resolution labels while preserving the backbone architecture (Liao et al., 2026). Efficient video retrieval methods instead rely mainly on CLIP-style encoders and reduce cost through frame selection or frame-level aggregation (Wu et al., 2023; Hu et al., 2023; Shen et al., 2025). Building on ResAdapt's input-side allocation design, AllocEmbed learns per-frame resolutions for VLM-based video embedding models from the positive target's rank and separation from competing candidates.

3 Temporal Coverage and Spatial Fidelity

To study the interplay between temporal coverage and spatial fidelity in video retrieval, we vary the sampled frame count under two token budget regimes. Here, input cost is the number of tokens fed to the embedding backbone after preprocessing. Full Resolution preserves the original per-frame resolution, allowing the token count to grow with the number of frames. Under the fixed-budget regime, Uniform Allocation uniformly resizes frames to match the token cost of the 8-frame baseline, while AllocEmbed adaptively allocates resolution across frames under the same budget. Within each comparison, the embedding backbone, dataset, evaluation metric, and temporal sampling rule remain unchanged. Figure 2 reports mean Hit@1 across ten MMEB-V2 video question answering (V-QA) and video retrieval (V-RET) tasks for four backbones spanning model families and sizes.

Observation 1: More frames help when per-frame resolution is preserved. Across all four backbones, denser sampling yields overall gains when the original per-frame resolution is preserved. This suggests that fixed-frame inputs miss useful temporal evidence. Sampling more frames helps capture brief or unevenly distributed events, while preserving resolution retains their visual cues.

Observation 2: More frames also help under a fixed visual budget. Under a fixed visual-input budget, moderately increasing the number of uniformly resized frames improves retrieval across all four backbones. The added temporal evidence can compensate for lower per-frame resolution. This benefit diminishes as further downscaling causes spatial-detail loss to outweigh broader temporal coverage.

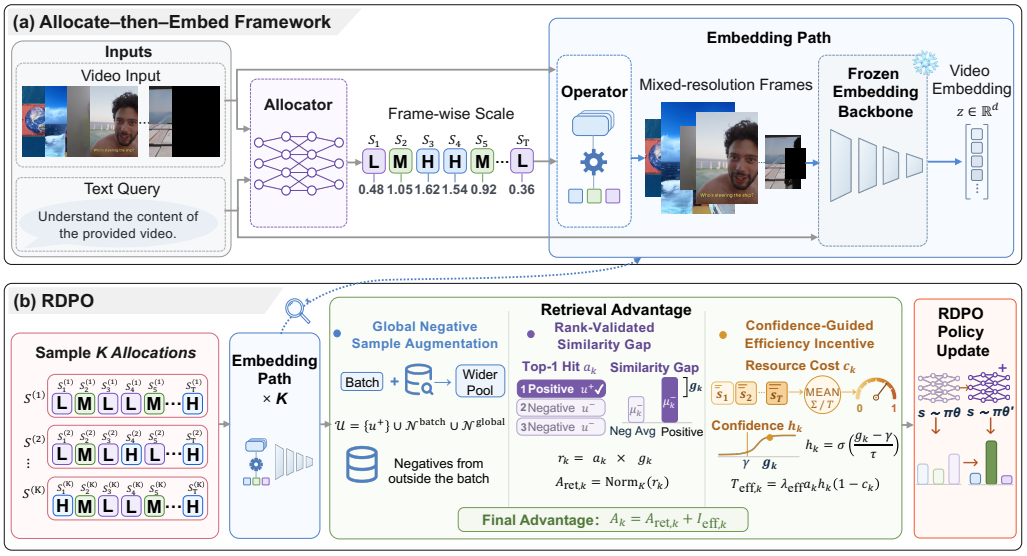


Figure 3: Overview of AllocEmbed. (a) The allocator predicts per-frame resolution scales from the video and task text, and a scale-aware operator constructs a mixed-resolution input for the frozen embedding backbone. (b) RDPO evaluates multiple allocations against a candidate set augmented with global negatives, then updates the allocator using a group-relative retrieval advantage based on the rank-validated similarity gap and a confidence-guided efficiency incentive.

Together, these findings show that temporal coverage and spatial fidelity must be balanced under a fixed visual-input budget. AllocEmbed therefore retains a denser frame sequence while allocating resolution non-uniformly, avoiding the uniform loss of spatial detail imposed by global downscaling.

4 AllocEmbed

4.1 Problem Formulation

Let V denote a video and q the task-related text input, such as a retrieval instruction or question. The embedding model must preserve retrieval-relevant visual evidence under a limited visual-input budget. Standard pipelines control cost by sampling a small, fixed number of frames at a common resolution, but their restricted temporal coverage can miss brief or sparsely distributed evidence.

We instead sample a denser candidate sequence $X = (x_1, \ldots, x_T)$, where T exceeds the standard frame count. Because the budget prevents all T frames from being encoded at high resolution, we introduce a frame-wise allocation vector $\mathbf{s} = (s_1, \ldots, s_T)$, where s_t controls the spatial information retained from x_t. Let $U(\mathbf{s})$ denote the retrieval utility and $C(\mathbf{s})$ the number of tokens fed to the embedding backbone after preprocessing. Given a token budget B, the desired allocation is

$$\mathbf{s}^\star = \arg\max_{\mathbf{s}:C(\mathbf{s})\le B} U(\mathbf{s}). \tag{1}$$

AllocEmbed learns a content- and text-conditioned allocation that redistributes resources across the sampled frames, expanding temporal coverage without uniformly sacrificing the spatial detail required for accurate ranking.

4.2 Allocate-Then-Embed Framework

As illustrated in Figure 3(a), AllocEmbed follows an allocate-then-embed design. Given a densely sampled candidate sequence X and task text q, it first uses a lightweight allocator to predict a spatial scale for each candidate frame:

$$\mathbf{s} = \pi_\theta(X, q) = (s_1, \ldots, s_T). \tag{2}$$

Each s_t determines the input resolution of frame x_t. The scale-aware operator then resizes the frames according to their assigned scales, producing a mixed-resolution sequence. This sequence

and q are passed to the video embedding backbone to obtain the final embedding. AllocEmbed improves retrieval through more effective use of the same visual-input budget.

4.2.1 ALLOCATOR ARCHITECTURE

Our allocator builds on ResAdapt's architecture and Beta scale policy (Liao et al., 2026). A frozen SmolVLM2-256M feature extractor (Marafioti et al., 2025) first obtains patch-level visual features from low-cost frame previews $P(X)$ and token-level text features from the task text q. A lightweight frame-representation module then projects the two streams into a shared space, performs cross-modal fusion and spatiotemporal refinement, and aggregates the refined patch features into one representation per frame:

$$(h_1, \ldots, h_T) = G_\theta(F_{\text{Smol}}(P(X), q)) \,. \tag{3}$$

Here, F_{Smol} and G_θ denote the frozen SmolVLM feature extractor and the trainable frame-representation module, respectively. Each $h_t \in \mathbb{R}^d$ represents candidate frame x_t through its visual content, task relevance, and context within the candidate sequence. A lightweight policy head maps each h_t to the parameters of a frame-specific scale distribution, as described in the next subsection. We keep F_{Smol} frozen and train G_θ and the policy head with RDPO, using retrieval rankings and positive-negative similarity gaps rather than ResAdapt's generated-response feedback.

4.2.2 SCALE POLICY PARAMETERIZATION

For each candidate frame, the policy head maps h_t to the positive shape parameters of a Beta distribution:

$$(\alpha_t, \beta_t) = \phi(H_\theta(h_t)) \,, \quad \alpha_t > 0,\ \beta_t > 0. \tag{4}$$

Here, H_θ is a lightweight residual MLP, and ϕ applies Softplus (Dugas et al., 2000) element-wise to ensure that both parameters are positive. A normalized action a_t is drawn from the predicted Beta distribution and linearly mapped to the allowed scale range:

$$a_t \sim \text{Beta}(\alpha_t, \beta_t), \quad s_t = s_{\min} + a_t(s_{\max} - s_{\min}). \tag{5}$$

Since the Beta distribution is supported on $[0, 1]$, the mapped scale satisfies $s_t \in [s_{\min}, s_{\max}]$. At inference, we retain the training-time scale range by default. The policy assigns $s_t < 1$ to most frames, enabling the backbone to process more frames within the same visual budget. Additionally, adjusting $s_{\min}$ and $s_{\max}$ can provide further control over the visual budget without retraining.

4.2.3 SCALE-AWARE PREPROCESSING

For a candidate frame x_t with spatial size $H_t \times W_t$, the predicted scale s_t gives:

$$H'_t \approx s_t H_t, \quad W'_t \approx s_t W_t, \quad \widetilde{x}_t = R(x_t, s_t). \tag{6}$$

The preprocessing operator R resizes each frame and aligns its dimensions with the embedding model's patch-size requirements. Applying R frame-wise yields the mixed-resolution sequence $\widetilde{X} = (\widetilde{x}_1, \ldots, \widetilde{x}_T)$. Frames assigned larger scales preserve more spatial detail, whereas frames assigned smaller scales consume fewer visual resources.

4.2.4 EMBEDDING BACKBONE INTEGRATION

The frozen embedding backbone E encodes the mixed-resolution sequence and task text as $z = E(\widetilde{X}, q)$. Since allocation is completed before this stage, AllocEmbed leaves the backbone architecture, output dimensionality, and downstream similarity computation unchanged.

4.3 RETRIEVAL-DRIVEN POLICY OPTIMIZATION

Existing datasets do not annotate the optimal input scale for each frame, and the utility of an allocation becomes observable only after the resulting frames have passed through preprocessing, embedding, and retrieval. We therefore introduce Retrieval-Driven Policy Optimization (RDPO), which learns the allocator directly from retrieval outcomes without frame-level scale supervision. As illustrated in Figure 3(b), RDPO builds on group-relative policy optimization (GRPO) (Shao et al., 2024) and compares multiple allocations for the same input using an augmented retrieval candidate set, a rank-validated similarity gap, and an efficiency incentive. RDPO therefore trains the

allocator using retrieval outcomes from the actual, potentially non-differentiable preprocessing and visual tokenization pipeline, without requiring a differentiable surrogate.

4.3.1 ALLOCATION SAMPLING

For each training input (X, q), RDPO samples a group of K frame-wise allocations from the allocator:

$$\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(K)} \sim \pi_\theta(\cdot \mid X, q). \tag{7}$$

Each sampled allocation $\mathbf{s}^{(k)}$ is applied to X through scale-aware preprocessing, and the frozen embedding backbone encodes the resulting mixed-resolution sequence together with q to produce z_k. Within the group, the video, task text, positive target, and retrieval candidates remain fixed, while only the sampled allocation varies.

4.3.2 GLOBAL NEGATIVE SAMPLE AUGMENTATION

Because in-batch negatives alone may be insufficiently challenging, we supplement them with text negatives sampled from the training corpus to form the candidate set

$$\mathcal{U} = \{u^+\} \cup \mathcal{N}^{\text{batch}} \cup \mathcal{N}^{\text{global}}, \tag{8}$$

where u^+ is the positive target embedding, and $\mathcal{N}^{\text{batch}}$ and $\mathcal{N}^{\text{global}}$ are the in-batch and global negative embedding sets, respectively. Since the added negatives are text-only, their encoding cost is small relative to video encoding.

4.3.3 RANK-VALIDATED SIMILARITY GAP

For the k-th allocation, RDPO evaluates retrieval success and the positive target's separation from the negatives. The retrieval success indicator is

$$a_k = \mathbf{1}[\text{rank}_k(u^+) = 1]. \tag{9}$$

For the negative set $\mathcal{N} = \mathcal{U} \setminus \{u^+\}$, the mean negative similarity is

$$\mu_k^- = \frac{1}{|\mathcal{N}|} \sum_{u^- \in \mathcal{N}} \text{sim}(z_k, u^-), \tag{10}$$

and the positive-negative similarity gap and its rank-validated form are

$$g_k = \text{sim}(z_k, u^+) - \mu_k^-, \qquad r_k = a_k g_k. \tag{11}$$

This assigns a score of zero to unsuccessful retrievals and grades successful ones by their separation from the negatives.

We obtain the group-relative retrieval advantage by normalizing r_k across the K allocations of the same input:

$$A_{\text{ret},k} = \frac{r_k - \bar{r}}{\text{std}(r_1, \ldots, r_K) + \epsilon}, \quad \bar{r} = \frac{1}{K} \sum_{j=1}^{K} r_j. \tag{12}$$

This normalization compares allocation quality within each input, reducing sensitivity to differences in example difficulty.

4.3.4 CONFIDENCE-GUIDED EFFICIENCY INCENTIVE

RDPO additionally encourages efficient allocations, but only when they already retrieve the correct target with sufficient confidence. For the k-th allocation, define the mean scale and its normalized cost proxy as

$$\bar{s}_k = \frac{1}{T} \sum_{t=1}^{T} s_t^{(k)}, \quad c_k = \text{clip}\left(\frac{\bar{s}_k - s_{\min}}{s_{\max} - s_{\min}}, 0, 1 \right). \tag{13}$$

We derive a soft confidence score from the similarity gap:

$$h_k = \sigma\left(\frac{g_k - \gamma}{\tau} \right), \tag{14}$$

Method	V-RET					V-QA					Avg.	Cost
	VTT	MSVD	DiDeMo	YC2	VATEX	V-MME	NExT	EgoSch.	MVB	ANetQA		
VLM2Vec-V2-2B												
Base	28.30	48.06	30.38	10.63	26.46	30.70	20.92	34.00	33.70	52.30	31.55	1.000×
Full Res.†	31.90	50.15	33.07	11.20	30.77	31.07	21.28	36.80	34.15	49.80	33.02 (+1.47)	2.952×
Uniform Alloc.	25.80	**49.10**	31.87	11.61	28.20	32.26	21.54	34.80	34.25	51.90	32.13 (+0.58)	1.025×
CA-Alloc.	26.40	45.67	31.87	11.48	27.78	32.52	21.19	35.80	33.50	**53.70**	31.99 (+0.44)	1.003×
CA-Select.	26.80	45.22	30.68	9.81	25.10	31.07	20.95	32.20	33.62	51.20	30.67 (−0.88)	1.000×
AllocEmbed	**29.00**	48.66	**32.77**	**12.05**	**28.94**	**32.56**	**21.82**	**38.60**	**34.48**	53.20	**33.21** (**+1.66**)	0.943×
Qwen3-VL-Embedding-2B												
Base	47.00	68.21	43.33	23.28	40.31	50.48	64.85	57.80	57.63	73.60	52.65	1.000×
Full Res.†	51.10	71.49	48.21	28.22	45.40	53.56	67.90	62.40	59.17	75.40	56.28 (+3.63)	2.923×
Uniform Alloc.	47.20	67.91	46.91	25.86	41.27	53.93	**65.30**	57.60	**57.95**	73.70	53.76 (+1.11)	1.002×
CA-Alloc.	46.00	68.66	45.92	24.91	41.02	52.93	65.26	54.80	56.60	**74.00**	53.01 (+0.36)	1.038×
CA-Select.	46.30	65.67	45.82	22.84	39.01	50.93	65.14	**58.40**	56.95	72.40	52.35 (−0.30)	1.000×
AllocEmbed	**50.20**	**69.70**	**47.61**	**27.27**	**43.52**	**54.85**	63.70	55.80	56.97	73.70	**54.33** (**+1.68**)	0.997×

Table 1: MMEB-V2 Hit@1 (%) with two 2B backbones. Avg. is the ten-task mean, with changes from Base in parentheses; Cost is normalized to Base. Bold marks the best budget-matched result per column. † Full Resolution uses 24 frames at Base resolution and is not budget matched. CA-Alloc./CA-Select. denote Content-Aware Allocation/Selection. Task columns: MSR-VTT, MSVD, DiDeMo, YouCook2, VATEX, Video-MME, NExTQA, EgoSchema, MVBench, ActivityNetQA.

where γ is a confidence threshold and τ controls the softness of the transition. The efficiency incentive is

$$I_{\text{eff},k} = \lambda_{\text{eff}} a_k h_k (1 - c_k). \tag{15}$$

This auxiliary incentive rewards scale reduction only after successful retrieval, with weak incentives for small-gap allocations and stronger incentives for confident ones. The final advantage is

$$A_k = A_{\text{ret},k} + I_{\text{eff},k}. \tag{16}$$

4.3.5 Policy Optimization

For the k-th allocation, RDPO assigns A_k to every valid frame action. Let $\pi_{\theta_{\text{old}}}$ denote the allocator policy used to sample the group of allocations. The likelihood ratio for the normalized Beta action $a_t^{(k)}$ is

$$\rho_{k,t} = \frac{\pi_\theta(a_t^{(k)} \mid X, q)}{\pi_{\theta_{\text{old}}}(a_t^{(k)} \mid X, q)}. \tag{17}$$

RDPO updates the allocator by maximizing the following clipped PPO objective (Schulman et al., 2017):

$$\mathcal{J}(\theta) = \mathbb{E}_{k,t}[\min(\rho_{k,t} A_k, \text{clip}(\rho_{k,t}, 1 - \epsilon_l, 1 + \epsilon_h) A_k)]\,. \tag{18}$$

The expectation is over the sampled allocations and their valid frame actions. The embedding backbone remains frozen throughout optimization.

5 Experiments

5.1 Experimental Setup

Implementation Details. We train AllocEmbed on the video training set used by VLM2Vec-V2 (Meng et al., 2025) while keeping both the SmolVLM2-256M feature extractor and the VLM2Vec-V2-2B embedding backbone frozen. Only the frame-wise allocation module is optimized for 125 steps on eight NVIDIA H100 80GB GPUs with a learning rate of 2×10^{-5}. We use a batch size of 1,024 video–text examples, sampling $K = 16$ allocations per example with continuous scales in $[0.2, 1.8]$. We set $(\lambda_{\text{eff}}, \gamma, \tau) = (0.4, 0.45, 0.10)$ and add 3,072 global text negatives. Further training and RDPO details are provided in Appendix A.1.

Benchmarks and Baselines. We evaluate AllocEmbed on ten MMEB-V2 video tasks (Meng et al., 2025) and LongRet, a newly constructed benchmark for long-video retrieval. We report budget-matched Hit@1 unless noted otherwise. Base encodes eight uniformly sampled frames at their original resolution, while the other methods begin with the same 24 uniformly sampled frames. Uniform Allocation, Content-Aware Allocation, and AllocEmbed retain all 24 but allocate resolution differently, whereas Content-Aware Selection embeds eight selected frames. All methods match the Base budget except Full Resolution, which processes all 24 at the Base resolution as a high-cost reference. Appendix A.2 provides benchmark and baseline details.

Cost and Budget. In our experiments, Cost denotes the average number of tokens fed to the embedding backbone. We report either the absolute average token count or the cost normalized to the corresponding Base, whose reference value is $1.000\times$. Unless otherwise specified, the reference budget is the input-token cost of the same backbone's 8-frame Base.

5.2 Main Results

Table 1 reports task-level and average MMEB-V2 results together with realized costs. AllocEmbed achieves the best budget-matched average with both 2B models while matching or reducing the Base cost. It thus better balances temporal coverage and spatial fidelity. This confirms the value of learned frame-wise resolution allocation.

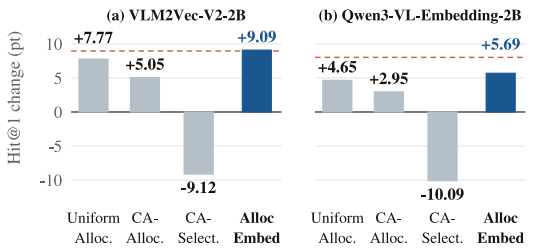


Figure 4: Mean LongRet Hit@1 gains over the 8-frame Base. Bars compare methods at matched cost; dashed lines show Full Resolution at roughly $3\times$ Base cost. CA-Alloc. and CA-Select. denote Content-Aware Allocation and Selection.

Figure 4 summarizes performance on the four LongRet subsets, with complete results provided in Appendix B.3. AllocEmbed's gains over Base are more pronounced on LongRet, ranging from 5.69 to 9.09 Hit@1 points, as sparse frame sampling is more likely to miss retrieval-relevant evidence in long videos. It is the strongest budget-matched method on all four subsets and recovers most of Full Resolution's improvement at roughly one-third of the cost. This highlights the importance of preserving broad temporal coverage while varying spatial fidelity in long-video retrieval.

5.3 Ablation Studies

We ablate RDPO across the ten MMEB-V2 video tasks using VLM2Vec-V2-2B, retraining each variant on the same training data. During evaluation, all variants use 24 input frames, with the remaining allocation settings unchanged from training. Table 2 reports average Hit@1 and realized visual cost.

Variant	Hit@1	Δ (pt)	Cost	Δ (%)
Full RDPO	33.21	—	2,008.44	—
Retrieval Success Only	32.05	−1.16	1,576.40	−21.51
Similarity Gap Only	32.92	−0.29	2,138.27	+6.46
w/o Global Negatives	31.91	−1.30	1,242.87	−38.12
w/o Confidence Guidance	32.90	−0.31	2,155.19	+7.31
w/o Efficiency Incentive	32.85	−0.36	2,180.98	+8.59
w/o Success Gating	33.05	−0.16	2,452.24	+22.10

Table 2: RDPO ablation on ten MMEB-V2 tasks with VLM2Vec-V2-2B. Hit@1 is a percentage; both Δ columns compare with Full RDPO.

Full RDPO achieves the best quality–cost balance. Retrieval success alone provides too sparse a learning signal, while the ungated similarity gap preserves most of the accuracy at greater visual cost. Combining retrieval success with the similarity gap provides both a reliable success criterion and graded ranking feedback. Removing global negatives causes the largest accuracy loss of 1.30 points, highlighting the importance of challenging negatives beyond in-batch comparisons. Confidence guidance, success gating, and the efficiency incentive primarily regulate cost: removing any of them offers little benefit while increasing resource use.

5.4 Analysis and Discussion

5.4.1 Generalization to Diverse Embedding Models

To assess cross-model transfer, we evaluate AllocEmbed with the four embedding models in Table 3, spanning 2B to 8B parameters. Trained only with VLM2Vec-V2-2B, the allocator transfers directly to the other three models without further training. Table 3 reports the Hit@1 gain and visual cost relative to each model's own Base. Across all four models, AllocEmbed improves both averages while matching or reducing the visual cost of the corresponding Base. The gains are consistent on MMEB-V2 and substantially larger on LongRet, showing that the learned allocation policy transfers across embedding architectures and parameter scales.

Model	V-RET & V-QA		LongRet	
	Gain	Cost	Gain	Cost
VLM2Vec-V2-2B	+1.66	0.943×	+9.09	0.979×
VLM2Vec-Qwen2VL-7B	+1.34	0.943×	+7.07	0.979×
Qwen3-VL-Emb.-2B	+1.68	0.997×	+5.69	0.977×
Qwen3-VL-Emb.-8B	+1.69	0.997×	+7.63	0.977×

Table 3: Generalization across embedding models. Gain is the Hit@1 change from the 8-frame Base, and Cost is visual cost normalized to Base.

5.4.2 Performance across Visual Input Budgets

To assess robustness across visual input budgets, we evaluate AllocEmbed with VLM2Vec-V2-2B at three budget levels. Base processes 8, 16, or 20 native-resolution frames, while AllocEmbed uses candidate pools of 24, 48, or 60 frames under the corresponding Base budget. Figure 5 shows that AllocEmbed outperforms Base at every level on both MMEB-V2 and LongRet. Gains are especially pronounced on LongRet, reaching 6.25 Hit@1 points at the largest 20-frame budget. These results show that learned allocation remains effective as the budget scales up. Further analysis of inference efficiency is provided in Appendix B.2.

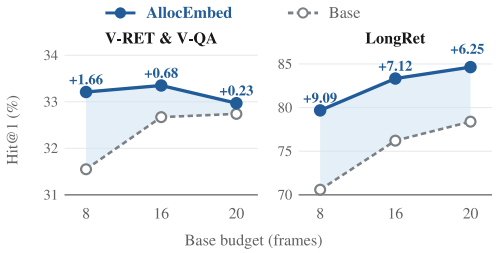


Figure 5: Mean Hit@1 on MMEB-V2 (left) and LongRet (right). Base uses 8, 16, or 20 frames; AllocEmbed is evaluated at matched cost. Labels show gains over Base in percentage points.

5.4.3 Case Study

To inspect AllocEmbed's allocations, we analyze two examples with complete 24-frame visualizations in Appendix B.1. For the Oscars location question, clear frames with the red carpet, statuettes, and location-revealing text receive larger scales, while a black transition receives the minimum. For character counting, sharp game footage and text-bearing frames receive larger scales than blank or blurred transitions. These examples show that AllocEmbed prioritizes informative, high-quality frames.

6 Conclusion

We introduced AllocEmbed, an allocate-then-embed framework that balances temporal coverage and spatial fidelity by distributing a constrained visual-input budget across a denser frame sequence. AllocEmbed predicts input resolutions from video content and task text before embedding, allowing existing VLM-based embedding models to process mixed-resolution videos without architectural changes. RDPO learns this allocation directly from retrieval outcomes through a rank-validated similarity gap and a confidence-guided efficiency incentive. Experiments on MMEB-V2 and our newly constructed LongRet benchmark show improved aggregate retrieval performance under matched visual-input budgets, together with transfer across backbones from 2B to 8B parameters without retraining. These results highlight input-side visual allocation as a practical approach to improving video retrieval with existing embedding backbones.

AI USE STATEMENT

During manuscript preparation, generative AI tools were only used for language polishing and presentation formatting. The authors take full responsibility for the final content, including all scientific claims, results, and AI-assisted edits.

ETHICS STATEMENT

This work studies efficient video retrieval using existing datasets. LongRet is constructed from LoVR and Vript, as described in Appendix A.2. Video retrieval systems can inherit biases from source datasets and pretrained models, and can be misused for privacy-invasive search. These considerations should inform downstream deployment. The planned release of LongRet will respect the access and redistribution conditions of its source datasets.

REPRODUCIBILITY STATEMENT

We provide the implementation code in the supplementary material. Section 5 presents the experimental protocol and results. Appendix A.1 provides the training configuration and hyperparameters, while Appendix A.2 details benchmark construction and evaluation protocols. We will publicly release the LongRet evaluation benchmark to support independent evaluation and future comparisons.

REFERENCES

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *Proceedings of the IEEE international conference on computer vision*, pp. 5803–5812, 2017.

Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1708–1718. IEEE, 2021.

Shyamal Buch, Arsha Nagrani, Anurag Arnab, and Cordelia Schmid. Flexible frame selection for efficient video reasoning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29071–29082. IEEE, 2025.

David Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pp. 190–200, 2011.

Charles Dugas, Yoshua Bengio, François Bélisle, Claude Nadeau, and René Garcia. Incorporating second-order functional knowledge for better option pricing. *Advances in neural information processing systems*, 13, 2000.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24108–24118. IEEE, 2025.

Zhiming Hu, Angela Ning Ye, Salar Hosseini Khorasgani, and Iqbal Mohomed. Adaclip: Towards pragmatic multimodal video retrieval. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 5623–5633, 2023.

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.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22195–22206, 2024.

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.

Hao Liang, Qifeng Cai, Zhaoyang Han, Hejun Dong, Meiyi Qiang, Ruichuan An, Quanqing Xu, Bin Cui, and Wentao Zhang. Lovr: A benchmark for long video retrieval in multimodal contexts. In *Proceedings of the ACM Web Conference 2026*, pp. 1911–1922, 2026.

Huanxuan Liao, Zhongtao Jiang, Yupu Hao, Yuqiao Tan, Shizhu He, Ben Wang, Jun Zhao, Kun Xu, and Kang Liu. Resadapt: Adaptive resolution for efficient multimodal reasoning. *arXiv preprint arXiv:2603.28610*, 2026.

Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing*, 508: 293–304, 2022.

Yiwei Ma, Guohai Xu, Xiaoshuai Sun, Ming Yan, Ji Zhang, and Rongrong Ji. X-clip: End-to-end multi-grained contrastive learning for video-text retrieval. In *Proceedings of the 30th ACM international conference on multimedia*, pp. 638–647, 2022.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems*, 36:46212–46244, 2023.

Andrés Marafioti, Orr Zohar, Miquel Farré, Merve Noyan, Elie Bakouch, Pedro Cuenca, Cyril Zakka, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, et al. Smolvlm: Redefining small and efficient multimodal models. *arXiv preprint arXiv:2504.05299*, 2025.

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.

Yue Meng, Chung-Ching Lin, Rameswar Panda, Prasanna Sattigeri, Leonid Karlinsky, Aude Oliva, Kate Saenko, and Rogerio Feris. Ar-net: Adaptive frame resolution for efficient action recognition. In *European conference on computer vision*, pp. 86–104. Springer, 2020.

Michael S Ryoo, Honglu Zhou, Shrikant Kendre, Can Qin, Le Xue, Manli Shu, Jongwoo Park, Kanchana Ranasinghe, Silvio Savarese, Ran Xu, et al. xgen-mm-vid (blip-3-video): You only need 32 tokens to represent a video even in vlms. *arXiv preprint arXiv:2410.16267*, 2024.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Leqi Shen, Tianxiang Hao, Tao He, Sicheng Zhao, Yifeng Zhang, Yongjun Bao, Guiguang Ding, et al. Tempme: Video temporal token merging for efficient text-video retrieval. In *International Conference on Learning Representations*, volume 2025, pp. 60839–60860, 2025.

Leqi Shen, Guoqiang Gong, Tao He, Yifeng Zhang, Pengzhang Liu, Sicheng Zhao, et al. Fastvid: Dynamic density pruning for fast video large language models. *Advances in Neural Information Processing Systems*, 38:123553–123581, 2026.

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, et al. Longvu: Spatiotemporal adaptive compression for long video-language understanding. *arXiv preprint arXiv:2410.17434*, 2024.

Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4580–4590. IEEE, 2019.

Mengxia Wu, Min Cao, Yang Bai, Ziyin Zeng, Chen Chen, Liqiang Nie, and Min Zhang. An empirical study of frame selection for text-to-video retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 6821–6832, 2023.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9772–9781. IEEE, 2021.

Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 6787–6800, 2021.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5288–5296, 2016.

Dongjie Yang, Suyuan Huang, Chengqiang Lu, Xiaodong Han, Haoxin Zhang, Yan Gao, Yao Hu, and Hai Zhao. Vript: A video is worth thousands of words. *Advances in Neural Information Processing Systems*, 37:57240–57261, 2024.

Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 9127–9134, 2019.

Shaojie Zhang, Jiahui Yang, Jianqin Yin, Zhenbo Luo, and Jian Luan. Q-frame: Query-aware frame selection and multi-resolution adaptation for video-llms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22056–22065, 2025.

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.

Luowei Zhou, Chenliang Xu, and Jason Corso. Towards automatic learning of procedures from web instructional videos. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Category	Setting	Value	Notes
Infrastructure	Hardware	8 NVIDIA H100 80GB GPUs	Distributed training on a single node.
Infrastructure	Parallelism and precision	FSDP2, BF16	FSDP2 parameter sharding with BF16 computation.
Model	Embedding backbone	VLM2Vec-V2-2B ($\sim$2.21B)	Frozen model used to compute embeddings and retrieval rewards.
Model	Allocator implementation	SmolVLM2-256M-based allocator	Frozen SmolVLM2-256M feature extractor followed by a trainable frame-wise scale predictor.
Optimization	Allocator learning rate	2×10^{-5}	Applied only to the trainable scale predictor.
Optimization	Training duration	125 steps	Total number of training steps.
Data	Video splits	Complete training split; complete validation split	Train on the training split; use the validation split for checkpoint selection.
Batching	Examples per step	1,024	Video–text examples sampled before allocation expansion.
Batching	Allocations per example	$K = 16$	Scale allocations sampled for each example.
Allocation	Scale range	$[0.2, 1.8]$	Per-frame scale applied to both height and width.
RDPO	Additional global negatives	3,072	Text targets sampled beyond the current batch.
RDPO	Efficiency weight	$\lambda_{\text{eff}} = 0.4$	Strength of the success-gated efficiency incentive.
RDPO	Confidence gate	$\gamma = 0.45,\ \tau = 0.10$	Center and temperature for similarity-gap confidence.
RDPO	Clipping widths	$\epsilon_l = 0.20,\ \epsilon_h = 0.28$	Asymmetric policy-ratio clipping widths.

Table 4: Training configuration used for AllocEmbed. Only the frame-wise scale predictor is trained, while the SmolVLM2 feature extractor and embedding backbone remain frozen.

A Additional Experimental Setup

A.1 Implementation Details

Table 4 summarizes the training configuration. We train on the complete VLM2Vec-V2 video training split and use the complete validation split for checkpoint selection; neither split is manually truncated. The VLM2Vec-V2-2B embedding backbone contains approximately 2.21B parameters and remains frozen, providing the embeddings used to compute retrieval rewards. The allocator encodes candidate-frame previews and task text with a frozen SmolVLM2-256M feature extractor. Only the attached frame-wise scale predictor is optimized.

Training runs on a single node with eight NVIDIA H100 80GB GPUs using FSDP2 and BF16 computation. We optimize the scale predictor for 125 steps with a learning rate of 2×10^{-5}. Each step samples 1,024 video–text examples and draws $K = 16$ allocations per example, yielding 16,384 allocation-conditioned queries. The prompt mini-batch size is 256. After allocation expansion and distributed sharding, the allocator mini-batch contains 512 samples per GPU. Scale prediction and RDPO backpropagation use a micro-batch size of 64 per GPU, while the frozen embedding backbone encodes queries and targets in batches of 128 per GPU.

For each candidate frame, the predictor outputs a continuous scale in $[0.2, 1.8]$. Because the scale is applied to both spatial dimensions, visual input cost grows approximately quadratically with scale.

Following the RDPO formulation in the main text, we set $\lambda_{\text{eff}} = 0.4$, $\gamma = 0.45$, and $\tau = 0.10$. The success gate prevents an incorrect allocation from receiving an efficiency incentive solely by

(a) Evaluation composition

Subset	Queries	Unique full-video targets	Source videos	Query–target mapping
LoVR-Clip	1,401	467	467	Local description → complete video
LoVR-Video	467	467	467	Complete description → complete video
Vript-Clip	7,112	2,375	2,375	Local description → complete video
Vript-Video	2,375	2,375	2,375	Complete description → complete video

(b) Unique complete-video duration statistics

Corpus	Videos	Total (h)	Duration (min)						
			Mean	Min	P25	Median	P75	P95	Max
LoVR	467	202.41	26.01	16.64	18.49	21.56	29.00	50.59	59.57
Vript	2,375	563.53	14.24	10.00	11.28	13.39	16.36	20.93	29.82

(c) Unique complete-video duration distribution

Corpus	$<$ 10 min	10–15 min	15–20 min	20–30 min	$\geq$ 30 min
LoVR	0	0	189 (40.5%)	170 (36.4%)	108 (23.1%)
Vript	0	1,560 (65.7%)	661 (27.8%)	154 (6.5%)	0

Table 5: LongRet composition and complete-video duration statistics. Clip and Video share targets and duration distributions within each corpus but differ in query granularity and count. Bins are left-closed and right-open, except the final bin.

reducing its scale. We augment the 1,024 in-batch targets with 3,072 text targets sampled from the VLM2Vec-V2 training corpus. The resulting 4,096-target pool is shared across all 16,384 allocation-conditioned queries. Additional negatives exclude the current positive, targets with the same sample identity, and exact text duplicates.

A.2 Benchmark and Baseline Details

Our MMEB-V2 evaluation comprises five video-retrieval (V-RET) tasks and five video-question-answering (V-QA) tasks. The V-RET tasks are MSR-VTT (Xu et al., 2016), MSVD (Chen & Dolan, 2011), DiDeMo (Anne Hendricks et al., 2017), YouCook2 (Zhou et al., 2018), and VATEX (Wang et al., 2019). The V-QA tasks are Video-MME (Fu et al., 2025), NExTQA (Xiao et al., 2021), EgoSchema (Mangalam et al., 2023), MVBench (Li et al., 2024), and ActivityNet-QA (Yu et al., 2019). We evaluate four embedding models: VLM2Vec-V2-2B, Qwen3-VL-Embedding-2B, VLM2Vec-Qwen2VL-7B, and Qwen3-VL-Embedding-8B.

LongRet benchmark construction. LongRet evaluates text-to-video retrieval at two query granularities while using complete source videos as retrieval targets. Its candidate corpora contain 467 LoVR videos (Liang et al., 2026) and 2,375 Vript videos (Yang et al., 2024). Localized scene annotations contribute query text only: no scene clip is treated as a retrieval target, and every query must retrieve the complete source video from which its description originates. We therefore compute all duration statistics over unique complete-video targets rather than over localized scene annotations.

The Clip subsets use localized descriptions. We randomly sample up to three eligible segment captions per video and treat each caption as a separate query, producing 1,401 LoVR-Clip queries and 7,112 Vript-Clip queries. The Video subsets use complete-video descriptions. LoVR provides one video-level caption per source video, yielding 467 LoVR-Video queries. For Vript, we concatenate the clip captions associated with each source video into one complete-video query, yielding 2,375 Vript-Video queries. Overall, LongRet contains 11,355 queries over 2,842 source videos. Table 5 reports the evaluation composition, full-video duration statistics, and duration-bin counts.

Baseline and evaluation protocols. Under the default protocol, Base embeds 8 uniformly sampled frames at the original resolution, while the other methods begin with 24 candidate frames. Full

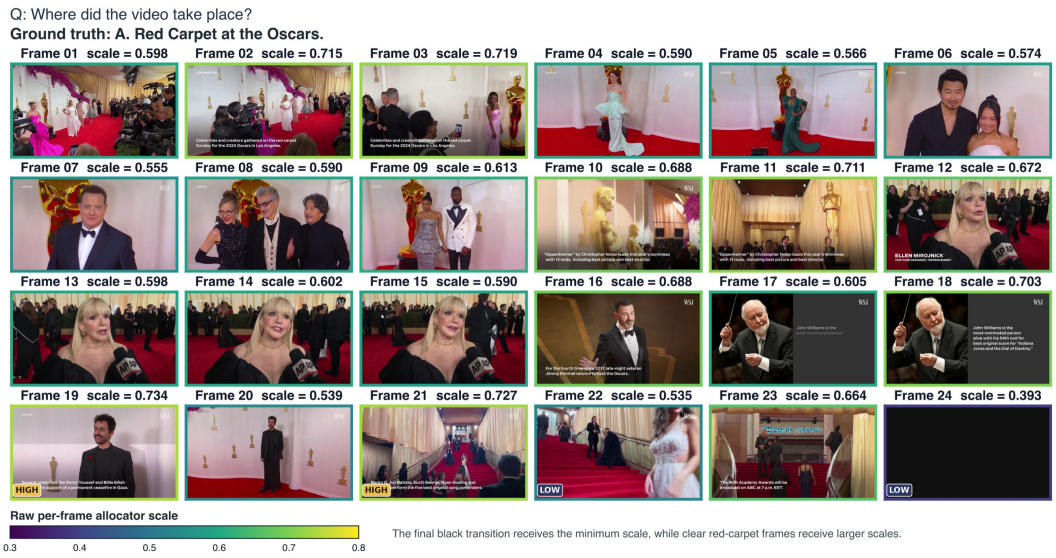


Figure 6: Frame-level scales for a location question. AllocEmbed assigns the minimum scale to the final black transition and larger scales to clear views of the red carpet and Oscar iconography.

Resolution processes all 24 frames at the Base resolution and incurs a higher visual cost. Uniform Allocation lowers their resolution uniformly to match the Base budget. Content-Aware Allocation assigns resolutions using a frame-importance heuristic, while Content-Aware Selection embeds the 8 highest-scoring frames. AllocEmbed predicts a continuous, task-conditioned scale for each candidate frame.

Across methods, we hold the input video, task text, retrieval targets, and metric fixed, and report Hit@1 with realized visual input cost. All methods except Full Resolution are evaluated under matched budgets. At inference, AllocEmbed predicts one allocation and invokes the frozen backbone once after scale-aware preprocessing. Group sampling with $K = 16$ is used only for training.

B Additional Analysis and Discussion

B.1 Case Study Visualizations

Figures 6 and 7 present the complete 24-frame visualizations for the examples discussed in the Case Study subsection. Each frame is labeled with its raw predicted scale, and the border color encodes the same value.

B.2 Inference Overhead

We measure inference overhead on the LoVR-Video candidate videos using VLM2Vec-V2.0 (Qwen2-VL). All experiments run in BF16 with batch size one on a single NVIDIA H100 80GB GPU. In paired runs, AllocEmbed and the full-input baseline process the same videos, text, and 24 sampled frames in the same order under a 64K video-token cap. AllocEmbed predicts scales in two-frame chunks within $[0.2, 1.32]$. The full-input baseline bypasses allocation and sends all 24 frames directly to the backbone.

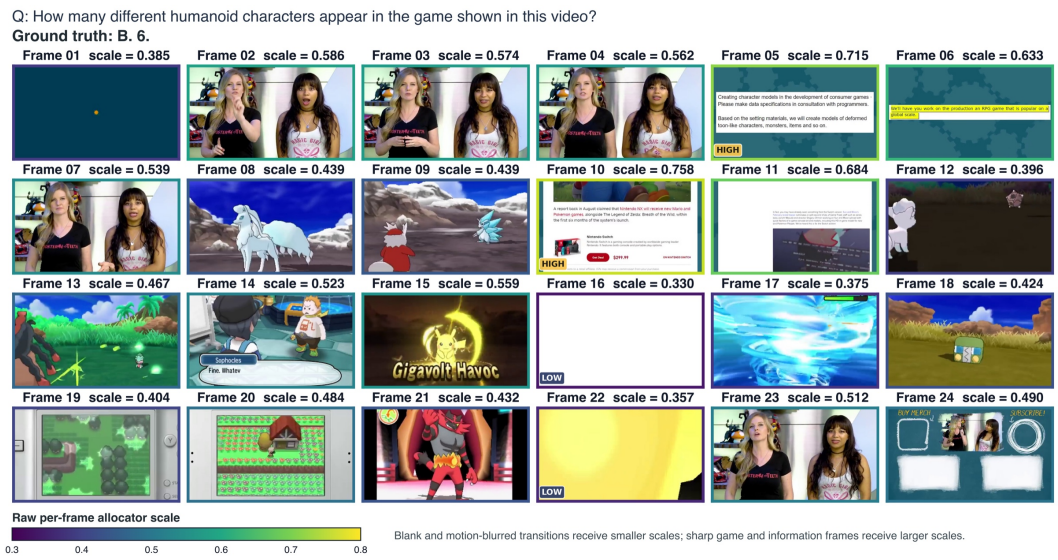


Figure 7: Frame-level scales for a character-counting question. AllocEmbed assigns smaller scales to blank and motion-blurred transitions and larger scales to sharp game footage and text-rich frames.

We discard five warm-up samples and average latency over the remaining 462 videos, synchronizing CUDA around each timed region. Allocator latency includes text preparation, host-to-device transfer, the allocator encoders and prediction head, and scale generation. Backbone latency includes host-to-device transfer and one VLM2Vec forward pass. Gradient computation and parameter updates are disabled for all runs.

		Latency (ms/sample)		
Method	Params	Allocator	Backbone	Total
Full input	2.218B	–	754.77	754.77
AllocEmbed	2.523B	149.59	239.22	**388.81**

Table 6: Model-path latency on LoVR-Video. Results are averaged over 462 samples after five warm-up samples. Params denotes the total number of loaded parameters.

As shown in Table 6, the allocator adds 149.59 ms per sample but reduces backbone latency from 754.77 to 239.22 ms. Total model-path latency decreases by 48.49%, corresponding to a $1.941\times$ speedup over full-input inference. These measurements exclude allocator preprocessing, scale-dependent resizing, video transforms, and backbone preprocessing. The excluded stages can run on the CPU and are amenable to further systems optimization through parallelism and pipelining. A full optimization of these stages is outside the scope of this work.

B.3 Complete LongRet Results

Table 7 reports the complete subset-level LongRet results summarized in Figure 4.

Method	LoVR		Vript		Avg.	Cost
	Clip	Video	Clip	Video		
VLM2Vec-V2-2B						
Base	61.10	74.09	69.92	77.26	70.59	1.000×
Full Resolution†	71.23	83.73	78.63	84.59	79.54 (+8.95)	2.997×
Uniform Allocation	70.45	82.66	75.58	84.76	78.36 (+7.77)	1.016×
Content-Aware Allocation	65.67	80.09	72.60	84.21	75.64 (+5.05)	1.001×
Content-Aware Selection	48.39	65.10	58.96	73.43	61.47 (−9.12)	1.000×
AllocEmbed	**71.45**	**83.73**	**77.24**	**86.32**	**79.68** (+**9.09**)	0.979×
Qwen3-VL-Embedding-2B						
Base	80.94	83.51	81.83	88.51	83.70	1.000×
Full Resolution†	91.01	90.58	91.84	93.43	91.72 (+8.02)	2.984×
Uniform Allocation	82.87	89.51	88.05	92.97	88.35 (+4.65)	1.036×
Content-Aware Allocation	78.52	88.87	85.24	93.98	86.65 (+2.95)	1.018×
Content-Aware Selection	59.96	78.59	67.74	88.17	73.61 (−10.09)	1.000×
AllocEmbed	**83.87**	**90.15**	**89.20**	**94.36**	**89.39** (+**5.69**)	0.977×

Table 7: Complete LongRet results corresponding to Figure 4. Base embeds 8 uniformly sampled frames. The remaining methods start from 24 candidate frames, with Content-Aware Selection embedding 8. Values are Hit@1 (%), and Avg. is the mean over four subsets. Parentheses show absolute changes from Base. Cost is normalized to Base (1.000×). Bold marks the best budget-matched result in each column. † Full Resolution processes all 24 frames at the Base per-frame resolution and is not budget matched.