Training-free Focus-Ambient Retention for Memory-Efficient Video Large Language Models

Haozheng Zeng[1,2], **Qi Bi**[2,3✉*], **Gui-Song Xia**[1,2,4]
[1] School of Computer Science, Wuhan University, China
[2] School of Artificial Intelligence, Wuhan University, China
[3] Computer Vision Research Group, University of Amsterdam, the Netherlands
[4] National Engineering Research Center for Multimedia Software, Wuhan University, China
stephenzeng@whu.edu.cn, q.bi@ieee.org, guisong.xia@whu.edu.cn

Abstract

Video large language models are often dominated by visual tokens, which lengthen prefill, enlarge KV-cache residency, and raise peak memory. In spirit of the philosophy *focus more and memorize less*, we introduce Focus–Ambient Retention (FAR), a training-free visual token retention method that routes visual evidence through two complementary streams. The Focus stream preserves task-critical objects, actions, and fine details, while the Ambient stream keeps scene context mainly when it differs from a temporal cache. FAR combines visual attention, local frequency variation, and bounded query relevance to score tokens, and then assembles a fixed-budget context with diversity control before language-model prefill. Across five video understanding benchmarks under a variety of model sizes, FAR reaches accuracy comparable to the state-of-the-art under matched retained-token budgets while reducing peak VRAM by up to 50.6% (from 36.46 to 18.02 GiB against FlashVID on Qwen2.5-VL-7B), improving the accuracy–memory trade-off. Compared with other pre-LLM methods at the same memory level, FAR gives the highest mean score from 5% to 25% retention. Overall, FAR offers a position-aware alternative to frame- or block-level compression for efficient inference. Code is available at `https://gitlab.com/stephenzeng/far`.

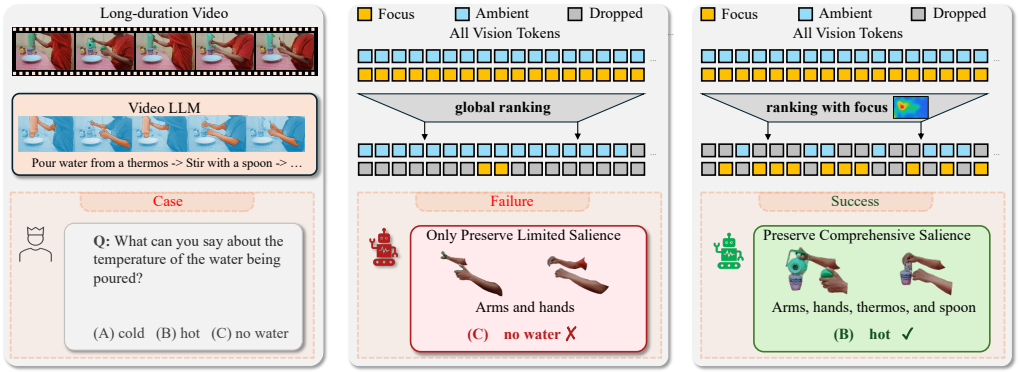
1 Introduction

Video large language models (VLLMs) have improved multimodal understanding of events, actions, text, and long-range temporal context in videos [23, 60, 3, 11]. However, long-duration video inference remains expensive. A single video can contribute thousands of visual tokens to the multimodal prefill sequence, making visual tokens the dominant source of attention cost and KV-cache residency [45]. Training-free visual token compression [39, 4, 7, 52] is therefore a practical route to lower memory use without retraining the backbone or changing the model interface.

Existing visual token compression methods prune or merge tokens using salience, similarity, or density [4, 7, 52, 59]. Video-specific methods further exploit temporal structure, dynamic density, similarity-importance fusion, keyframe priors, or holistic merging [39, 17, 21, 38, 37, 27]. These methods have demonstrated that video tokens are highly redundant, yet many of them still reduce retention to one global ranking over importance. While such a single ranking forces all evidence to compete for the same budget, the task-critical details and slowly changing scene context in long videos can however exhibit in various cases and for different reasons (exemplified in Fig. 1a). This granularity mismatch can repeatedly keep conspicuous foreground evidence while dropping low-

*✉: Correspondence to Qi Bi (q.bi@ieee.org)

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) extensive foreground evidence (b) existing global ranking paradigm (c) the proposed method

Figure 1: (a) A video large language model needs to *focus* comprehensive salience to understand a video. (b) The state-of-the-art training-free efficient video understanding methods usually implement global ranking. Some tokens that deserve to *focus* are discarded after retention, degrading the understanding capability. (c) The proposed method aims to *focus* as much salience in a video as possible even with a modest and even limited budget, allowing efficient and effective video inference.

salience context that anchors the scene, or discard stable-looking background (demonstrated in Fig. 1b).

The central observation of this work is that long video visual evidence serves two complementary properties. *Focus* evidence contains objects, actions, and query-relevant details whose removal can directly change the answer. *Ambient* evidence captures layout, background, camera context, and slowly changing scene state. Such high redundancy from the Ambient evidence should not be transmitted repeatedly, whereas cache-dissimilar context should be promoted when it indicates a new scene state. This asymmetry motivates role-aware evidence routing (demonstrated in Fig. 1c) rather than a single top-k ranking.

We propose Focus–Ambient Retention (FAR), a training-free visual token retention method for memory-efficient long video inference. In spirit of the philosophy *focus more and memorize less*, its high-level idea is to handle the visual evidence from the above two complementary properties in a divide-and-conquer manner. With the aid of the visual salience, FAR scores tokens and routes them into two tracks. The focus tokens are retained as direct task evidence. The ambient tokens are compared with a temporal cache, allowing stable context to be suppressed and changed context to be promoted under the same budget. FAR finally assembles a compact, diversity-aware visual sequence and restores retained tokens to their original order before LLM prefill.

Technically, FAR follows a coupled Scoring-Cache-Assembly pipeline. Scoring computes bounded multimodal importance scores, where query relevance refines visual salience within a semantic trust region. These scores separate the directly retained focus tokens from the ambient candidates, which are then matched against the temporal cache. Stable tokens softly refresh memory, while cache-deviating tokens are promoted as changed context. Assembly combines focus and changed ambient tokens under the target retention ratio, with retention-aware Focus allocation and diversity-aware selection preventing the budget from collapsing into redundant foreground evidence. This flow preserves task-critical cues and evolving scene context while remaining lightweight and training-free.

Large-scale experiments covering broad and long-form reasoning and across multiple retention ratios have been evaluated on five standard video understanding benchmarks, namely, VideoMME [16], EgoSchema [32], LongVideoBench [48], MVBench [25], and MLVU [61]. The experiment results across Qwen2.5-VL-3/7/32B [3] and GLM-4.6V-Flash-10B [19] show that FAR keeps accuracy comparable to the state-of-the-art and consistently lowers memory under matched retained-token budgets, improving the accuracy–memory trade-off.

Our contributions are summarized as follows:

- We formulate long video retention as a Focus–Ambient context construction problem that balances direct preservation of task-critical evidence with incremental maintenance of slowly changing ambient context.

- We introduce Focus–Ambient Retention (FAR), a lightweight training-free framework that couples scoring, temporal cache matching, and budgeted assembly to produce a compact visual context before LLM prefill.
- Extensive matched budget evaluation is conducted on long video understanding benchmarks under a variety of model sizes, showing up to 50.6% peak-VRAM reduction at accuracy comparable to the state-of-the-art.

2 Related Work

MLLMs. Modern multimodal large language models (MLLMs) connect pretrained visual encoders with language models. Earlier CLIP-style vision-language pretraining established the foundation [35, 24, 12, 28, 29], paving the way for recent high-resolution perception, long-context multimodal reasoning, and visual instruction following [30, 23, 46, 3, 8, 62, 47, 44, 19]. However, when encountering long visual sequences, the visual tokens can dominate prefill cost and memory residency.

Efficient MLLMs. One research line focuses on token compression. Typical works include FastV [7], VisionZip [52], SparseVLM [59], PyramidDrop [49], TopV [51], VScan [55], VisPruner [56], LLaVA-PruMerge [36], and DivPrune [2]. Another research line is token merging, which shortens sequences by fusing similar representations [4, 58, 5, 20]. A third research line is relevance scoring, which selects the visual tokens by attention or salience [57, 40, 6]. In contrast, FAR presents a lightweight prefill-time signal inside a track-separated retention pipeline, which first preserves Focus evidence and then decides whether Ambient tokens carry new temporal evidence through cache matching.

Video LLMs & Efficient Video LLMs. Video LLMs extend image-centric MLLMs to temporal inputs. Representative works include Video-ChatGPT [31], LLaVA-Video [60], LLaMA-VID [26], PLLaVA [50], LongVU [39], and LLaVA-OneVision [23]. However, their dense frame sampling still creates large spatiotemporal token sets. Recent efficient video LLM methods use dynamic pruning, density control, similarity-importance fusion, multi-granular spatiotemporal merging, holistic token merging, dynamic token compression, or LLM-guided keyframe priors [21, 17, 38, 22, 37, 43, 27]. The state-of-the-art FlashVID combines attention token selection with tree-based spatiotemporal token merging [15]. These approaches commonly compress the video token budget with a unified policy over visual redundancy or importance. In contrast, the proposed FAR keeps Focus tokens in every frame and treats Ambient tokens as cache-querying evidence, yielding a retained context that mixes direct task evidence with incremental background refresh.

Pre-LLM Visual Cache and Memory Reuse. Pre-LLM visual cache and memory mechanisms make long video understanding more feasible by storing visual history before evidence enters the language decoder. Video object segmentation methods provide precedents for feature-memory matching and bounded long-term visual state reuse [33, 10, 54, 53, 9]. MovieChat [41], MA-LMM [18], VideoStreaming [34], ReWind [13], and VideoAgent [14] maintain long-term, online, instructed, or object-centric memories for video reasoning. In contrast, the proposed FAR is a training-free token retention method before LLM prefill. It treats cache-similar tokens as stable context for suppression or backfill, turning long video compression from a single token ranking into budgeted evidence routing under temporal redundancy.

3 Method

3.1 Problem Setup, Motivation & Framework Overview

Problem Setup. Given a video with F frames, a base video large language model (VLLM) encodes each frame into visual embeddings $X = \{X_t\}_{t=1}^{F}$, where $X_t \in \mathbb{R}^{N \times d}$ contains N visual tokens of dimension d. The visual tokens are concatenated with embedded text tokens and consumed by the language model during prefill. Then, the response tokens are decoded autoregressively. Since the visual length grows with FN, long videos increase prefill attention and feed-forward computation, enlarge KV-cache residency, and raise peak memory. This paper addresses a training-free before-LLM retention problem. Under a target retention ratio R, the retained visual tokens $\hat{X}_t \subseteq X_t$ are selected before downstream LLM layers see the multimodal sequence, and later written back into the multimodal sequence while preserving their original global order and positional indices.

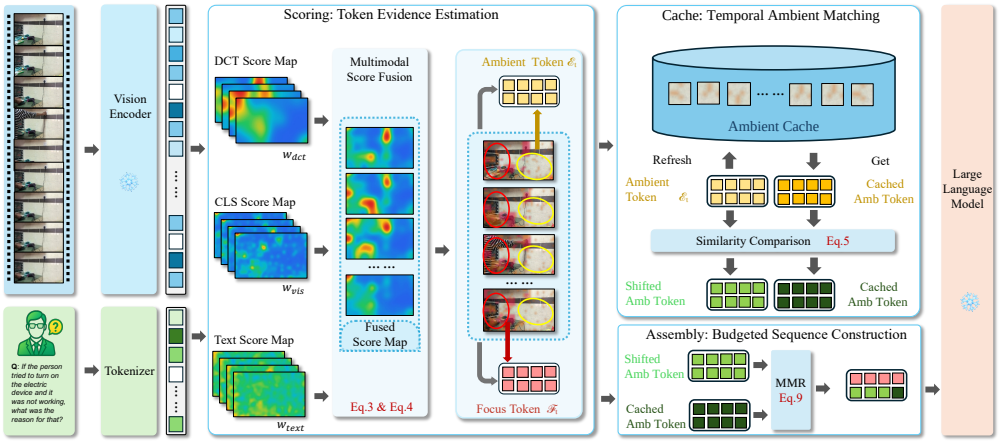


Figure 2: Overview of Focus–Ambient Retention. FAR performs training-free before-LLM visual token retention by separating direct task evidence from incrementally refreshed scene context. Scoring estimates token evidence, Cache suppresses redundant *ambient tokens* while promoting changed context, and Assembly produces a compact position-preserving visual sequence for LLM prefill.

Motivation. The key abstraction is to separate visual evidence into two streams. *Focus* evidence contains task-critical objects, actions, and fine details, and is retained directly. *Ambient* evidence contains scene layout and background context, and is transmitted incrementally, mainly when it differs from cached context. This distinction lets FAR spend its budget on salient or changed evidence rather than repeatedly forwarding stable background.

Framework Overview. Fig. 2 shows the full pipeline of the proposed FAR. FAR consists of three modules, namely, Scoring, Cache, and Assembly. Scoring computes multimodal token scores and partitions each frame into *focus* and *ambient* tracks. Cache compares *ambient tokens* against a temporal memory to identify shifted context and refreshes the memory with motion-aware time decay. Assembly combines *focus tokens*, shifted *ambient tokens*, and optional stable backfill under a fixed budget. Then, it rewrites the visual portion of the input before LLM prefill.

3.2 Scoring: Token Evidence Estimation

An ideal scoring for efficient video LLMs is supposed to identify the task-critical evidence while excluding ambiguous query cues. To address this challenge, the proposed FAR computes bounded multimodal scores that remain visually grounded while incorporating prompt relevance. These scores distinguish the directly retained *focus tokens* from the *ambient* candidates for temporal matching.

Memory-aware [CLS] Attention Encoding. To identify evidence emphasized by the vision encoder, FAR first computes the [CLS] Attention score $A_t \in \mathbb{R}^N$ by averaging the attention weights each token receives within its frame. For encoders without an explicit [CLS] token (e.g., Qwen2.5-VL), A_t is derived from the last-block attention matrix following FlashVID [15]. A naive extraction would materialize an auxiliary tensor of shape (F, H, N, N) for H attention heads, with peak memory $\mathcal{O}(FHN^2)$. To address this issue, FAR instead computes attention frame by frame, which immediately reduces the single-frame attention over heads and query tokens and stores only one vector A_t per frame. The auxiliary peak of this extraction step therefore becomes $\mathcal{O}(HN^2)$ while preserving (F, N) interface for token retention.

Multi-Cue Fusion. Beyond [CLS] Attention $A_{t,i}$, FAR also considers the text relevance, which is quantified by $T_{t,i}$ and injects instruction-conditioned preference. $T_{t,i}$ is the cosine similarity between the visual token and the mean text embedding, following the intuition that instruction-conditioned pruning should preserve prompt-relevant evidence [57]. Both lie in the LLM input embedding space: visual tokens are the outputs of the vision merger, and the text embedding averages the input embeddings of all prompt tokens. However, attention and text cues can over-concentrate on semantically dominant regions and miss answer-critical changes. Therefore, FAR introduces a Discrete Cosine Transform (DCT) [1] based energy score $D_{t,i}$, which highlights local spatial variation (e.g., texture and boundaries) on the token grid, providing a correction to purely semantic scoring.

For each frame, FAR reshapes X_t to a token grid $G_t \in \mathbb{R}^{h_t \times w_t \times d}$ with $h_t w_t = N$, and compresses channels into $Y_t \in \mathbb{R}^{h_t \times w_t}$ by averaging the feature dimension. For a local block B of size $h_B \times w_B$ in this grid, the 2D DCT-II coefficient is

$$C_t^B(u,v) = \rho_u^{h_B} \rho_v^{w_B} \sum_{p=0}^{h_B-1} \sum_{q=0}^{w_B-1} Y_t^B(p,q) \cos \frac{\pi(2p+1)u}{2h_B} \cos \frac{\pi(2q+1)v}{2w_B}, \tag{1}$$

where $\rho_0^h = \sqrt{1/h}$, $\rho_u^h = \sqrt{2/h}$ for $u > 0$, and analogously for the width dimension. FAR uses non-overlapping 4×4 blocks, and a border block is shifted inward to keep its full size. Let $B(i)$ be the block containing token $x_{t,i}$. FAR uses the normalized AC energy

$$D_{t,i} = \mathrm{Norm}_{[0,1]} \left(\frac{1}{|B(i)|} \sum_{(u,v) \neq (0,0)} \left(C_t^{B(i)}(u,v) \right)^2 \right), \tag{2}$$

where $\mathrm{Norm}_{[0,1]}$ denotes frame-wise min–max normalization. This frequency cue favors texture and boundaries while remaining a scoring feature rather than an independent selector.

For each token $x_{t,i}$, FAR combines complementary signals:

$$S_{t,i} = w_{\mathrm{vis}} A_{t,i} + w_{\mathrm{dct}} D_{t,i} + w_{\mathrm{text}} T_{t,i}. \tag{3}$$

To avoid overreacting to ambiguous prompts, FAR applies a semantic trust region when text guidance is enabled. Let $S_{t,i}^{\mathrm{vis}} = (w_{\mathrm{vis}} A_{t,i} + w_{\mathrm{dct}} D_{t,i})/(w_{\mathrm{vis}} + w_{\mathrm{dct}})$. FAR uses a centered residual update

$$S_{t,i} = S_{t,i}^{\mathrm{vis}} + \gamma_t \left(T_{t,i} - \frac{1}{N} \sum_{j=1}^{N} T_{t,j} \right), \qquad \gamma_t \in [\gamma_{\min}, \gamma_{\max}]. \tag{4}$$

When text guidance is enabled ($w_{\mathrm{text}} > 0$), Eq. 4 replaces Eq. 3, and the scale of the text term is given by γ_t. The gain γ_t is entropy-aware. The concentrated text relevance indicates a reliable query signal and receives stronger guidance, while diffuse relevance is down-weighted.

Focus–Ambient Partition. FAR partitions each frame by score. The *focus* set $\mathcal{F}_t$ contains the highest-scoring tokens selected for direct retention, and the *ambient pool* is $\mathcal{E}_t = X_t \setminus \mathcal{F}_t$. *Focus tokens* preserve task-critical evidence in every frame, while the remaining *ambient tokens* are passed to the temporal cache for redundancy-aware filtering.

3.3 Cache: Temporal Ambient Matching

After Scoring, the key challenge is to separate changed *ambient evidence* from redundant background without extra motion models. FAR treats *ambient tokens* as temporal state, promoting cache-deviating tokens while preserving stable context. Motion-aware refresh allows the cache to remain stable in static scenes and update faster under scene change.

The Cache module removes temporal redundancy from the *ambient stream*. It maintains a memory M_{t-1} that stores the full, variable-length snapshot of normalized *ambient* features from the previous frame. For each current *ambient token* $x_{t,i} \in \mathcal{E}_t$, FAR computes its nearest cached similarity

$$c_{t,i} = \max_{m_j \in M_{t-1}} \cos(x_{t,i}, m_j). \tag{5}$$

A token is treated as shifted *ambient evidence* if $c_{t,i} < \tau_t$ and stable evidence otherwise. Shifted tokens are promoted because they may contain newly appearing context or changed background. Stable tokens are deprioritized and used only as optional backfill during Assembly.

The threshold τ_t is motion-aware. FAR estimates frame change with a low-cost cosine gap between mean frame features and optionally smooths it with short- and long-horizon exponential averages. Given motion proxy g_t and clipping function clip, the matching threshold is

$$\tau_t = \mathrm{clip}(\tau_0 + a_g g_t, \tau_{\min}, \tau_{\max}). \tag{6}$$

Nevertheless, more frequent motion raises the threshold, making dynamic *ambient tokens* harder to classify as stable and reducing stale-background reuse. To address this issue, after processing each frame, FAR refreshes the cache with a soft time-decay update rather than hard replacement, given by

$$m_{t,i} = \mathrm{normalize}\left(\alpha_t \, m_{j^\star(i)} + (1 - \alpha_t) \, \hat{x}_{t,i} \right), \qquad j^\star(i) = \arg \max_{m_j \in M_{t-1}} \cos(x_{t,i}, m_j), \tag{7}$$

where $\hat{x}_{t,i}$ is the ℓ_2-normalized *ambient token* and $M_t = \{m_{t,i}\}_{x_{t,i} \in \mathcal{E}_t}$ holds one entry per current *ambient* candidate. An empty cache is initialized with the normalized tokens. The decay coefficient is also motion-aware, given by

$$\alpha_t = \mathrm{clip}\left(\alpha_0 + k(1 - g_t), \alpha_{\min}, \alpha_{\max}\right). \tag{8}$$

By using Eq. 8, the scenes with less frequent motion receive larger α_t and preserve stable background memory, while the scenes with more frequent motions receive smaller α_t and incorporate new context.

3.4 Assembly: Budgeted Sequence Construction

The proposed Assembly component is supposed to satisfy the target budget without collapsing the retained context into redundant *focus tokens*. FAR combines *focus tokens* and changed *ambient evidence* with retention-aware allocation and diversity-aware selection. The resulting sequence is compact, position-preserving, and ready for LLM prefill.

The Assembly module constructs the compact visual sequence actually consumed by the LLM. For fair comparison, FAR follows the same per-frame visual token budget protocol as FlashVID [15], where the target retention ratio determines the maximum number of retained visual tokens per frame, with the same optional expansion and token-cap settings used in matched budget evaluation. When truncating or backfilling Ambient candidates, FAR uses an MMR-style value [6]:

$$V_{t,i} = \bar{A}_{t,i} - \lambda \max_{x_j \in \mathcal{A}_t} \cos(x_{t,i}, x_j), \tag{9}$$

where $\bar{A}_{t,i}$ is the salience $A_{t,i}$ min–max normalized over the candidates, and $\mathcal{A}_t$ denotes already selected anchors, typically *focus tokens*, and, during stable backfill, selected shifted tokens. The anchors are fixed within each stage, which performs a single TopK. This discourages *ambient* selections from clustering around evidence that has already been retained, improving scene coverage under the same budget.

Finally, FAR rewrites the multimodal input before LLM prefill. Retained frame-local indices are mapped back to global visual positions, sorted in their original order, and used to gather the corresponding entries in the embedding sequence. Auxiliary tensors, including position IDs, attention masks, and cache positions, are gathered with the same global indices. Thus the base LLM receives a shorter but positionally consistent multimodal sequence, and the discarded visual tokens never materialize in downstream prefill activations or KV-cache entries.

4 Experiments

4.1 Experimental Setup

Baselines. The proposed FAR is compared under matched visual token budgets with the state-of-the-art training-free video token compression method FlashVID [15] and inner-LLM pruning method FastV [7]. For fair evaluation, FastV uses pruning at LLM layer 20 to match the pruning-layer setting used by FlashVID and FAR. For broader comparison, FAR is further compared with three pre-LLM visual token compression methods, namely, DivPrune [2], PruneVid [21], and DyCoKe [43], which are integrated as controlled adapters. The base models include Qwen2.5-VL-3/7/32B [3], and GLM-4.6V-Flash-10B [19].

Datasets & Evaluation Metrics. Five video understanding benchmarks, namely, VideoMME [16], EgoSchema [32], LongVideoBench [48], MVBench [25], and MLVU [61], are used to evaluate the performance. Retention ratios are denoted by R and use the same visual token budget definition for all methods. We report $R \in \{1\%, 5\%, 10\%, 15\%, 20\%, 25\%\}$. The main metric is task score; *Avg. Score* averages the reported datasets within the table scope, and *Avg. Acc* is normalized by FlashVID at the same retention ratio unless noted otherwise. Efficiency is reported through retained visual tokens, average VRAM, and peak VRAM. Peak VRAM is the maximum allocated memory within each dataset, averaged over the datasets in the table scope; Avg. VRAM is the mean sampled allocated memory.

Implementation Details. FAR uses base cache-matching threshold $\tau_0 = 0.80$ (motion coefficient $a_g = 0.10$, clipped to $[0.72, 0.88]$), base cache-refresh coefficient $\alpha_0 = 0.75$ ($k = 0.20$, clipped to $[0.55, 0.95]$), MMR diversity coefficient $\lambda = 0.20$, text-guidance gain range $[\gamma_{\min}, \gamma_{\max}] =$

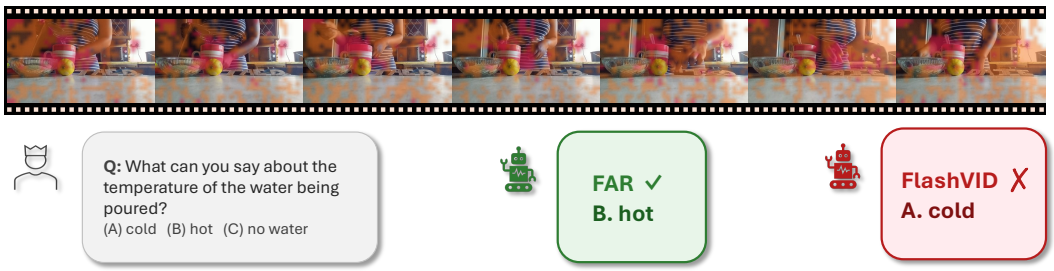


Figure 3: Qualitative long video QA example. FAR preserves task-relevant *Focus* evidence together with changed *Ambient* context, yielding a compact visual sequence that maintains the information needed for answer selection.

Table 1: Qwen2.5-VL-7B score and memory comparison under aligned retention ratios. Avg. Acc is normalized by FlashVID at the same retention ratio. Peak VRAM is in GiB. For fair evaluation, all the experiments are conducted on a single H100 GPU with 94 GB memory. The compared methods use the default configurations.

Retention	Method	VideoMME	EgoSchema	MVBench	LongVideoBench	MLVU	Avg. Score	Avg. Acc	Peak VRAM
1%	FastV [7]	41.370	35.062	38.300	40.464	21.336	35.306	74.1	51.90
	FlashVID [15]	49.444	**49.075**	**55.900**	**48.616**	**35.298**	**47.667**	100.0	36.46
	FAR	**49.593** (↑0.149)	48.678 (↓0.397)	54.200 (↓1.700)	46.672 (↓1.944)	33.806 (↓1.492)	46.590 (↓1.077)	97.7 (↓2.3)	**18.02** (↓50.58%)
5%	FastV [7]	50.185	44.186	50.625	43.381	28.490	43.373	80.4	51.90
	FlashVID [15]	**56.074**	54.303	**63.925**	**54.824**	40.699	**53.965**	100.0	36.46
	FAR	55.000 (↓1.074)	**54.521** (↑0.218)	63.900 (↓0.025)	53.328 (↓1.496)	**41.112** (↑0.413)	53.572 (↓0.393)	99.3 (↓0.7)	**18.02** (↓50.58%)
10%	FastV [7]	53.926	48.618	57.325	46.896	35.902	48.533	87.4	51.90
	FlashVID [15]	**57.852**	55.893	**65.350**	**57.068**	41.590	**55.551**	100.0	36.46
	FAR	57.407 (↓0.445)	**55.933** (↑0.040)	65.275 (↓0.075)	55.423 (↓1.645)	**41.933** (↑0.343)	55.194 (↓0.357)	99.4 (↓0.6)	**18.02** (↓50.58%)
15%	FastV [7]	55.593	50.049	59.575	49.963	37.763	50.588	89.8	51.90
	FlashVID [15]	**58.852**	56.390	66.600	57.218	42.669	56.346	100.0	36.46
	FAR	58.370 (↓0.482)	**56.529** (↑0.139)	**66.900** (↑0.300)	**57.666** (↑0.448)	**43.196** (↑0.527)	**56.532** (↑0.186)	100.3 (↑0.3)	**18.02** (↓50.58%)
20%	FastV [7]	56.148	51.142	60.475	51.384	37.557	51.341	90.2	51.90
	FlashVID [15]	**59.704**	56.927	**67.200**	**58.040**	**42.652**	**56.905**	100.0	36.46
	FAR	59.333 (↓0.371)	**57.086** (↑0.159)	66.925 (↓0.275)	57.442 (↓0.598)	41.486 (↓1.166)	56.454 (↓0.451)	99.2 (↓0.8)	**18.02** (↓50.58%)
25%	FastV [7]	57.074	50.745	60.900	52.132	37.265	51.623	90.7	51.90
	FlashVID [15]	59.889	57.145	67.250	58.414	41.898	56.919	100.0	36.46
	FAR	**60.667** (↑0.778)	**57.264** (↑0.119)	**67.450** (↑0.200)	**58.639** (↑0.225)	**42.714** (↑0.816)	**57.347** (↑0.428)	100.8 (↑0.8)	**18.02** (↓50.58%)

$[0.25, 0.85]$, and candidate expansion factor 1.25. All these hyper-parameters keep the same values across all the datasets and base models, indicating that FAR is stable without dataset- or backbone-specific tuning. For the scoring mixture, we use $(w_{\text{vis}}, w_{\text{dct}}, w_{\text{text}}) = (0.6, 0.3, 0.1)$ by default. Their sensitivity analysis is separately studied in Sec. 4.6. We use deterministic decoding and the same prompt, parser, and frame-processing pipeline within each backbone.

4.2 Accuracy–Memory Trade-off

Table 1 evaluates FAR on Qwen2.5-VL-7B, and aims to answer *what accuracy–memory trade-off does FAR achieve against FlashVID and FastV?* Its accuracy improves with larger retention (e.g., 15%, 20%, and 25%). In the highest reported retention 25%, FAR improves every listed dataset, yielding a gain of 0.816 points on MLVU and 0.428 points on average score. Nevertheless, FastV's accuracy remains below FlashVID and FAR on the row-level average (e.g., 1%, 5%, and 10%), indicating that early-layer LLM pruning is not a drop-in substitute for prefill-time video token compression. Across the sweep, FAR consistently uses substantially less peak VRAM than FlashVID [15]. Overall, FAR shows a more favorable accuracy–memory trade-off.

Fig. 3 provides a qualitative video question and answer example. It highlights the intended behavior of the two tracks. *Focus* retains answer-changing local evidence, while *Ambient* keeps scene context mainly when it changes sufficiently relative to the temporal cache. This complements the aggregate scores by showing how FAR preserves both discriminative details and non-repeated contextual anchors under a compact visual budget.

4.3 Comparison with Pre-LLM Token Compression Methods

Table 2 further compares FAR with three pre-LLM methods, namely, DivPrune [2], PruneVid [21], and DyCoKe-TTM [43], on Qwen2.5-VL-7B. All methods share the same inputs and visual token

Table 2: Comparison with three pre-LLM methods on Qwen2.5-VL-7B under aligned retention ratios. The mean averages the five benchmarks. For fair evaluation, all the experiments are conducted on a single H100 GPU with 94 GB memory. Unless noted otherwise, the compared methods use the default configurations.

Retention	Method	VideoMME	EgoSchema	MVBench	LongVideoBench	MLVU	Mean
1%	FAR	**49.593** (↑1.112)	**48.678** (↑1.232)	54.200 (↓0.725)	46.672 (↓1.720)	33.806 (↓3.456)	46.590 (↓0.636)
	DivPrune [2]	48.481	47.446	**54.925**	48.018	**37.262**	**47.226**
	PruneVid [21]	47.519	47.346	53.950	**48.392**	34.621	46.366
	DyCoKe-TTM [43]	43.074	41.046	44.025	43.605	28.607	40.071
5%	FAR	**55.000** (↑1.963)	**54.521** (↑2.643)	**63.900** (↑2.525)	53.328 (↓0.748)	**41.112** (↑1.149)	**53.572** (↑1.943)
	DivPrune [2]	53.037	51.620	61.375	53.254	38.176	51.492
	PruneVid [21]	50.852	51.878	61.375	**54.076**	39.963	51.629
	DyCoKe-TTM [43]	44.926	46.372	47.200	43.904	31.007	42.682
10%	FAR	**57.407** (↑1.370)	**55.933** (↑2.305)	**65.275** (↑1.400)	55.423 (↓0.972)	**41.933** (↑1.339)	**55.194** (↑1.522)
	DivPrune [2]	56.037	53.628	63.875	54.226	40.594	53.672
	PruneVid [21]	52.259	53.628	63.300	**56.395**	39.596	53.036
	DyCoKe-TTM [43]	52.333	52.673	62.400	53.029	37.719	51.631
15%	FAR	**58.370** (↑1.185)	**56.529** (↑2.007)	**66.900** (↑1.675)	**57.666** (↑1.495)	**43.196** (↑2.161)	**56.532** (↑1.734)
	DivPrune [2]	57.185	54.522	65.225	56.021	41.035	54.798
	PruneVid [21]	53.630	54.383	64.375	56.171	40.228	53.757
	DyCoKe-TTM [43]	54.148	53.806	63.850	53.852	38.836	52.898
20%	FAR	**59.333** (↑2.296)	**57.086** (↑1.670)	**66.925** (↑1.150)	**57.442** (↑0.972)	41.486 (↓0.348)	**56.454** (↑1.237)
	DivPrune [2]	57.037	55.416	65.775	56.021	**41.834**	55.217
	PruneVid [21]	54.444	55.079	65.250	56.470	39.761	54.201
	DyCoKe-TTM [43]	54.519	54.522	64.775	53.702	39.265	53.357
25%	FAR	**60.667** (↑2.297)	**57.264** (↑1.152)	**67.450** (↑1.450)	**58.639** (↑1.047)	**42.714** (↑0.607)	**57.347** (↑1.311)
	DivPrune [2]	58.370	56.112	66.000	57.592	42.107	56.036
	PruneVid [21]	54.963	55.794	65.375	57.218	40.369	54.744
	DyCoKe-TTM [43]	55.111	54.860	64.700	53.702	39.592	53.593

Table 3: GLM-4.6V-Flash score comparison under aligned retention ratios. Avg. Acc and FAR deltas are computed against FastV at the same retention ratio. For fair evaluation, all the experiments are conducted on a single H100 GPU with 94 GB memory. The compared methods use the default configurations.

Retention	Method	VideoMME	EgoSchema	LongVideoBench	MLVU	Avg. Score	Avg. Acc
1%	FastV [7]	44.333	48.280	46.148	34.315	43.269	100.0
	FAR	**48.148** (↑3.815)	**51.679** (↑3.399)	**51.159** (↑5.011)	**35.750** (↑1.435)	**46.684** (↑3.415)	107.9 (↑7.9)
5%	FastV [7]	**52.741**	49.612	49.888	36.824	47.266	100.0
	FAR	51.593 (↓1.148)	**52.892** (↑3.280)	**54.450** (↑4.562)	**39.798** (↑2.974)	**49.683** (↑2.417)	105.1 (↑5.1)
10%	FastV [7]	**57.037**	**56.151**	51.458	40.819	51.366	100.0
	FAR	54.407 (↓2.630)	54.859 (↓1.292)	**56.470** (↑5.012)	**41.810** (↑0.991)	**51.886** (↑0.520)	101.0 (↑1.0)
15%	FastV [7]	**59.370**	**59.153**	52.954	**42.936**	**53.603**	100.0
	FAR	55.333 (↓4.037)	56.191 (↓2.962)	**57.292** (↑4.338)	41.479 (↓1.457)	52.574 (↓1.029)	98.1 (↓1.9)
20%	FastV [7]	**60.556**	**60.047**	54.525	**43.836**	**54.741**	100.0
	FAR	56.185 (↓4.371)	57.304 (↓2.743)	**58.489** (↑3.964)	42.444 (↓1.392)	53.605 (↓1.136)	97.9 (↓2.1)
25%	FastV [7]	**61.556**	**60.882**	56.021	**44.531**	**55.747**	100.0
	FAR	57.000 (↓4.556)	57.901 (↓2.981)	**58.564** (↑2.543)	43.056 (↓1.475)	54.130 (↓1.617)	97.1 (↓2.9)

budget; DyCoKe uses its prefill-time temporal token merging (TTM) only. Within this tier, FAR gives the highest five-benchmark mean accuracy from 5% to 25% retention, by 1.24–1.94 points over the second-best method, and is 0.64 points below DivPrune at 1% retention. The comparison thus separates the gain of the proposed FAR from the common memory-saving property that is shared by the state-of-the-art pre-LLM methods.

4.4 Cross-Backbone Evaluation

Table 3 reports GLM-4.6V-Flash-10B scores using FastV as the reference. Under all retention ratios, FAR shows superior task scores over FastV on LongVideoBench. Under smaller retention ratios (e.g., 1%, 5%, and 10%), FAR shows higher Avg. Score over FastV, particularly on LongVideoBench and MLVU. However, under five out of six retention ratios on VideoMME, its performance is inferior to FastV.

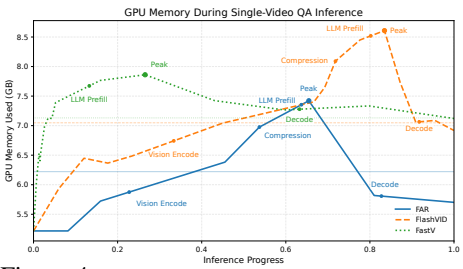


Figure 4: Phase-wise GPU memory during inference on Qwen2.5-VL-3B. FAR compresses and rewrites the visual sequence before LLM prefill, reducing memory residency in the prefill stage and preserving the reduction during decoding under the same inference pipeline.

Table 4: Qwen2.5-VL scale results under available coverage for VideoMME, EgoSchema, LongVideoBench, and MLVU. Avg./Peak VRAM are in GiB. '*': for Qwen2.5-VL-32B, the memory columns of both methods are averaged over EgoSchema and MLVU, where both methods run. For fair evaluation, all the experiments are conducted on a single H100 GPU with 94 GB memory.

Model	Retention	Method	VideoMME	EgoSchema	LongVideoBench	MLVU	Avg. VRAM	Peak VRAM
Qwen2.5-VL-3B	1%	FlashVID [15]	45.000	37.348	**47.195**	**29.939**	7.12	17.11
		FAR	**45.407** (↑0.407)	**38.958** (↑1.610)	46.896 (↓0.299)	29.024 (↓0.915)	**6.28** (↓11.85%)	**8.82** (↓48.45%)
	5%	FlashVID [15]	49.296	43.649	**50.112**	**34.701**	7.12	17.11
		FAR	**49.370** (↑0.074)	**43.788** (↑0.139)	49.140 (↓0.972)	33.715 (↓0.986)	**6.29** (↓11.72%)	**8.82** (↓48.45%)
	10%	FlashVID [15]	**50.593**	**45.736**	**51.085**	**36.002**	7.13	17.11
		FAR	50.481 (↓0.112)	45.637 (↓0.099)	49.888 (↓1.197)	35.980 (↓0.022)	**6.29** (↓11.73%)	**8.82** (↓48.45%)
	15%	FlashVID [15]	**51.259**	47.088	**51.159**	35.903	7.13	17.11
		FAR	**51.259** (↑0.000)	**47.326** (↑0.238)	50.561 (↓0.598)	**36.818** (↑0.915)	**6.29** (↓11.73%)	**8.82** (↓48.45%)
	20%	FlashVID [15]	**51.519**	47.823	51.458	**38.532**	7.13	17.11
		FAR	51.111 (↓0.408)	**47.903** (↑0.080)	**51.832** (↑0.374)	37.853 (↓0.679)	**6.30** (↓11.73%)	**8.82** (↓48.45%)
	25%	FlashVID [15]	51.519	**48.618**	51.907	37.866	7.13	17.11
		FAR	**51.963** (↑0.444)	48.439 (↓0.179)	**52.655** (↑0.748)	**38.259** (↑0.393)	**6.30** (↓11.71%)	**8.82** (↓48.45%)
Qwen2.5-VL-32B	1%	FlashVID [15]	OOM	48.081	OOM	11.121	62.55*	78.76*
		FAR	**47.667**	**49.453** (↑1.372)	**47.120**	**11.911** (↑0.790)	**61.55**	**64.50** (↓18.11%)
	5%	FlashVID [15]	OOM	**55.217**	OOM	**25.115**	62.58*	78.81*
		FAR	**53.111**	54.720 (↓0.497)	**49.215**	24.586 (↓0.529)	**61.59**	**64.50** (↓18.16%)
	10%	FlashVID [15]	OOM	58.656	OOM	**31.342**	62.61*	78.87*
		FAR	**56.593**	**59.749** (↑1.093)	**51.683**	30.864 (↓0.478)	**61.62**	**64.50** (↓18.22%)
	15%	FlashVID [15]	OOM	60.286	OOM	**34.783**	62.64*	78.92*
		FAR	**57.037**	**60.862** (↑0.576)	**52.132**	32.564 (↓2.219)	**61.65**	**64.50** (↓18.28%)
	20%	FlashVID [15]	OOM	61.339	OOM	**37.360**	62.67*	78.98*
		FAR	**58.926**	**61.935** (↑0.596)	**53.104**	34.838 (↓2.522)	**61.68**	**64.50** (↓18.34%)
	25%	FlashVID [15]	OOM	62.055	OOM	**36.241**	62.66*	78.63*
		FAR	**59.667**	**62.254** (↑0.199)	**53.777**	35.464 (↓0.777)	**61.70**	**64.49** (↓17.98%)

Table 5: Per-component ablation on Qwen2.5-VL-7B across two VideoMME settings. Differences are computed against Full FAR.

Variant	Scoring	Cache	Assembly	VideoMME (25%)	VideoMME (5%)
Full FAR	✓	✓	✓	**60.667**	**54.926**
w/o Scoring	×	✓	✓	60.148 (↓0.519)	**54.926** (0.000)
w/o Cache	✓	×	✓	60.407 (↓0.259)	54.778 (↓0.148)
w/o Assembly	✓	✓	×	60.185 (↓0.481)	54.556 (↓0.370)

Table 6: Scoring-weight ablation on Qwen2.5-VL-7B VideoMME at 25% retention.

Setting	w_v	w_d	w_t	Score	Δ
default FAR	0.6	0.3	0.1	**60.667**	0.000
balanced visual	0.5	0.4	0.1	60.481	-0.185
high visual	0.7	0.2	0.1	60.333	-0.333
CLS-only	1.0	0.0	0.0	60.148	-0.519
DCT-only	0.0	1.0	0.0	59.963	-0.704
text-only	0.0	0.0	1.0	60.370	-0.296

4.5 Scale Coverage

Table 4 reports smaller- and larger-backbone coverage on four benchmarks. At all measured retention ratios, Qwen2.5-VL-3B has mixed quality deltas but consistent memory reduction for FAR. For Qwen2.5-VL-32B, FlashVID runs out of memory on VideoMME and LongVideoBench, so those cells are marked OOM. For such scenarios, the memory columns of both methods are averaged over EgoSchema and MLVU, where both run; FAR itself runs on all five benchmarks with a peak of at most 65.2 GiB. FAR outperforms FlashVID on EgoSchema under five out of six retention ratios. It is competitive to or even outperforms FlashVID on MLVU under three out of six retention ratios.

Fig. 4 further reports the phase-wise GPU-memory trace during inference. The memory gap appears after FAR rewrites the visual sequence before LLM prefill, and remains visible through the subsequent decode stage. This supports the mechanism in Sec. 3.4. Discarded visual tokens do not allocate downstream prefill activations or KV-cache entries, so the retained-token budget translates into lower inference-time memory residency rather than only a smaller input count. The overall peak of pre-LLM methods is reached earlier during vision encoding (Sec. 4.3); a per-phase breakdown is given in Appendix C.5.

4.6 Ablation Studies

Impact of Each Component. Table 5 reports a breakdown analysis on the three FAR modules (in Sec. 3), namely, Scoring, Cache, and Assembly. The scoring ablation keeps a CLS-only ranking signal because FAR still needs salience to partition *focus* and *ambient* tokens. The cache ablation removes temporal matching and time-decay refresh while preserving the same final budget. The assembly ablation replaces staged *focus*/*ambient* composition with score top-k within each temporal

group under the same per-group budget, while retaining the original-order rewrite required by the backbone interface. Removing Assembly lowers the score in both settings, by 0.370–0.481 points. Removing Cache lowers the score in both settings, by 0.148–0.259 points. Removing Scoring has setting-dependent effects, from 0.519 to 0.000 points. All variants cost nearly the same memory (Appendix C.2).

Impact of Scoring Weights. Table 6 shows that, on VideoMME at 25% retention, the default mixture $(w_{\text{vis}}, w_{\text{dct}}, w_{\text{text}}) = (0.6, 0.3, 0.1)$ gives the best score among the measured settings, and CLS-only, DCT-only, and text-only scoring are 0.296–0.704 points lower. These results clarify the roles of DCT and text guidance. A nontrivial DCT term is useful, but the DCT-only row remains below the default. DCT therefore acts as a visual-variation term that counterbalances semantic over-concentration by exposing local spatial variation. Text-only scoring is competitive in this VideoMME run but still below the default mixture, which keeps text as a light correction rather than the main ranking signal. These results match the design principle behind FAR scoring, which jointly incorporates visual-attention-dominant, DCT-assisted, and lightly text-calibrated selection.

4.7 Limitations

FAR does not improve accuracy in every setting. On Qwen2.5-VL-7B, its average score is below FlashVID at 1%, 5%, 10%, and 20% retention (Table 1). On Qwen2.5-VL-32B, it trails FlashVID on MLVU under five of six ratios (Table 4), and on GLM-4.6V-Flash it is below FastV on VideoMME under five of six ratios (Table 3). At 1% retention on LongVideoBench, removing Cache or Assembly lowers the overall score by 0.299 or 0.673 points, yet raises the performance of some individual question categories by up to 4.17 or 7.41 points (Appendix C.3).

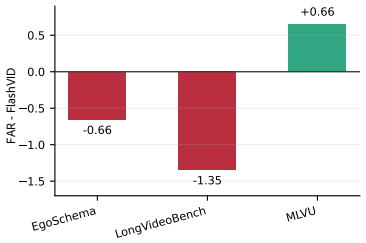


Figure 5: LLaVA-OneVision-Qwen2-7B-OV limitation evidence at 25% retention on EgoSchema, LongVideoBench, and MLVU.

FAR also depends on positional compatibility. It performs sparse token-level retention, so each retained visual token should preserve its original temporal, height, and width coordinates after compression. This condition is natural for Qwen2.5-VL-style MRoPE [46, 3] and GLM-style 2D/3D RoPE [42, 19], where the implementation can gather matching position IDs for retained tokens. It is less natural for flattened one-dimensional or embed-only layouts, where sparse tokens may be reindexed, distant frames can become adjacent, and spatial relations can be weakened. Fig. 5 gives a conservative compatibility example on LLaVA-OneVision-Qwen2-7B-OV [23]. At matched retention, FAR improves on MLVU but trails on EgoSchema and LongVideoBench. FAR can be integrated through three paths: gathered multidimensional positions on Qwen2.5-VL and GLM-4.6V, flattened-sequence rewriting on LLaVA models, and embedding-level rewriting on InternVL3.5 (Appendix A.3); only the first path preserves the original positions. In addition, scoring, cache matching, and MMR assembly add compression latency before prefill, so lower memory does not imply lower latency (Appendix C.5).

5 Conclusion

We presented Focus–Ambient Retention (FAR), a training-free method for memory-efficient video large language models. The proposed FAR separates focus evidence from ambient evidence, refreshes ambient context through cache matching, and assembles a diversity-aware token budget before language-model prefill. This design targets the central bottleneck of long video VLLMs, which retains task-relevant visual evidence while reducing redundant visual tokens, memory pressure, and prefill cost. Experiments on five video understanding benchmarks showed that FAR improved the accuracy–memory trade-off, reaching accuracy comparable to the state-of-the-art while reducing peak VRAM by up to 50.6% under matched retained-token budgets.

Future Work & Societal Impact. Future work will improve the generalizability of position-preserving adapters in more video LLMs. More efficient long video inference can make multimodal analysis more accessible, but may amplify risks in privacy-sensitive video understanding. Therefore, practical deployment should include privacy safeguards, and clear limits on high-stakes use.

References

[1] Nasir Ahmed, T_ Natarajan, and Kamisetty R Rao. Discrete cosine transform. *IEEE transactions on Computers*, 100(1):90–93, 1974.

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

[3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[4] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *International Conference on Learning Representations*, 2023.

[5] Mu Cai, Jianwei Yang, Jianfeng Gao, and Yong Jae Lee. Matryoshka multimodal models. In *International Conference on Learning Representations*, 2025.

[6] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336, 1998.

[7] 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. In *Proceedings of the European Conference on Computer Vision*, pages 19–35, 2024.

[8] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2024.

[9] Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *European conference on computer vision*, pages 640–658, 2022.

[10] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. *Advances in neural information processing systems*, 34:11781–11794, 2021.

[11] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

[12] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems*, pages 49250–49267, 2023.

[13] Anxhelo Diko, Tinghuai Wang, Wassim Swaileh, Shiyan Sun, and Ioannis Patras. Rewind: Understanding long videos with instructed learnable memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13734–13743, 2025.

[14] Yue Fan, Xiaojian Ma, Rujie Wu, Yuntao Du, Jiaqi Li, Zhi Gao, and Qing Li. Videoagent: A memory-augmented multimodal agent for video understanding. In *European Conference on Computer Vision*, pages 75–92, 2024.

[15] Ziyang Fan, Keyu Chen, Ruilong Xing, Yulin Li, Li Jiang, and Zhuotao Tian. FlashVID: Efficient video large language models via training-free tree-based spatiotemporal token merging. In *The Fourteenth International Conference on Learning Representations*, 2026.

[16] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24108–24118, 2025.

[17] Tianyu Fu, Tengxuan Liu, Qinghao Han, Guohao Dai, Shengen Yan, Huazhong Yang, Xuefei Ning, and Yu Wang. Framefusion: Combining similarity and importance for video token reduction on large visual language models. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22654–22663, 2025.

[18] Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. Ma-lmm: Memory-augmented large multimodal model for long-term video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13504–13514, 2024.

[19] Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, et al. Glm-4.5 v and glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025.

[20] Wenbo Hu, Zi-Yi Dou, Liunian H Li, Amita Kamath, Nanyun Peng, and Kai-Wei Chang. Matryoshka query transformer for large vision-language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 50168–50188, 2024.

[21] Xiaohu Huang, Hao Zhou, and Kai Han. Prunevid: Visual token pruning for efficient video large language models. In *Findings of the Association for Computational Linguistics*, pages 19959–19973, 2025.

[22] Jeongseok Hyun, Sukjun Hwang, Su Ho Han, Taeoh Kim, Inwoong Lee, Dongyoon Wee, Joon-Young Lee, Seon Joo Kim, and Minho Shim. Multi-granular spatio-temporal token merging for training-free acceleration of video llms. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23990–24000, 2025.

[23] 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. *Transactions on Machine Learning Research*, 2025.

[24] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*, pages 19730–19742, 2023.

[25] 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*, pages 22195–22206, 2024.

[26] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *Proceedings of the European Conference on Computer Vision*, pages 323–340, 2024.

[27] Yulin Li, GUI Haokun, Ziyang Fan, Junjie Wang, Bin Kang, BIN CHEN, and Zhuotao Tian. Less is more, but where? dynamic token compression via llm-guided keyframe prior. *Advances in Neural Information Processing Systems*, 2025.

[28] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in Neural Information Processing Systems*, 36:34892–34916, 2023.

[29] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024.

[30] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. URL `https://llava-vl.github.io/blog/2024-01-30-llava-next/`.

[31] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12585–12602, 2024.

[32] 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.

[33] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9226–9235, 2019.

[34] Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. Streaming long video understanding with large language models, 2024.

[35] 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*, pages 8748–8763, 2021.

[36] Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22857–22867, 2025.

[37] Kele Shao, TAO Keda, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. Holitom: Holistic token merging for fast video large language models. *Advances in Neural Information Processing Systems*, 2025.

[38] Leqi Shen, Guoqiang Gong, Tao He, Yifeng Zhang, Sicheng Zhao, and Guiguang Ding. Fastvid: Dynamic density pruning for fast video large language models. *Advances in Neural Information Processing Systems*, 2025.

[39] 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. In *International Conference on Machine Learning*, pages 54582–54599, 2025.

[40] Yixian Shen, Qi Bi, Jia-Hong Huang, Hongyi Zhu, Andy D Pimentel, and Anuj Pathania. Macp: Minimal yet mighty adaptation via hierarchical cosine projection. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 20602–20618, 2025.

[41] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18221–18232, 2024.

[42] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[43] Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. Dycoke: Dynamic compression of tokens for fast video large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18992–19001, 2025.

[44] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[46] 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.

[47] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

[48] Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37: 28828–28857, 2024.

[49] Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, et al. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *arXiv preprint arXiv:2410.17247*, 2024.

[50] Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See Kiong Ng, and Jiashi Feng. Pllava: Parameter-free llava extension from images to videos for video dense captioning, 2024.

[51] Cheng Yang, Yang Sui, Jinqi Xiao, Lingyi Huang, Yu Gong, Chendi Li, Jinghua Yan, Yu Bai, Ponnuswamy Sadayappan, Xia Hu, et al. Topv: Compatible token pruning with inference time optimization for fast and low-memory multimodal vision language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19803–19813, 2025.

[52] 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*, pages 19792–19802, 2025.

[53] Zongxin Yang and Yi Yang. Decoupling features in hierarchical propagation for video object segmentation. *Advances in Neural Information Processing Systems*, 35:36324–36336, 2022.

[54] Zongxin Yang, Yunchao Wei, and Yi Yang. Associating objects with transformers for video object segmentation. *Advances in Neural Information Processing Systems*, 34:2491–2502, 2021.

[55] Ce Zhang, Kaixin Ma, Tianqing Fang, Wenhao Yu, Hongming Zhang, Zhisong Zhang, Haitao Mi, and Dong Yu. Vscan: Rethinking visual token reduction for efficient large vision-language models. *Transactions on Machine Learning Research*, 2025.

[56] Qizhe Zhang, Aosong Cheng, Ming Lu, Renrui Zhang, Zhiyong Zhuo, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. Beyond text-visual attention: Exploiting visual cues for effective token pruning in vlms. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20857–20867, 2025.

[57] Qizhe Zhang, Mengzhen Liu, Lichen Li, Ming Lu, Yuan Zhang, Junwen Pan, Qi She, and Shanghang Zhang. Beyond attention or similarity: Maximizing conditional diversity for token pruning in mllms. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[58] Shaolei Zhang, Qingkai Fang, Zhe Yang, and Yang Feng. Llava-mini: Efficient image and video large multimodal models with one vision token. In *International Conference on Learning Representations*, 2025.

[59] 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*, pages 74840–74857, 2025.

[60] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun MA, Ziwei Liu, and Chunyuan Li. Llava-video: Video instruction tuning with synthetic data. *Transactions on Machine Learning Research*, 2024.

[61] Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, et al. Mlvu: Benchmarking multi-task long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13691–13701, 2025.

[62] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

Supplementary Material

Contents

A Implementation Details **16**

A.1 Focus–Ambient Retention Pipeline 16

A.2 Default Hyperparameters 16

A.3 Backbone-Specific Positional Handling 17

B Additional Method Diagnostics **18**

B.1 Cache Dynamics 18

B.2 Score Components and Memory Trace 19

B.3 Additional Qualitative Examples 20

C Additional Comparisons and Diagnostics **23**

C.1 Pre-LLM Baselines 23

C.2 Additional Ablations and Sensitivity 23

C.3 Low-Retention Categories 24

C.4 Evidence Routing Diagnostics 24

C.5 Memory Attribution and Runtime 26

This supplementary material provides implementation and diagnostic details for FAR.

Sec. A first describes the proposed Focus–Ambient retention pipeline, including score computation, Focus–Ambient partition, cache querying, budgeted assembly, and position-preserving sequence rewriting in Sec. A.1. It then summarizes the default hyperparameters in Sec. A.2 and backbone-specific positional handling in Sec. A.3.

Sec. B further provides method diagnostics, including cache-dynamics analysis in Sec. B.1, score-component and memory-trace diagnostics in Sec. B.2, and additional qualitative examples with token-routing visualizations (Figs. 13 and 14) in Sec. B.3.

Sec. C reports additional comparisons and diagnostics: ablations and sensitivity in Sec. C.2, low-retention categories in Sec. C.3, evidence-routing diagnostics in Sec. C.4, and memory attribution and runtime in Sec. C.5.

A Implementation Details

A.1 Focus–Ambient Retention Pipeline

FAR is applied after visual feature extraction and before language-model prefill. Let a video be represented as $X = \{X_t\}_{t=1}^{F}$, where $X_t \in \mathbb{R}^{N\times d}$ contains the visual tokens of frame t. The implementation returns both the retained embeddings and their original global visual indices, so that downstream sequence tensors can be rewritten without reindexing the visual coordinates. Table 7 lists the symbols.

The procedure is given in Algorithm 1 and proceeds as follows.

1. Compute a frame-wise score vector S_t from visual salience and DCT energy, with text relevance added through Eq. 4 when text guidance is enabled.
2. Select the highest-scoring tokens as the Focus set $\mathcal{F}_t$ and place the remaining tokens in the Ambient pool $\mathcal{E}_t$.
3. Query the Ambient cache with each token in $\mathcal{E}_t$. Tokens whose nearest cached similarity is below τ_t are treated as shifted Ambient evidence; the rest are stable candidates.
4. Construct a per-frame budget $B_t = \lceil NR\eta \rceil$, optionally clipped by a token cap, where R is the target retention ratio and η is the expansion factor.
5. Assemble Focus tokens, shifted Ambient tokens, and optional stable backfill under B_t. If Ambient candidates exceed the remaining budget, FAR truncates them by the salience-based MMR value of Eq. 9; if the budget is not filled, FAR backfills from stable Ambient tokens with the same diversity penalty.
6. Sort retained frame-local indices, map them to original global visual indices, concatenate across frames, and sort the final sequence by the original global order.
7. Gather the corresponding visual embeddings and auxiliary sequence tensors. Discarded visual tokens are not passed into the LLM prefill.

Note that cache matching is performed only on the Ambient pool, while Focus tokens are directly preserved as task evidence. The final sorting step preserves the temporal-spatial order expected by position-aware video-language backbones.

A.2 Default Hyperparameters

Unless otherwise stated, the main experiments use a single FAR configuration across datasets and backbones. Retention sweeps change the target retention ratio and the retention-derived Focus allocation, but do not tune scoring or cache hyperparameters per dataset.

For retention sweeps, the Focus budget is tied to R rather than held fixed. In the full FAR, the base Focus ratio is $0.32R$, with the adaptive lower and upper bounds set to $0.28R$ and $0.48R$, respectively: $\rho_t = \mathrm{clip}(0.32R + 0.06\, H_t^S + 0.04\,(1-\mathrm{IQR}_t),\ 0.28R,\ 0.48R)$, where H_t^S is the normalized entropy of the softmax over the min–max normalized frame scores and IQR_t is their interquartile range. At $R = 1\%$, the three values are rounded to $0.3R$, $0.3R$, and $0.5R$. This keeps the Focus and

Algorithm 1 Implementation of the proposed FAR

Require: $X \in \mathbb{R}^{F \times N \times d}$, score components (A_t, D_t, T_t), (R, η, τ, λ)
Ensure: $\hat{X}$, retained global indices $\hat{G}$

$\mathcal{Y}, \mathcal{G} \leftarrow \emptyset$; initialize M_0
for $t = 1$ to F **do**
  $B_t \leftarrow \mathrm{cap}(\lceil NR\eta \rceil), \quad K_t \leftarrow \lceil N\rho_t \rceil$
  $S_t \leftarrow (w_{\mathrm{vis}}\bar{A}_t + w_{\mathrm{dct}}\bar{D}_t)/(w_{\mathrm{vis}} + w_{\mathrm{dct}}) + \gamma_t(\bar{T}_t - \mathrm{mean}(\bar{T}_t))$
  $\mathcal{F}_t \leftarrow \mathrm{TopK}(S_t, K_t), \quad \mathcal{E}_t \leftarrow [N] \setminus \mathcal{F}_t$
  $\tau_t \leftarrow f_\tau(\tau, X_t, X_{t-1})$
  $s_i \leftarrow \max_{m \in M_{t-1}} \cos(X_{t,i}, m), \quad \forall i \in \mathcal{E}_t$
  $\mathcal{H}_t \leftarrow \{i \in \mathcal{E}_t : s_i < \tau_t\}, \quad \mathcal{P}_t \leftarrow \mathcal{E}_t \setminus \mathcal{H}_t$
  $\mathcal{K}_t \leftarrow \mathcal{F}_t \cup \mathcal{H}_t, \quad \mathcal{B}_t \leftarrow \emptyset$
  $V(i; \mathcal{C}) \leftarrow \bar{A}_t(i) - \lambda \max_{j \in \mathcal{C}} \cos(X_{t,i}, X_{t,j})$
  if $|\mathcal{K}_t| > B_t$ **then**
    $\mathcal{H}_t \leftarrow \mathrm{TopK}_{i \in \mathcal{H}_t}(V(i; \mathcal{F}_t), [B_t - |\mathcal{F}_t|]_+)$
  else if $|\mathcal{K}_t| < B_t$ **then**
    $\mathcal{B}_t \leftarrow \mathrm{TopK}_{i \in \mathcal{P}_t}(V(i; \mathcal{F}_t \cup \mathcal{H}_t), B_t - |\mathcal{K}_t|)$
  end if
  $\mathcal{I}_t \leftarrow \mathrm{sort}(\mathcal{F}_t \cup \mathcal{H}_t \cup \mathcal{B}_t)$; append $X_{t,\mathcal{I}_t}$ to $\mathcal{Y}$ and $g(t, \mathcal{I}_t)$ to $\mathcal{G}$
  $M_t \leftarrow \mathrm{Refresh}(M_{t-1}, X_{t,\mathcal{E}_t}, \alpha_t)$, with $m' = \mathrm{norm}(\alpha_t m_{\mathrm{nn}} + (1 - \alpha_t)x)$ if soft refresh is enabled
end for

return $\hat{X}, \hat{G} \leftarrow \mathrm{sort}_{\mathcal{G}}(\mathcal{Y}, \mathcal{G})$

Ambient tracks active across low- and moderate-retention settings; otherwise, a fixed Focus ratio could consume nearly the whole budget at small R and starve the Ambient track.

Table 8 maps each hyperparameter to its equation. The text gain in Eq. 4 is $\gamma_t = \gamma_{\min} + (\gamma_{\max} - \gamma_{\min})(1 - H_t)$, where H_t is the normalized entropy of the softmax over the frame-wise min–max normalized text relevance T_t; Eq. 4 is used whenever $w_{\mathrm{text}} > 0$. The motion proxy g_t in Eqs. 6 and 8 is the cosine gap between mean frame features, smoothed by a short and a long exponential average (rates 0.60 and 0.20) that are mixed with weights 0.60 and 0.40.

Unless otherwise noted, all comparisons use deterministic decoding and the same prompt, parser, and frame-processing pipeline within each backbone. The main reported experiments are conducted on a single H100 GPU with 94 GB memory. Peak VRAM in the tables is the maximum over the samples of a dataset, averaged over the datasets in the table scope; Avg. VRAM averages the sampled allocated memory in the same way. Qwen2.5-VL-3B uses 16 sampled frames, Qwen2.5-VL-7B/32B use 32, and GLM-4.6V-Flash uses 64. The Qwen2.5-VL and GLM encoders merge every two adjacent frames into one temporal group, and FAR operates on these encoder output units.

A.3 Backbone-Specific Positional Handling

FAR assumes that retained visual tokens can keep their original temporal and spatial coordinates after compression. The implementation therefore tracks original global visual indices throughout retention and gathers auxiliary tensors with the same index set used for embeddings. Table 9 summarizes the interface of each backbone.

For Qwen2.5-VL, FAR rewrites the multimodal sequence by replacing the dense visual span with the retained visual embeddings, then gathers position IDs, attention-related masks, and cache positions according to the corresponding global indices. This preserves MRoPE coordinates for retained tokens rather than assigning new compacted visual positions, and decoding continues from the original sequence length.

For GLM-4.6V-Flash, FAR preserves the model's 3D positional chain by gathering the appropriate entries of the position tensor together with the attention mask; decoding also continues from the original sequence length. These results demonstrate that the sequence-rewrite interface is portable.

Table 7: Symbols used in the FAR pipeline.

Symbol	Meaning
F	Number of encoder temporal units (frames or frame pairs)
N	Number of visual tokens per frame before retention
d	Visual token feature dimension
R	Target visual token retention ratio
η	Candidate expansion factor for matched budget evaluation
X_t	Visual tokens of frame t
A_t	Visual salience score vector (also the base value in MMR assembly)
D_t	Local DCT-energy score vector
T_t	Text-relevance score vector
S_t	Final fused score vector
$\mathcal{F}_t$	Focus tokens retained as direct evidence
$\mathcal{E}_t$	Ambient candidates queried against cache
M_t	Ambient cache after frame t
τ_t	Motion-aware cache matching threshold
α_t	Motion-aware cache refresh coefficient
λ	Diversity penalty in MMR assembly
B_t	Per-frame retained-token budget

Table 8: Default FAR hyperparameters used in the reported experiments.

Component	Hyperparameter	Value	Equation
Scoring	$(w_{\text{vis}}, w_{\text{dct}}, w_{\text{text}})$	$(0.6, 0.3, 0.1)$	Eqs. 3, 4
Text guidance	Gain range $[\gamma_{\min}, \gamma_{\max}]$	$[0.25, 0.85]$	Eq. 4
Focus allocation	Base ratio, bounds	$0.32R$, $[0.28R, 0.48R]$	Sec. A.2
Cache matching	τ_0, a_g, $[\tau_{\min}, \tau_{\max}]$	0.80, 0.10, $[0.72, 0.88]$	Eq. 6
Cache refresh	α_0, k, $[\alpha_{\min}, \alpha_{\max}]$	0.75, 0.20, $[0.55, 0.95]$	Eq. 8
Assembly	MMR diversity coefficient λ	0.20	Eq. 9
Budgeting	Expansion factor η	1.25	Sec. A.1
	LLM pruning layer, kept ratio	20, 0.3	Sec. 4.1

Table 9: Positional-interface criterion for before-LLM token retention. FAR preserves the original spatio-temporal coordinates only when positions can be gathered together with the embeddings. Temporal granularity: sampled frames $\rightarrow$ temporal steps $\times$ tokens per step (mean over VideoMME).

Interface	Requirement	Backbone	Temporal granularity	Salience source
Gatherable multi-dim.	retained tokens keep their original (t, h, w) positions	Qwen2.5-VL-3B	$16 \rightarrow 8 \times 665$ (2 frames/step)	derived [CLS] attention
		Qwen2.5-VL-7B/32B	$32 \rightarrow 16 \times 665$ (2 frames/step)	derived [CLS] attention
		GLM-4.6V-Flash	$64 \rightarrow 32 \times 914$ (2 frames/step)	frame-prototype proxy
Flattened 1D	positions re-indexed	LLaVA-OneVision-7B	$32 \rightarrow 32 \times 196$ (per frame)	SigLIP attention
		LLaVA-Video-7B	$64 \rightarrow 64 \times 169$ (per frame)	SigLIP attention
Embed-level	embeddings and mask rewritten; positions regenerated	InternVL3.5-8B/14B	$32 \rightarrow 32 \times 256$ (per frame)	frame-prototype proxy

For LLaVA-OneVision-style layouts, sparse token retention is less natural because the visual sequence can be flattened or consumed through an embed-level interface that does not provide the same position-preserving sparse-token path. Positions are re-indexed on the compressed sequence, so the original temporal and spatial distances are not preserved. For InternVL3.5, FAR rewrites `inputs_embeds` and `attention_mask`, and positions are regenerated on the compressed sequence.

B Additional Method Diagnostics

B.1 Cache Dynamics

Fig. 6 traces the Ambient cache over the eight routing steps. At the first step the cache is empty, so every Ambient token is treated as shifted. Once the cache is populated, the shifted fraction drops and then varies with scene changes, and the number of retained tokens follows it.

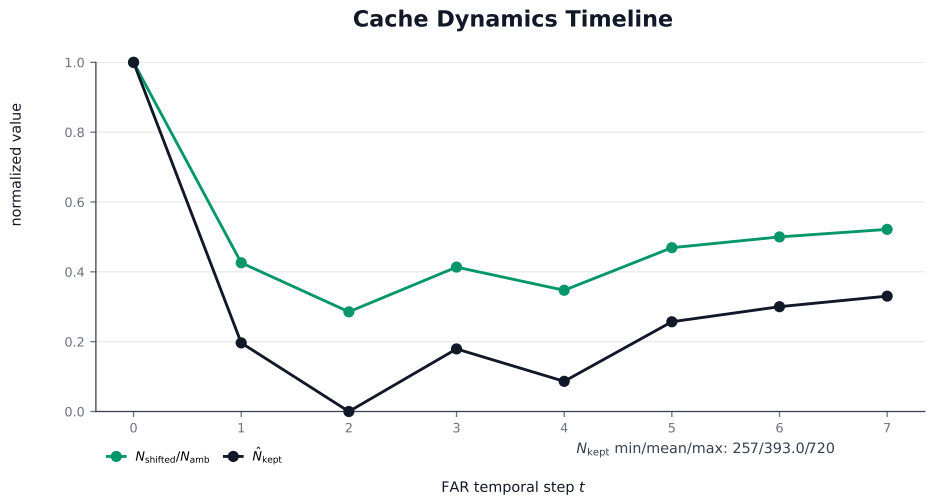


Figure 6: Cache-dynamics timeline on the MVBench `state_change_doc0107` diagnostic sample. Values are normalized for trend comparison across shifted-token and retained-token quantities.

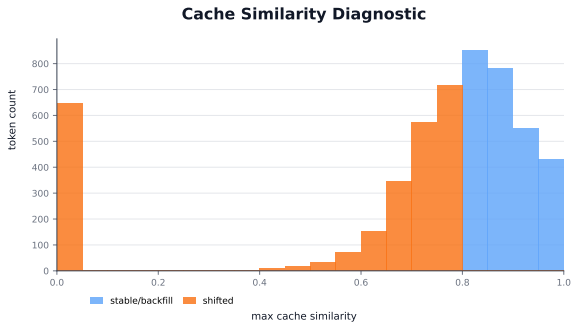


Figure 7: Ambient-cache similarity histogram for `state_change_doc0107`. Lower maximum cache similarity indicates Ambient evidence that deviates from cached context and is more likely to be promoted as shifted evidence.

Fig. 7 visualizes the cache-similarity decision more directly. The histogram separates stable tokens from shifted tokens by their maximum cosine similarity to the previous Ambient cache. Shifted tokens concentrate at lower similarity, while stable tokens occupy higher-similarity bins.

Fig. 8 shows where cache similarity is low or high over the frame token grid before the shift decision. Focus tokens bypass Ambient-cache matching and therefore have undefined similarity in this diagnostic.

Fig. 9 reports the retained-token composition at each temporal step. FAR keeps the Focus tokens first and then the Ambient tokens that deviate from the cache; stable Ambient tokens are used only as optional backfill. The bar heights therefore follow the cache dynamics in Fig. 6, unlike a single global top-k selector.

B.2 Score Components and Memory Trace

Fig. 10 displays the runtime score components exported by the FAR rerun. Attention, DCT energy, text relevance, and the final fused score are normalized within each panel, so the figure should be used

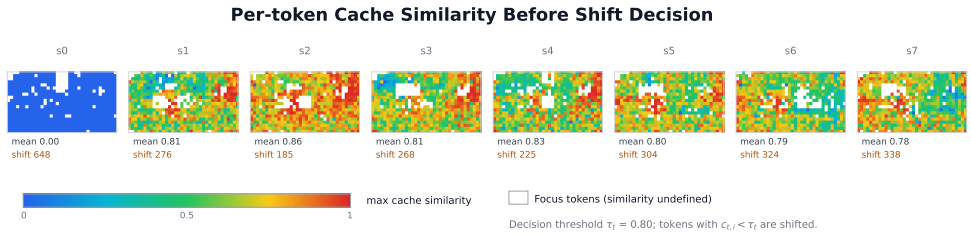


Figure 8: Ambient-cache similarity maps for `state_change_doc0107` at each temporal step, before the shift decision.

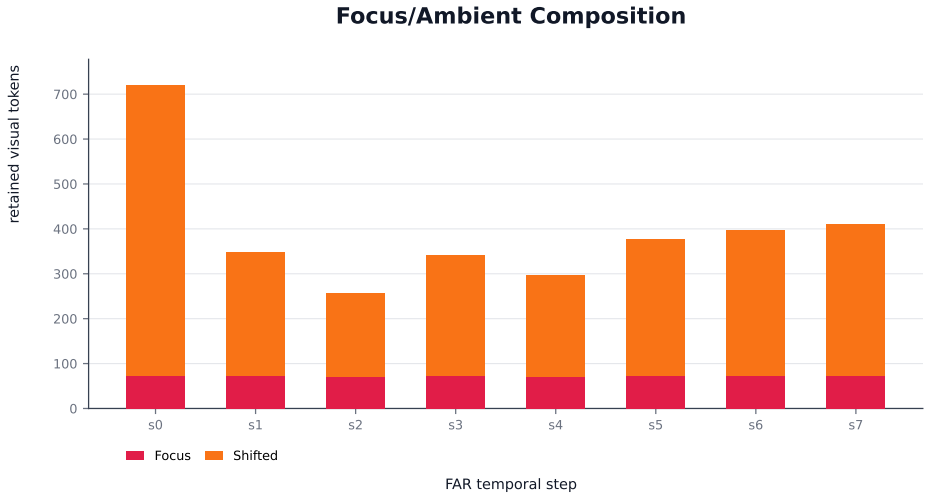


Figure 9: Per-step composition of retained tokens on the diagnostic sample. Focus tokens preserve direct task evidence, and shifted Ambient tokens carry cache-deviating context. In this diagnostic run every shifted Ambient token is kept, so the bar height reflects the number of cache deviations.

to compare spatial patterns rather than absolute magnitudes across rows. The diagnostic confirms the intended role of the scoring mixture, that is, the visual salience anchors selection, DCT highlights local variation, and text relevance acts as a bounded correction before fusion.

Fig. 11 gives the corresponding single-video phase-level memory profile. It complements the main-paper memory comparison by showing where memory is resident during vision encoding, FAR compression, LLM prefill, and decode. The curve is a diagnostic trace. The full quality–efficiency conclusions remain those reported in the main tables.

B.3 Additional Qualitative Examples

Fig. 12 provides additional long video QA examples. These cases complement the main-paper example by illustrating that FAR preserves answer-relevant local evidence while maintaining non-redundant contextual anchors under a compact visual token budget. The examples are intended as qualitative diagnostics; the aggregate conclusions are those reported in the main benchmark tables.

Figs. 13 and 14 show the token routing on two MVBench samples.

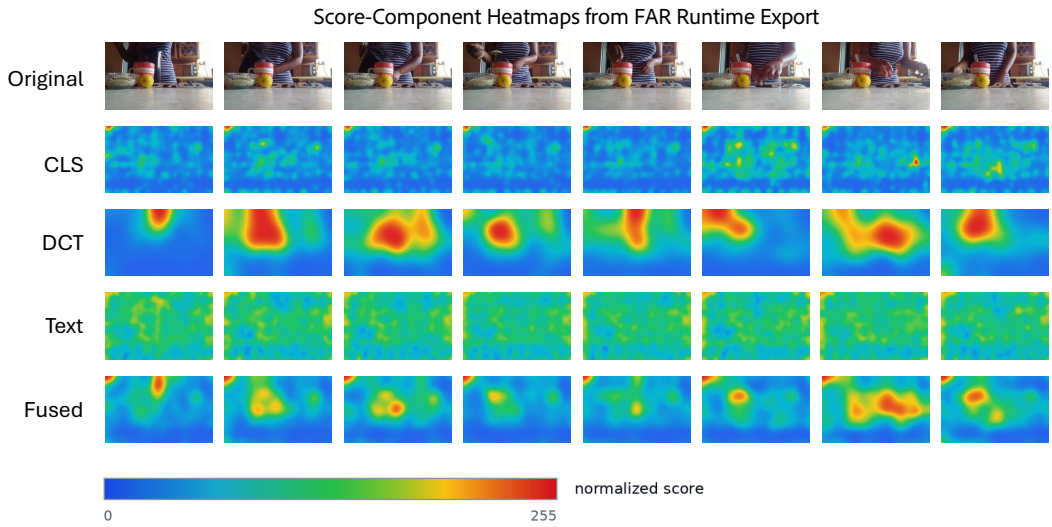


Figure 10: Score-component heatmaps for the Qwen2.5-VL-3B MVBench diagnostic sample. Rows show attention/proxy salience, DCT energy, text relevance, and the final fused score across temporal steps.

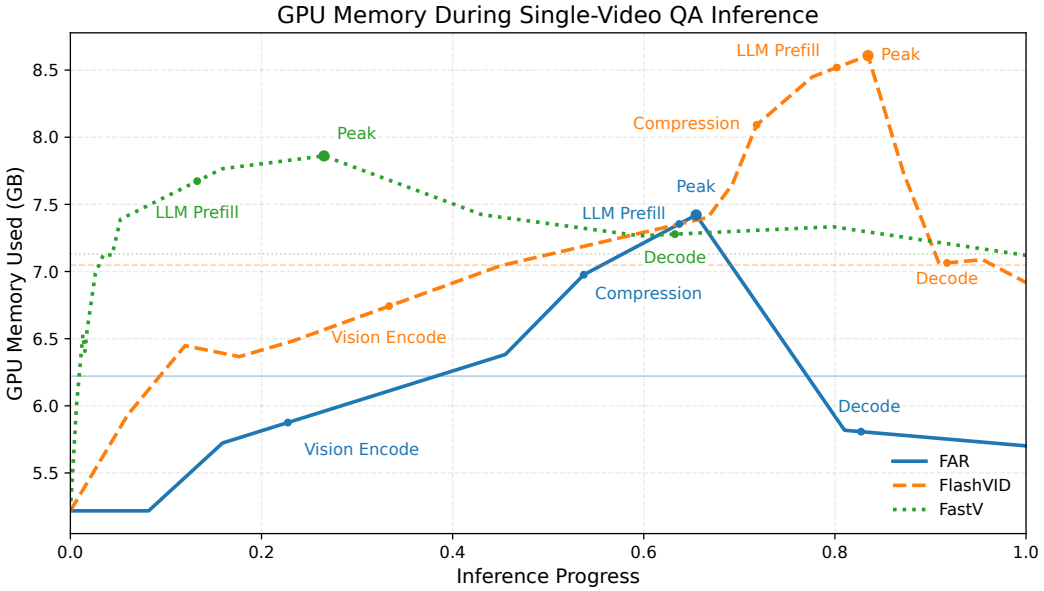


Figure 11: Single-video VRAM phase curve exported for the Qwen2.5-VL-3B MVBench diagnostic profile. The trace illustrates memory residency across inference phases rather than a benchmark average.



Figure 12: Supplementary qualitative examples for FAR. Across the examples, the retained sequence preserves task-discriminative visual evidence and changed Ambient context rather than uniformly sampling redundant frames.

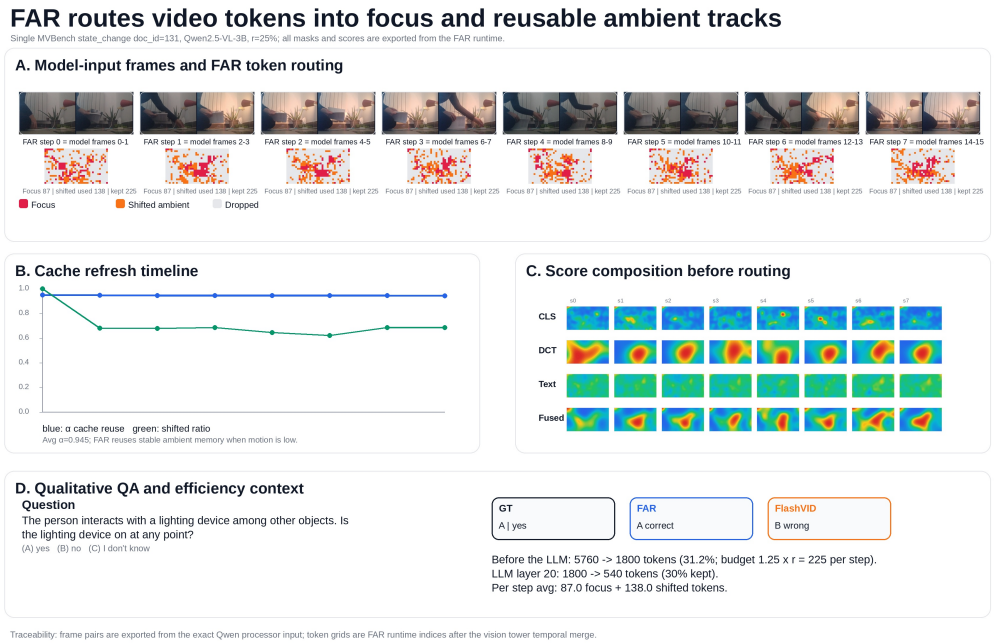


Figure 13: FAR routes video tokens into Focus and reusable Ambient tracks (MVBench `state_change`, sample 131, Qwen2.5-VL-3B).

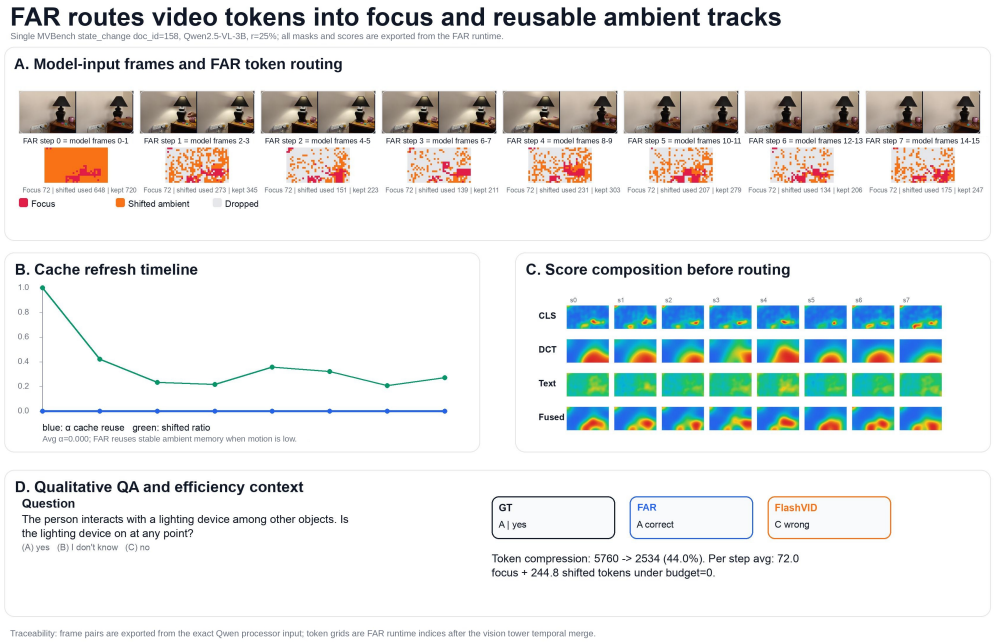


Figure 14: FAR routes video tokens into Focus and reusable Ambient tracks (MVBench `state_change`, sample 158, Qwen2.5-VL-3B).

C Additional Comparisons and Diagnostics

C.1 Pre-LLM Baselines

Table 10 reports paired bootstrap confidence intervals of FAR against each adapter on VideoMME and MLVU, using 10,000 resamples with seed 2026; VideoMME is resampled by video and MLVU is stratified by task. Among the 36 comparisons, FAR is higher with an interval above zero in 20 (DyCoKe-TTM 10, PruneVid 7, DivPrune 3) and is never lower with an interval below zero.

Table 10: Paired score difference (FAR minus adapter, in points) with 95% bootstrap confidence interval on Qwen2.5-VL-7B. *: the interval excludes zero.

Benchmark	Retention	vs. DivPrune	vs. PruneVid	vs. DyCoKe-TTM
VideoMME	1%	$+1.111\ [-0.407, +2.593]$	$+2.074\ [+0.519, +3.631]^*$	$+6.519\ [+4.815, +8.222]^*$
	5%	$+1.963\ [+0.444, +3.481]^*$	$+4.148\ [+2.556, +5.741]^*$	$+10.074\ [+8.148, +12.037]^*$
	10%	$+1.370\ [-0.037, +2.778]$	$+5.148\ [+3.556, +6.741]^*$	$+5.074\ [+3.444, +6.667]^*$
	15%	$+1.185\ [-0.111, +2.481]$	$+4.741\ [+3.185, +6.259]^*$	$+4.222\ [+2.630, +5.815]^*$
	20%	$+2.296\ [+1.037, +3.556]^*$	$+4.889\ [+3.407, +6.370]^*$	$+4.815\ [+3.222, +6.370]^*$
	25%	$+2.296\ [+1.111, +3.519]^*$	$+5.704\ [+4.296, +7.074]^*$	$+5.556\ [+3.926, +7.185]^*$
MLVU	1%	$-3.456\ [-7.230, +0.368]$	$-0.814\ [-4.435, +2.780]$	$+5.200\ [+1.281, +9.204]^*$
	5%	$+2.936\ [-0.434, +6.398]$	$+1.149\ [-2.070, +4.380]$	$+10.105\ [+6.131, +14.172]^*$
	10%	$+1.339\ [-1.807, +4.574]$	$+2.337\ [-0.885, +5.606]$	$+4.214\ [+0.701, +7.823]^*$
	15%	$+2.162\ [-0.778, +5.242]$	$+2.969\ [+0.277, +5.734]^*$	$+4.361\ [+0.866, +7.855]^*$
	20%	$-0.349\ [-3.223, +2.602]$	$+1.725\ [-1.035, +4.464]$	$+2.221\ [-1.220, +5.746]$
	25%	$+0.607\ [-1.830, +3.104]$	$+2.345\ [-0.608, +5.288]$	$+3.122\ [-0.241, +6.570]$

C.2 Additional Ablations and Sensitivity

Component memory. Table 11 lists the memory of the component variants on VideoMME at 25% retention. The variants differ by at most 0.012 GiB in average VRAM and 0.002 GiB in peak VRAM.

Table 11: Score and memory (GiB) of the component variants on Qwen2.5-VL-7B VideoMME at 25% retention.

Variant	VideoMME	Avg. VRAM	Peak VRAM
Full FAR	**60.667**	14.966	18.302
w/o Scoring	60.148	14.954	18.300
w/o Cache	60.407	14.966	18.302
w/o Assembly	60.185	14.966	18.302

Random controls. Random samples visual tokens under the matched budget, and Random-Ambient keeps score-based Focus selection but randomizes the Ambient decision. On VideoMME, FAR exceeds Random by 0.408, 1.222, and 1.482 points at 1%, 5%, and 25% retention, and exceeds Random-Ambient by 1.074, 0.222, and 0.667 points. At 1%, Random-Ambient is 0.667 points below Random, and the paired 95% bootstrap intervals of the three pairwise differences, computed as for Table 10, all include zero. Each control in Table 12 is a single run with sampling seed 0. The FAR entry at 5% comes from the same batch as the controls, and the entries at 1% and 25% are those of Table 1.

Table 12: Random controls on Qwen2.5-VL-7B VideoMME. Measured retention is based on visual tokens.

Target	Measured	Random	Random-Ambient	FAR
1%	1.0077%	49.185	48.519	**49.593**
5%	5.0065%	53.704	54.704	**54.926**
25%	25.0101%	59.185	60.000	**60.667**

Cache sensitivity. All sweeps are on VideoMME at 25% retention. Table 13 shifts the base threshold τ_0, together with its clipping window, on Qwen2.5-VL-7B; the score changes by -0.407 to $+0.037$

points relative to the default, a span of 0.444 points. Table 14 sets the base refresh coefficient α_0 to 0.60 or 0.80, which lowers the score by 0.185 or 0.111 points. Table 15 shifts τ_0 on LLaVA-OneVision-7B, which encodes each frame separately; the three settings span 0.370 points.

Table 13: Cache-threshold sweep on Qwen2.5-VL-7B VideoMME at 25% retention. The default value is $\tau_0 = 0.80$.

τ_0	0.70	0.75	0.80	0.85	0.90
VideoMME	60.704	60.667	60.667	60.259	60.296

Table 14: Cache-refresh sweep on Qwen2.5-VL-7B VideoMME at 25% retention. The default value is $\alpha_0 = 0.75$.

α_0	0.60	0.75	0.80
VideoMME	60.481	60.667	60.556

Table 15: Cache-threshold sweep on LLaVA-OneVision-7B VideoMME at 25% retention. The default value is $\tau_0 = 0.80$.

τ_0	0.75	0.80	0.85
VideoMME	56.667	56.815	57.037

C.3 Low-Retention Categories

Table 16 decomposes LongVideoBench at 1% retention by question category, with category codes following LongVideoBench [48]. Differences are ablation minus Full FAR, so a positive value means that the removed component costs accuracy in that category. Both components help overall, while individual categories are harmed: the largest costs are Assembly on SOS ($+7.41$) and Cache on SAA ($+4.17$) and T3O ($+4.05$).

Table 16: LongVideoBench at 1% retention on Qwen2.5-VL-7B by question category. Ablation columns are score differences relative to Full FAR, in points.

Category	n	Full FAR	Δ w/o Cache	Δ w/o Assembly
E3E	94	59.57	-5.32	-3.19
O3O	66	48.48	-1.52	-6.06
SSS	97	24.74	-3.09	-4.12
S2O	72	48.61	-4.17	-2.78
E2O	65	46.15	-1.54	-4.62
S2E	93	60.22	0.00	-2.15
T2A	79	43.04	-1.27	0.00
S2A	88	54.55	0.00	-1.14
T2E	65	63.08	0.00	0.00
TAA	82	41.46	$+1.22$	-2.44
T2O	76	36.84	-2.63	$+2.63$
O2E	87	55.17	$+1.15$	$+1.15$
T3E	73	38.36	$+2.74$	$+1.37$
TOS	73	31.51	$+2.74$	$+2.74$
T3O	74	43.24	$+4.05$	0.00
SAA	72	44.44	$+4.17$	0.00
SOS	81	53.09	0.00	$+7.41$
All	1337	46.67	-0.30	-0.67

C.4 Evidence Routing Diagnostics

Multi-video cache statistics. Fig. 15 extends the single-sample histogram in Fig. 7 to 300 VideoMME videos (900 questions, 13,500 temporal groups with a non-empty cache). In total, 6,988,625 Ambient

tokens fall below the matching threshold and are classified as shifted, and 789,805 are classified as stable, i.e., a shifted fraction of 89.85%. This is a feature-threshold statistic rather than a semantic change-detection accuracy.

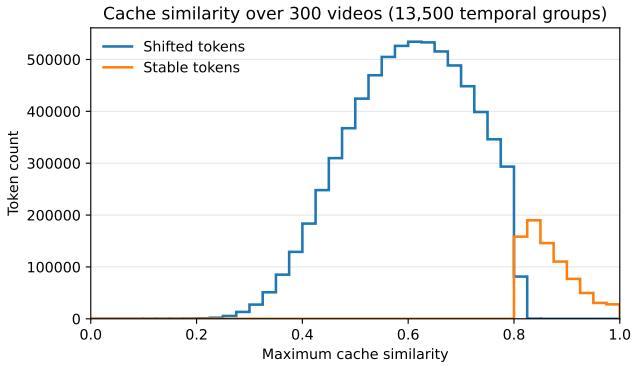


Figure 15: Maximum cache similarity of Ambient tokens over 300 VideoMME videos on Qwen2.5-VL-7B at 25% retention.

Cross-position matching. Table 17 reports label-free feature statistics over 735 adjacent-frame pairs from 49 videos. The nearest off-diagonal match has a higher mean cosine than the fixed-grid match (0.697 against 0.619), so cache matching is not locked to identical grid positions. These statistics do not establish object correspondence.

Table 17: Feature-matching diagnostics on Qwen2.5-VL-7B. Displacement is normalized by the grid scale.

Statistic	Observations	Mean	Std.
Fixed-grid cosine	188,160	0.6194	0.2703
Off-diagonal nearest cosine	188,160	0.6972	0.1205
Off-diagonal displacement	188,160	0.2735	0.1945
Live-cache maximum cosine	175,290	0.7127	0.1495

Temporal granularity. FAR updates the cache at the temporal-group granularity emitted by the encoder, i.e., one step per pair of sampled frames on Qwen2.5-VL. Table 18 and Fig. 16 vary the sampling density on VideoMME at 25% retention (150 questions over 50 videos per setting). With denser sampling, the stable fraction rises from 10.75% to 22.92%, while the refresh coefficient α_t moves only from 0.9286 to 0.9367.

Table 18: Sampling-density diagnostic on Qwen2.5-VL-7B VideoMME at 25% retention.

Metric	32 frames	64 frames	128 frames
Temporal groups	16	32	64
Stable fraction	10.75%	16.22%	22.92%
Mean maximum cosine	0.6313	0.6656	0.6990
Motion proxy g_t	0.1072	0.0904	0.0665
Refresh α_t	0.9286	0.9319	0.9367
Peak VRAM (GiB)	18.023	20.057	24.647

Retention overlays. Fig. 17 overlays the retained tokens of five VideoMME questions on Qwen2.5-VL-7B at 25% retention. The overlays project the token grid onto the frames and are not pixel-level localization. The examples are deliberately chosen failure cases.

Static-scene questions. Fig. 18 shows four VideoMME questions answered from persistent scene content: items hanging on the walls, the Earth below an astronaut, on-screen car features, and a classroom activity. FAR answers all four correctly. The Focus score includes salience and DCT energy, which do not depend on motion, so structured static content can enter Focus. At the first

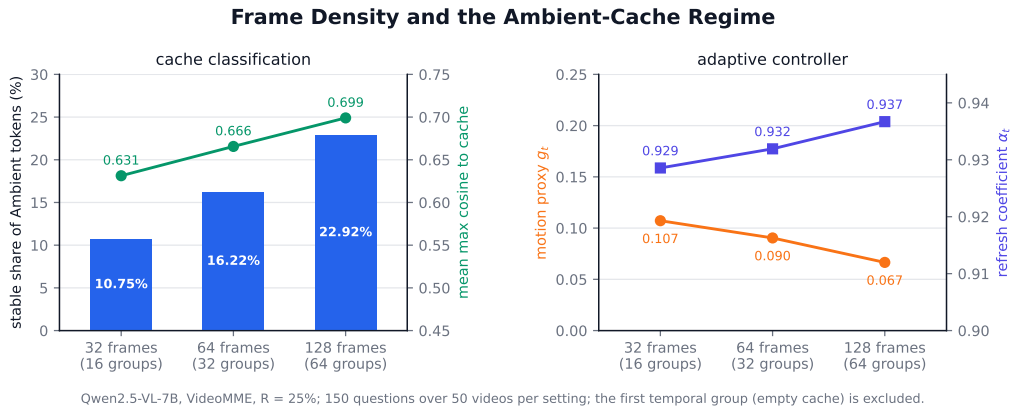


Figure 16: Frame-density diagnostic on Qwen2.5-VL-7B VideoMME at 25% retention. Left: stable share of Ambient tokens and mean maximum cosine to the cache. Right: motion proxy g_t and refresh coefficient α_t.

temporal step the cache is empty, so every Ambient token is treated as shifted. The retained tokens cover the wall items, the Earth region, the on-screen text, and the levitated objects.

C.5 Memory Attribution and Runtime

Materialization control. Table 19 compares attention materializations at the same 64-frame input on Qwen2.5-VL-7B. The frame-wise extraction of Sec. 3.2 peaks at 17.178 GiB in streaming form and 21.847 GiB in batched form, while naive video-wide attention runs out of memory.

Table 19: Matched-shape materialization control on an H100 with 94 GB memory.

Materialization	Peak VRAM (GiB)
Frame-wise streaming	17.178
Frame-wise batched	21.847
Naive video-wide attention	OOM

Phase memory. Table 20 decomposes memory by inference phase on MLVU (50 samples per cell, exclusive H100). Prefill and decode memory grow with retention, but vision encoding sets the peak, which stays nearly constant.

Table 20: Phase memory (GiB) of FAR on Qwen2.5-VL-7B MLVU. Phase values are means of per-sample phase peaks; Overall max is the maximum over the 50 samples.

Target / measured retention	Vision	Compression	Prefill	Decode	Overall max
1% / 1.119%	16.710	14.457	14.411	14.392	18.046
10% / 10.131%	16.709	14.460	14.529	14.431	18.047
25% / 25.152%	16.709	14.466	14.727	14.496	18.047

Latency scaling. Table 21 reports the compression time and the time from visual encoding start to the first generated token (TTFT) of FAR as the number of input frames grows. The sweep uses an exclusive H100, CUDA-event timing, MLVU at 25% retention, and ten samples per cell; both quantities grow with the input length. In a resolution sweep with 864 to 1,584 visual tokens, compression takes 100.5–118.2 ms and TTFT 321.1–356.9 ms.

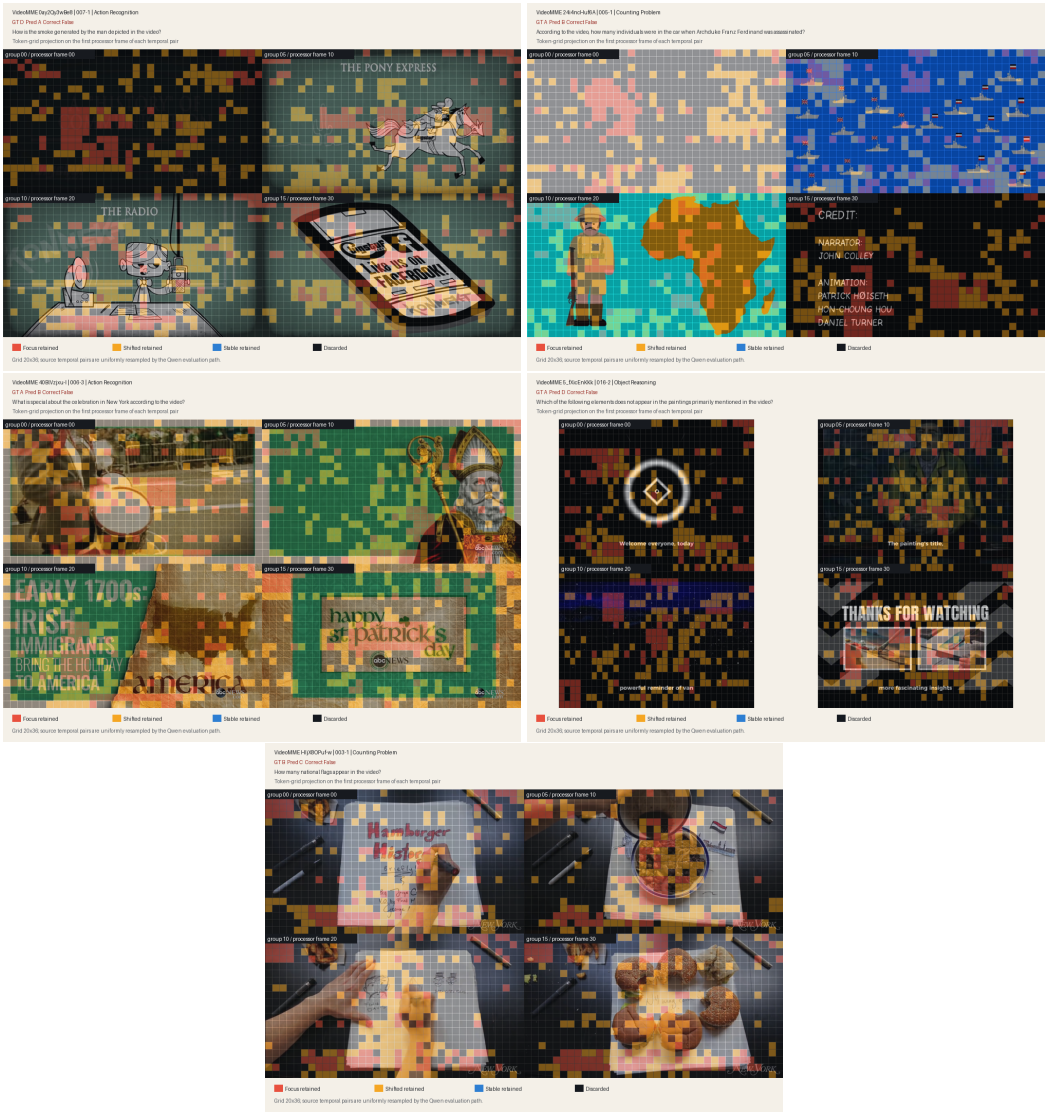


Figure 17: Token-retention overlays on five VideoMME questions (Qwen2.5-VL-7B, 25% retention). Each panel shows four temporal groups with Focus, shifted, stable, and discarded tokens.

Table 21: Frame-count scaling of FAR on Qwen2.5-VL-7B. Compression time and TTFT are in milliseconds.

Frames	Visual tokens	Compression	TTFT	Peak VRAM (GiB)
32	4,256	191.7	555.4	16.333
64	8,512	365.7	925.0	17.178
128	17,024	952.4	1,922.6	18.876
256	34,048	1,407.6	3,158.4	22.273



Figure 18: Token-retention overlays on four static-scene VideoMME questions (Qwen2.5-VL-7B, 25% retention). Each panel shows four temporal groups with Focus, shifted, and discarded tokens.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction covers the technical contribution, the developed method and the experimental contribution.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Sec. 4.7.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes, the baseline model, technical details, hyper-parameter and configuration are detailed in the submission for reproducibility.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The datasets this paper uses are publicly available, and the source code is available at `https://gitlab.com/stephenzeng/far`.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The implementation details are given at the beginning of the experimental section and in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Paired bootstrap confidence intervals against the pre-LLM baselines are reported in Appendix C.1 (10,000 resamples, seed 2026; VideoMME is resampled by video and MLVU is stratified by task).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The hardware, especially the GPU requirement, is listed in the main text and the supplementary materials. All the experiments are conducted on a H100 GPU with 94GB memory.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This paper focuses on a fundamental task of deep representation learning and conducts experiments on publicly available datasets.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The societal impact of this work has been discussed at the end of the conclusion section. More efficient long video inference can make multimodal analysis more accessible, but may amplify risks in privacy-sensitive video understanding. Therefore, practical deployment should include privacy safeguards, and clear limits on high-stakes use..

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This work focuses on a fundamental problem in deep representation learning and conducts experiments on standard datasets. It does not involve any new pre-trained language models, image generators, or scraped datasets. We do not envision such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes, all the assets have been properly cited, with a license to use for academia and no commercial purpose.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: In this paper, LLM is used only for writing, editing, or formatting purposes. The core method development in this research of this paper does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.