EVENT-ADAPTIVE TEMPORAL PATCHING FOR TOKEN-BUDGETED VIDEO GROUNDING AND CHAPTERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Video multimodal large language models (MLLMs) typically compress a video into a fixed number of visual tokens before temporal reasoning, using uniform sampling and fixed temporal patches. This can underrepresent brief actions and transitions, whereas increasing the sampling rate usually increases downstream processing cost. We introduce Event-Adaptive Temporal Patching (EATP), a query-independent, training-free front end that decouples observation density from the number of encoded visual units. EATP observes a dense candidate stream and uses exact dynamic programming over coarse visual descriptors to partition it into contiguous, variable-length temporal groups under a fixed visual-token budget. The resulting schedule allocates shorter groups to regions with large descriptor changes and longer groups to relatively stable content. A length-aware temporal patch embedding, derived from the pretrained two-frame kernel, encodes each group without fine-tuning. Experiments on QVHighlights moment retrieval and VidChapters chapter generation show training-free improvements over native uniform sampling, supporting the benefit of combining dense observation with content-adaptive temporal grouping rather than dense sampling alone. On QVHighlights, EATP improves mIoU by 2.16 points, strict R@1@0.7 by 8.04 points and R@1@0.5 by 3.14 points. On VidChapters, it improves F1, tIoU, SODA, CIDEr, and GRACE by 4.5, 1.4, 1.7, 6.5, and 1.0 points, respectively.

1 INTRODUCTION

Video multimodal large language models (MLLMs) increasingly target time-sensitive reasoning over long videos (Ren et al., 2024; Huang et al., 2024), but reason over a representation that has already discarded most of the source stream. Before a language model can answer a question, retrieve a moment, or predict a chapter boundary, the video front end must decide which frames to observe and how to convert them into visual tokens. By *front end* we mean the pipeline that turns a raw video into visual tokens—decoding, sampling, pixel preprocessing, and the vision encoder including its first Conv3D layer—as opposed to the language backbone that consumes those tokens. This decision becomes especially consequential for long videos, where information is distributed unevenly in time: minutes of nearly static content may be interrupted by a brief action, slide change, or topic transition. Boundary-sensitive benchmarks such as QVHighlights and VidChapters make this asymmetry explicit because success depends not only on recognizing what appears, but also on localizing when it begins and ends (Lei et al., 2021; Yang et al., 2023a).

The default response to long duration is temporal subsampling. Uniform sampling is simple and reproducible, but it allocates equal observation capacity to intervals with unequal information density. Existing work either selects a small set of frames (Wu et al., 2019; Yu et al., 2025; Tang et al., 2025; Li et al., 2026a) or compresses video representations at later stages (Xu et al., 2024; Shen et al., 2025; Ren et al., 2023; Song et al., 2024). These routes show that temporal allocation matters, yet neither specifies how a dense, temporally ordered observation stream should be formed into the contiguous patches consumed by a pretrained video front end.

This distinction matters because temporal patchification is not merely a reshape. In the Qwen3.5-9B instantiation (Qwen Team, 2026), neighboring frames are combined by the first Conv3D projection before the main vision transformer. With a fixed temporal patch size, a tubelet can straddle an action onset or semantic transition; its first embedding then mixes evidence from both sides. Increasing the

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



Figure 1: Overview of Event-Adaptive Temporal Patching (EATP). The native front end encodes consecutive two-frame temporal pairs. EATP first observes a denser candidate stream, computes coarse content descriptors, and partitions the stream into exactly the native number of contiguous, variable-length groups with fixed-budget dynamic programming. Each group is packed into an K_{max} -slot tensor and projected by a length-aware temporal Conv3D, preserving the downstream temporal-unit count.

input frame rate alone does not solve the problem: under the native patchification rule it increases the number of temporal visual units and expands the vision sequence presented downstream. Conversely, compressing only after patch embedding cannot restore an event that was never observed and does not determine whether the first temporal projection crossed a candidate boundary. The open design problem is therefore to *observe densely but encode sparsely*: expose the front end to more frames, place temporal units according to content, and keep the number of downstream visual tokens bounded.

We introduce **Event-Adaptive Temporal Patching (EATP)**, a query-independent, training-free front end for this setting. EATP observes a denser candidate stream, uses a fixed-budget dynamic program to form the native number of contiguous temporal groups, and projects each variable-length group into one temporal unit per spatial patch. It consequently reallocates temporal evidence without increasing the downstream visual-token count; the grouping objective and the pretrained-compatible projection are specified in Section 3.

Figure 1 gives an overview of the resulting front end.

We evaluate EATP on QVHighlights moment retrieval (Lei et al., 2021) and VidChapters chapter generation and boundary localization (Yang et al., 2023a). The controlled comparison uses the same training-free Qwen3.5-9B checkpoint and native temporal-unit budget for EATP and the uniform baseline; budget and grouping controls test the scope of this operating point.

Our contributions are:

- We formulate dense observation and pre-encoder temporal patch construction as a coupled bottleneck for boundary-sensitive Video-MLLMs.
- We introduce a query-independent, fixed-budget dynamic-programming front end that groups dense candidate frames into content-dependent temporal patches and projects them with a length-aware Conv3D interface.

- We evaluate the controlled native-budget comparison on QVHighlights and VidChapters, together with temporal-budget and grouping controls that delimit the selected operating point.

2 Background and Related Work

The design of EATP follows from two observations about the native uniform-sampling pipeline, measured on a long-video chaptering benchmark and a short-video moment-retrieval benchmark. We run a native Qwen3.5-9B configuration (2 FPS, at most 768 frames) and use the same evaluation protocol as the main experiments. The two analyses isolate complementary dimensions of a single bottleneck: how a fixed frame budget allocates observations over time.

Chaptering: observations near boundaries are more useful On VidChapters, the 768-frame cap makes long videos more sparsely observed, but video duration is also confounded with content difficulty. We therefore compare ground-truth chapter boundaries within matched actual sampling-interval bins. A boundary is a sample "hit" when its nearest sampled frame lies within 0.5 seconds, and a "miss" otherwise. Across every bin with a valid comparison, hit boundaries have higher Boundary@5s recall than missed boundaries; the within-bin gaps range from 1.03 to 2.96 points (Table 5). This observational association supports the premise that observations placed near chapter boundaries are more useful than equally sparse observations placed elsewhere.

Moment retrieval: more uniform samples are not uniformly better QVHighlights isolates sampling density because its clips do not reach the tested frame caps. Increasing uniform sampling from 1 to 2 FPS improves retrieval, but further increasing it to 4 FPS does not improve mIoU despite the higher visual-token cost (Figure 2; full values are in Table 7). Uniformly increasing observation density can therefore add cost without consistently improving boundary-sensitive localization.

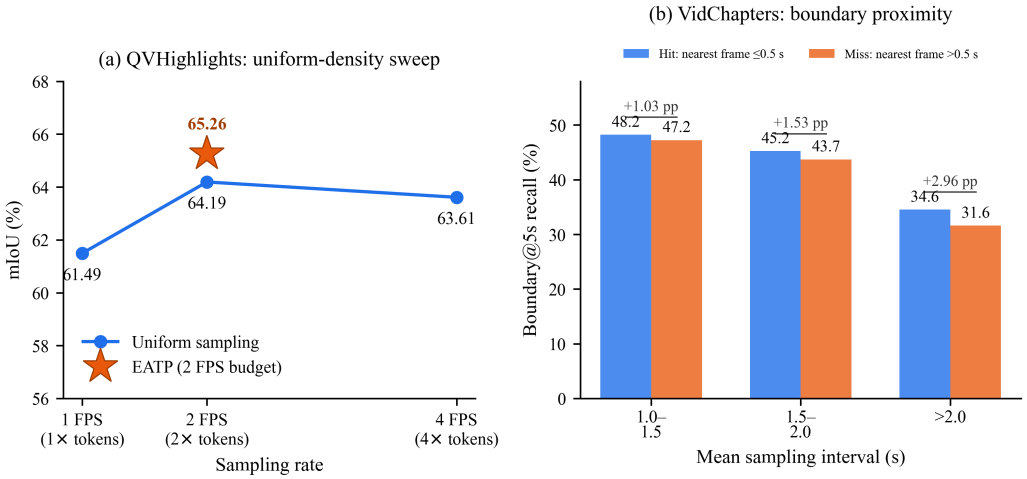

Figure 2: Motivating evidence for content-adaptive temporal allocation. Left: QVHighlights mIoU under uniform sampling, with EATP shown as the star. Right: Boundary@5s recall for boundary-proximal and boundary-missed samples within matched sampling-interval bins on VidChapters. Brackets report within-bin differences; the right-panel comparison is observational.

2.1 Two Regimes, One Bottleneck

Together, the two analyses show that a fixed-budget front end must determine both how densely to observe a video and where to allocate its encoded temporal units. EATP addresses this bottleneck by

observing a denser candidate stream while preserving the native temporal-unit budget; its grouping objective and pretrained-compatible projection are described in Section 3.

2.2 Related Work

Adaptive frame selection. Prior work reduces video-processing cost by selecting a small subset of frames. AdaFrame learns a sequential selection policy for efficient video recognition (Wu et al., 2019); Frame-Voyager learns query-conditioned frame combinations for Video-LLMs (Yu et al., 2025). AKS balances prompt relevance and temporal coverage under a fixed keyframe budget (Tang et al., 2025), while MaxInfo selects a diverse training-free subset in embedding space (Li et al., 2026a). These methods select individual frames or frame sets before encoding. EATP instead retains its decoded candidate stream and partitions it into contiguous, content-dependent groups that are consumed by the first temporal projection.

Token-budgeted video compression. Other approaches compress video after or during visual feature formation. SlowFast-LLaVA combines low-rate high-resolution and high-rate pooled streams (Xu et al., 2024); LongVU performs frame and token reduction (Shen et al., 2025); and TESTA learns temporal–spatial token aggregation inside the encoder (Ren et al., 2023). EATP operates earlier, at the input interface, by deciding which dense observations form each temporal patch and adapting the pretrained two-frame projection to variable-length groups without fine-tuning.

Temporal grounding and chaptering. QVHighlights supports query-based moment grounding and highlight detection (Lei et al., 2021), while VidChapters-7M provides timestamped chapter annotations for chapter generation and boundary localization (Yang et al., 2023a). Both tasks directly evaluate temporal boundaries, providing complementary testbeds for EATP.

3 Method

EATP is applicable to pretrained video front ends with temporal patchification. In the following, we consider the common case where the first vision layer is a two-frame temporal Conv3D.

3.1 Problem formulation and budget contract

Consider a video of duration T processed by the native front end. The native sampler selects

$$n \;=\; \min\big(\lceil T \cdot f_n \rceil,\; N_{\max}\big) \tag{1}$$

frames at rate f_n with a maximum frame budget $N_{\max}$, where both are model-specific hyperparameters. It pairs consecutive frames into temporal patches of size two and projects each pair with the first Conv3D layer, so the front end produces

$$G \;=\; \lceil n/2 \rceil \tag{2}$$

temporal units. Each unit emits one embedding per spatial patch; after the standard patch merger, the downstream visual-token count is proportional to G.

EATP keeps the temporal-unit budget G exactly and changes what each unit contains. The output therefore has the same temporal-unit count as the native pipeline, while each unit can contain a variable number of real frames. This preserves the downstream visual-token ceiling but does not make denser candidate decoding, descriptor computation, or wider first-layer inputs free.

3.2 Dense candidate observation

EATP first decodes a candidate stream $x_1, \ldots, x_C$ at rate $f_c = \rho f_n$, where $\rho > 1$ is the candidate-density multiplier, each candidate carrying its true timestamp t_i with $t_1 < t_2 < \cdots < t_C$. The candidate count is capped by twice the native frame budget and by the group capacity,

$$G \cdot K_{\min} \le C \le \min\left(\rho n, G \cdot K_{\max}\right), \tag{3}$$

where $K_{\min}$ and $K_{\max}$ are grouping hyperparameters; the lower bound guarantees that the partition below is feasible for every video. Candidates are resized with the native smart-resize rule computed at the *native* frame budget equation 1, so observing twice as many candidates does not reduce per-frame resolution.

3.3 FIXED-BUDGET CONTENT GROUPING

Each candidate x_i is summarized by a compact content descriptor $d_i \in \mathbf{R}^D$. The descriptor passes the resized frame through a small separable blur pool (a six-tap triangular kernel with stride four), averages the result over a spatial grid, and scales each channel by the pretrained normalization constants, yielding D dimensions. The blur pool is intended to suppress codec noise and minor motion, while the grid retains coarse spatial layout. The resulting distance is therefore used as a coarse visual-change proxy rather than a semantic-event detector.

Grouping partitions the ordered candidates into exactly G contiguous groups $g_1, \ldots, g_G$ with lengths $K_j = |g_j| \in [K_{\min}, K_{\max}]$, minimizing the within-group centroid reconstruction error

$$Q(\pi) \;=\; \sum_{j=1}^{G} \sum_{i \in g_j} \|d_i - \mu(g_j)\|_2^2, \qquad \mu(g_j) \;=\; \frac{1}{K_j} \sum_{i \in g_j} d_i. \tag{4}$$

Because the groups are contiguous and the objective decomposes additively, the exact optimum is computed by dynamic programming over the candidate index. With $D(j, i)$ denoting the minimal cost of partitioning the first i candidates into j groups,

$$D(j, i) \;=\; \min_{k \in [K_{\min}, K_{\max}]} D(j-1,\, i-k) \;+\; \mathrm{cost}(i-k+1,\, i), \tag{5}$$

where $\mathrm{cost}(a, b)$ is the within-group SSE of equation 4 on candidates $a \ldots b$. Prefix sums of the descriptors and their squared norms evaluate every $\mathrm{cost}(a, b)$ in $O(D)$, giving $O(G\,C\,K_{\max}\,D)$ time in total; D is fixed in our implementation.

The partition covers every candidate exactly once: no frame is discarded, and grouping re-allocates the native budget rather than subsampling it. Static stretches can admit long groups at low descriptor cost, whereas large descriptor changes can favor boundaries. The resulting schedule is content-adaptive in descriptor space under the fixed budget G.

3.4 ADAPTIVE PROCESSOR CONTRACT

For each group the Processor applies the native resize and normalization path to its K_j real frames, writes them into a $K_{\max}$-slot temporal tensor, zero-pads only the inactive slots, and records the true length K_j. It then flattens each spatial patch across its $K_{\max}$ temporal slots, exactly as the native processor does for its two-slot patches, producing the standard processor outputs: the flattened tubelets `pixel_values_videos` and the grid metadata `video_grid_thw` whose temporal dimension is G. Padding is performed only to fill the fixed-width compute path; the model always receives the true group length and never treats padding as content.

3.5 LENGTH-AWARE GROUPED TEMPORAL PROJECTION

Typical Video-MLLMs with temporal patch embeddings employ a 3D convolution with a temporal kernel size and stride of 2 as the first layer of the vision encoder. We denote its two temporal slices by W_0, W_1. EATP expands the temporal kernel to size $K_{\max}$ by repeating the native pair,

$$\widehat{W}_r = W_{r \bmod 2}, \qquad r = 0, \ldots, K_{\max} - 1., \tag{6}$$

and feeds each $K_{\max}$-slot group through it as one temporal patch with stride $K_{\max}$, emitting one embedding per spatial patch per group. To keep the projection consistent across group lengths, the model prepares each group as follows. If K_j is odd, the last real frame is repeated once inside the model, giving an even length

$$L_j \;=\; K_j + (K_j \bmod 2). \tag{7}$$

The valid prefix of length L_j is scaled by $2/L_j$ and the remaining slots stay zero before the projection.

The scaling is chosen so that, at initialization, the adaptive projection is exactly the mean of the native two-frame responses over the prepared group. Write the native temporal projection per spatial patch as $\Phi_2(a, b) = W_0\, a + W_1\, b$, and let $\tilde{x}$ denote the prepared group (tail-duplicated when K_j is odd). Since the expanded kernel alternates W_0, W_1, the scaled projection reduces to a pairwise pass of the original convolution,

$$\Phi_{K_{max}}\big(\tfrac{2}{L_j}\,\tilde{x}\big) \;=\; \frac{2}{L_j} \sum_{p=1}^{L_j/2} \Phi_2\big(\tilde{x}_{2p-1},\, \tilde{x}_{2p}\big). \tag{8}$$

In words, every prepared group runs through the original two-frame convolution pair by pair, and the responses are averaged: no new parameters are introduced, and a group of any length reuses exactly the pretrained temporal slices. The special cases follow directly:

- a two-frame group is identical to the native Conv3D response ($L_j = 2$, scale $2/L_j = 1$);
- a singleton reduces to the native duplicated-frame response $(W_0 + W_1)\, x$;
- an even-length group equals the average of its $K_j/2$ consecutive native pair responses;
- an odd-length group equals the average of $(K_j + 1)/2$ native pair responses, with the tail pair duplicated.

Consequently EATP is training-free: the front end runs on the pretrained checkpoint with the expanded kernel, requires no gradient updates or additional data, and any window with $K_j = 2$ reproduces the pretrained response exactly. Fine-tuning the expanded kernel afterwards remains an optional extension, not a prerequisite.

4 Experimental Setup

4.1 Datasets and tasks

We evaluate on two boundary-sensitive video understanding benchmarks. **QVHighlights** uses video-only input and pairs natural-language queries with moment boundaries and saliency scores; we evaluate moment retrieval on its test split with mean IoU (mIoU) and top-1 recall at IoU thresholds 0.5 and 0.7 (Lei et al., 2021). Its videos are roughly 150 seconds long, which keeps the frame caps inactive so that sampling rate is the only variable in the scaling experiments. **VidChapters** provides timestamped chapter annotations; we evaluate chapter generation and boundary localization with GRACE (the BERTScore-based ARC-Chapter metric, using roberta-large) (Wang et al., 2025b; Zhang et al., 2020), tIOU (Krishna et al., 2017)matched segment F1, matched caption-token F1, CIDEr (Vedantam et al., 2015), and SODA_c (Fujita et al., 2020; Yang et al., 2023a). Every VidChapters row uses video frames together with the same timestamped ASR transcript, and is evaluated on the test split.

4.2 Headline comparison and ablation settings

The headline comparison evaluates Native uniform and EATP at the 2 FPS / 768-frame operating point. Both use the same training-free Qwen3.5-9B checkpoint, prompts, per-frame resolution, and decoding rule. QVHighlights uses video-only input, whereas VidChapters additionally uses the same timestamped ASR transcript. For a video of duration T, the native front end samples $n = \min(\lceil 2T \rceil, 768)$ frames and produces $G = \lceil n/2 \rceil$ temporal units by applying two-frame temporal patches. EATP instead observes a 4 FPS candidate stream with true timestamps, then uses the fixed-budget grouping and length-aware projection of Section 3 to map these candidates back to the same G temporal units. Thus, the headline comparison preserves the downstream temporal-unit budget while changing the allocation of temporal units.

We use a fixed-group control to test whether the gain comes from content-adaptive grouping rather than merely observing denser candidates. This control uses the same 4 FPS candidate stream and the same output budget as EATP, but divides candidates into content-independent contiguous groups of four (Uniform K=4). Relative to this matched control, EATP changes only the placement of group boundaries. Uniform K=8 and K=16 further characterize the sensitivity of content-independent aggregation to coarser temporal grouping; they are not equal-budget tests of the DP objective.

Additionally, the temporal-budget sweep tests whether EATP can be explained simply by increasing the uniform sampling budget. We evaluate both methods at 1 FPS / 256, 2 FPS / 768, and 4 FPS / 1,536. In particular, the 4 FPS Native uniform setting provides a high-budget uniform-sampling reference: it passes more uniformly sampled frames and a larger downstream temporal-unit budget to the model than the 2 FPS / 768 EATP setting. Together with the matched Uniform $K = 4$ control, this sweep helps distinguish the benefit of dense observation followed by adaptive grouping from a gain caused by uniformly sampling more frames alone.

Table 1: VidChapters results using the published $100\times$ metric convention. Published results from ARC-Chapter (Wang et al., 2025b) are included as reference only and are not directly comparable across heterogeneous settings. The local Qwen3.5-9B rows use video+ASR on the full test split without fine-tuning; boldface and Δ compare only these rows.

Method	Setting	FT	F1	tIoU	SODA	CIDEr	GRACE
Published VidChapters reference							
GPT-4o-mini	proprietary MLLM	no	31.2	63.6	6.8	37.8	–
GPT-4o	proprietary MLLM	no	37.6	68.0	8.1	51.0	–
Gemini-2.0-Flash	proprietary MLLM	no	40.2	69.3	11.4	69.7	–
Gemini-1.5-Pro	proprietary MLLM	no	42.2	70.9	11.4	63.2	–
Llama 3.1-8B	zero-shot baseline	no	29.5	62.5	6.2	30.7	–
Chapter-Llama (Ventura et al., 2025)	trained chaptering model	yes	45.3	71.8	19.3	100.9	–
ARCChapter-asr	ARC-Chapter (SFT)	yes	54.5	76.7	25.3	144.0	28.9
ARCChapter-vid	ARC-Chapter (SFT)	yes	50.2	74.3	22.9	138.3	25.4
ARCChapter-vidasr	ARC-Chapter (SFT)	yes	59.3	79.6	30.6	186.6	34.3
ARCChapter-vidasr +RL	ARC-Chapter (GRPO)	yes	60.8	80.7	31.0	190.7	34.6
Controlled Qwen3.5-9B comparison (training-free)							
Qwen3.5-9B (baseline)	native uniform	no	61.6	74.2	29.1	173.2	60.1
EATP (ours)	adaptive front end	no	**66.1**	**75.6**	**30.8**	**179.7**	**61.1**
Δ vs. baseline	EATP $-$ Qwen3.5-9B baseline	–	+4.5	+1.4	+1.7	+6.5	+1.0

4.3 Implementation and evaluation protocol

We run the training-free setting throughout the main results: the pretrained Qwen3.5-9B checkpoint is used without any gradient updates, and the expanded kernel of EATP is constructed from the pretrained two-frame slices by repetition, so no method introduces new parameters. We use $f_n = 2$ FPS, $N_{\max} = 768$, $\rho = 2$, $K_{\min} = 2$, and an 8×8 descriptor grid; $K_{\max}$ is selected per task. Candidates are decoded at true timestamps and resized with the native smart-resize rule at the native frame budget, keeping per-frame resolution identical across front ends. Inference runs in batch mode with vLLM (Kwon et al., 2023) and greedy decoding; every front end receives the same prompts and the same evaluation protocol. For VidChapters, the ASR transcript is passed as timestamped text in every row; EATP changes only the visual temporal front end.

5 Results and Analysis

We report results on QVHighlights moment retrieval and VidChapters chapter generation with Qwen3.5-9B in the training-free setting.

5.1 Main results

Tables 1 and 2 separate the headline results by task. Only the two local Qwen3.5-9B rows in each table form a controlled comparison: they share the checkpoint, prompts, resolution, decoding, and native 2 FPS / 768-frame temporal-unit budget. EATP observes denser visual candidates but returns to that native budget before the language model.

EATP improves mIoU from 64.19 to 66.35 and strict-threshold recall from 53.96 to 62.00 while preserving the native temporal-unit budget; loose R@1@0.5 changes from 66.86 to 70.00. On VidChapters with the same video+ASR input, EATP improves over Qwen3.5-9B native uniform by 4.5 F1, 1.4 tIoU, 1.7 SODA, 6.5 CIDEr, and 1.0 GRACE points.

5.2 Ablation study

We use the ablations to answer two questions about EATP. First, does increasing the number of uniformly sampled frames by itself consistently improve temporal understanding? Second, given a denser candidate stream, is it sufficient to simply enlarge the temporal patch, or is content-adaptive grouping necessary? The temporal-budget sweep addresses the first question, while the grouping ablations address the second.

Table 2: QVHighlights moment-retrieval results. FT denotes task-specific fine-tuning. Non-local values are from TimeLens (Zhang et al., 2025a).

Method	Setting	FT	mIoU	R@1@0.5	R@1@0.7
Published reference					
GPT-4o	proprietary MLLM	no	52.1	54.8	38.5
GPT-5	proprietary MLLM	no	56.8	60.4	46.4
Gemini-2.0-Flash	proprietary MLLM	no	60.8	66.4	48.3
Gemini-2.5-Flash	proprietary MLLM	no	64.3	69.4	55.0
VideoChat-Flash-7B (Li et al., 2026b)	short-to-long SFT	no	32.7	30.6	16.7
VideoChat-R1-7B (Li et al., 2025a)	multi-task GRPO/RFT	yes	21.5	19.1	9.4
Time-R1-7B (Wang et al., 2025c)	TVG post-training	yes	49.2	51.5	36.1
TimeSuite (Zeng et al., 2025)	grounded instruction tuning	yes	21.7	16.9	9.9
Grounded-VideoLLM (Wang et al., 2025a)	multi-stage grounding SFT	yes	33.4	33.8	22.5
Controlled Qwen3.5-9B comparison (training-free)					
Qwen3.5-9B (baseline)	native uniform	no	64.19	66.86	53.96
EATP (ours)	adaptive front end	no	**66.35**	**70.00**	**62.00**
Δ vs. baseline	EATP − Qwen3.5-9B baseline	–	+2.16	+3.14	+8.04

Table 3: Temporal-budget ablation with training-free Qwen3.5-9B. QVHighlights uses video-only input; VidChapters uses video+ASR on the full test split. Baseline: native uniform sampling with two-frame temporal patches at the given rate and frame cap. EATP: dense candidate observation with DP grouping at the native temporal-unit budget of the same configuration. Within each budget configuration, the better of Baseline and EATP is bold.

		QVHighlights			VidChapters				
Budget config	Method	mIoU	R@1@0.5	R@1@0.7	CIDEr	$SODA_c$	Seg. F1	Cap. F1	GRACE
1 FPS / 256	Baseline	61.49	62.91	50.45	1.602	0.266	**0.614**	0.433	**59.69**
	EATP	**63.32**	**67.00**	**58.00**	**1.684**	**0.268**	0.602	**0.446**	59.28
2 FPS / 768	Baseline	64.19	66.86	53.96	1.732	0.291	0.616	**0.473**	60.05
	EATP	**65.26**	**67.00**	**60.00**	**1.797**	**0.308**	**0.661**	0.466	**61.10**
4 FPS / 1536	Baseline	63.61	**68.00**	**58.00**	**1.773**	0.272	0.563	**0.483**	56.72
	EATP	**64.55**	67.00	57.00	1.684	**0.287**	**0.632**	0.453	**58.38**

5.2.1 Temporal budget sweep

Table 3 varies the temporal budget from 1 to 4 FPS. A key observation is that simply sampling more frames does not consistently improve performance. For the native uniform baseline on QVHighlights, increasing the sampling rate from 1 to 2 FPS improves mIoU from 61.49 to 64.19, but further increasing it to 4 FPS reduces mIoU to 63.61. A similar non-monotonic trend appears on VidChapters: the baseline GRACE score increases slightly from 59.69 at 1 FPS to 60.05 at 2 FPS, but drops to 56.72 at 4 FPS. Thus, a denser uniform observation stream alone is not sufficient; additional frames need to be allocated and aggregated in a way that preserves useful temporal structure.

EATP addresses this problem by observing a denser candidate stream while adaptively assigning those candidates to a fixed number of temporal units. At the 2 FPS / 768-frame budget, this adaptive allocation gives consistent gains over the corresponding uniform baseline. On QVHighlights, EATP improves mIoU from 64.19 to 65.26 and R@1@0.7 from 53.96 to 60.00. On VidChapters, segment F1 increases from 0.616 to 0.661 and GRACE from 60.05 to 61.10.

5.2.2 Effect of Adaptive Grouping And Maximum Group Length

Table 4 first examines whether the gains of EATP can be explained simply by using larger temporal patches. We replace the content-adaptive DP partition with fixed, content-independent groups of size $K = 4$, 8, or 16.

The fixed-grouping results show that larger temporal patches alone are not sufficient. On QVHighlights, Uniform $K = 4$ achieves 64.60 mIoU, 66.00 R@1@0.5, and 60.00 R@1@0.7. Increasing the fixed group size to $K = 8$ reduces these metrics to 60.38, 61.00, and 52.00, respectively, and $K = 16$ further reduces them. The same trend is more pronounced on VidChapters: GRACE decreases

Table 4: Ablation on temporal grouping and maximum group length with training-free Qwen3.5-9B. The fixed-grouping block replaces EATP's content-adaptive partition with content-independent groups of fixed size K, while keeping the same 4 FPS candidate stream. The maximum-group-length block keeps the candidate stream fixed and varies only the maximum group length $K_{\max}$ allowed by dynamic programming. The main EATP configuration uses $K_{\max} = 16$ for QVHighlights and $K_{\max} = 8$ for VidChapters. QVHighlights uses video-only input, whereas VidChapters uses video+ASR. Boldface indicates the best result within each ablation block.

		QVHighlights			VidChapters				
Control	Configuration	mIoU	R@1@0.5	R@1@0.7	CIDEr	$SODA_c$	Seg. F1	Cap. F1	GRACE
Headline reference									
Main result	**EATP (main)**	66.35	70.00	62.00	1.797	0.308	0.661	0.466	61.10
Fixed aggregation (4 FPS candidates)									
Fixed group	Uniform K=4	**64.60**	**66.00**	**60.00**	**1.559**	**0.267**	**0.615**	**0.435**	**58.51**
	Uniform K=8	60.38	61.00	52.00	1.236	0.191	0.501	0.381	52.20
	Uniform K=16	59.27	59.00	51.00	1.089	0.170	0.457	0.372	50.17
Maximum DP length (fixed candidate stream)									
DP max length	$K_{\max} = 2$ (native baseline)	64.19	66.86	53.96	1.732	0.291	0.616	**0.473**	60.05
	$K_{\max} = 4$	64.60	66.00	60.00	1.559	0.267	0.615	0.435	58.51
	$K_{\max} = 8$	65.31	67.00	60.00	**1.797**	**0.308**	**0.661**	0.466	**61.10**
	$K_{\max} = 16$	**66.35**	**70.00**	**62.00**	1.747	0.291	0.647	0.449	60.21

from 58.51 for $K = 4$ to 52.20 for $K = 8$ and 50.17 for $K = 16$. Therefore, simply aggregating more consecutive frames into a larger temporal patch progressively loses temporal detail rather than reproducing the benefit of EATP.

More importantly, EATP can be compared directly with Uniform $K = 4$; the key difference is whether the group boundaries are fixed or determined adaptively from the video content. With adaptive grouping, EATP improves QVHighlights mIoU from 64.60 to 66.35, R@1@0.5 from 66.00 to 70.00, and R@1@0.7 from 60.00 to 62.00. On VidChapters, CIDEr increases from 1.559 to 1.797, and GRACE from 58.51 to 61.10. This controlled comparison shows that the gain does not come merely from using a wider temporal patch; it comes from adapting *where* temporal group boundaries are placed.

We next vary the maximum group length $K_{\max}$ allowed by the DP partition while keeping the candidate stream fixed. The preferred value is task-dependent. We use $K_{\max} = 16$ for QVHighlights based on moment-retrieval performance, and $K_{\max} = 8$ for VidChapters based on GRACE.

6 Conclusion and Limitations

We present Event-Adaptive Temporal Patching (EATP), a training-free front end for boundary-sensitive video understanding. EATP observes a dense candidate stream and uses dynamic programming to construct a fixed number of contiguous, variable-length temporal groups, allocating finer temporal patches to rapidly changing content while compressing relatively stable regions. Across QVHighlights and VidChapters, EATP improves most reported localization and chaptering measures over the native uniform front end while preserving the same downstream temporal-unit budget. These results support content-adaptive temporal allocation as an effective alternative to uniform sampling under a fixed visual-token budget. EATP is currently evaluated with the Qwen3.5-9B temporal-patch front end; extending the method to other video MLLMs and query-aware grouping remains future work.

AI Use Statement

Generative AI tools were used to assist with manuscript drafting, language editing, organization, and consistency checking. The authors reviewed the resulting text and take full responsibility for the research methodology, experiments, reported results, citations, and all scientific claims in this paper.

REFERENCES

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 vision language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22654–22663, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Fu_FrameFusion_Combining_Similarity_and_Importance_for_Video_Token_Reduction_on_ICCV_2025_paper.html.

Soichiro Fujita, Tsutomu Hirao, Hidetaka Kamigaito, Manabu Okumura, and Masaaki Nagata. SODA: Story oriented dense video captioning evaluation framework. In *Computer Vision – ECCV 2020*, volume 12351 of *Lecture Notes in Computer Science*, pp. 517–531. Springer, 2020. doi: 10.1007/978-3-030-58539-6_31.

Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. VTimeLLM: Empower llm to grasp video moments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14271–14280, 2024. URL https://arxiv.org/abs/2311.18445.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. *CoRR*, abs/1705.00754, 2017. URL http://arxiv.org/abs/1705.00754.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023. doi: 10.1145/3600006.3613165.

Jie Lei, Tamara L. Berg, and Mohit Bansal. QVHighlights: Detecting moments and highlights in videos via natural language queries. In *Advances in Neural Information Processing Systems*, volume 34, pp. 12611–12623, 2021.

Pengyi Li, Irina Abdullaeva, Alexander Gambashidze, Andrey Kuznetsov, and Ivan Oseledets. MaxInfo: A training-free key-frame selection method using maximum volume for enhanced video understanding. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 7198–7207, 2026a.

Xinhao Li, Ziang Yan, Desen Meng, Lu Dong, Xiangyu Zeng, Yinan He, Yali Wang, Yu Qiao, Yi Wang, and Limin Wang. Videochat-r1: Enhancing spatio-temporal perception via reinforcement fine-tuning. *arXiv preprint arXiv:2504.06958*, 2025a. URL https://arxiv.org/abs/2504.06958.

Xinhao Li, Yi Wang, Jiashuo Yu, Xiangyu Zeng, Yuhan Zhu, Haian Huang, Jianfei Gao, Kunchang Li, Yinan He, Chenting Wang, Yu Qiao, Yali Wang, and Limin Wang. Videochat-flash: Hierarchical compression for long-context video modeling. In *International Conference on Learning Representations*, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/b14d7175755b180dc2163e15e3110cb6-Abstract-Conference.html.

Zeqian Li, Shangzhe Di, Zhonghua Zhai, Weilin Huang, Yanfeng Wang, and Weidi Xie. Universal video temporal grounding with generative multi-modal large language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025b. doi: 10.52202/085713-2159. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/5d2e24df9cfaad3189833b819c40b392-Abstract-Conference.html.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Shuhuai Ren, Sishuo Chen, Shicheng Li, Xu Sun, and Lu Hou. TESTA: Temporal-spatial token aggregation for long-form video-language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 932–947. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.66.

Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. TimeChat: A time-sensitive multimodal large language model for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://arxiv.org/abs/2312.02051.

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, Zhuang Liu, Hu Xu, Hyunwoo J. Kim, Bilge Soran, Raghuraman Krishnamoorthi, Mohamed Elhoseiny, and Vikas Chandra. LongVU: Spatiotemporal adaptive compression for long video-language understanding. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 54582–54599. PMLR, 2025.

Enxin Song, Wenhao Chai, Guanhong Wang, Haoyang Zhou, Feiyang Wu, Yucheng Zhang, Tian Ye, Haozhe Chi, Xun Guo, Yanting Zhang, Yan Lu, Jenq-Neng Hwang, and Gaoang Wang. MovieChat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18221–18232, 2024. URL https://arxiv.org/abs/2307.16449.

Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 29118–29128, 2025.

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*, pp. 18992–19001, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Tao_DyCoke_Dynamic_Compression_of_Tokens_for_Fast_Video_Large_Language_CVPR_2025_paper.html.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. CIDEr: Consensus-based image description evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4566–4575, 2015. doi: 10.1109/CVPR.2015.7299087.

Lucas Ventura, Antoine Yang, Cordelia Schmid, and Gül Varol. Chapter-Llama: Efficient chaptering in hour-long videos with LLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18947–18958, 2025.

Haibo Wang, Zhiyang Xu, Yu Cheng, Shizhe Diao, Yufan Zhou, Yixin Cao, Qifan Wang, Weifeng Ge, and Lifu Huang. Grounded-VideoLLM: Sharpening fine-grained temporal grounding in video large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 959–975. Association for Computational Linguistics, 2025a. doi: 10.18653/v1/2025.findings-emnlp.50. URL https://aclanthology.org/2025.findings-emnlp.50/.

Teng Wang, Yixiao Ge, Yuying Ge, Chen Li, and Ying Shan. ARC-Chapter: Structuring hour-long videos into navigable chapters and hierarchical summaries. *arXiv preprint arXiv:2511.14349*, 2025b. URL https://arxiv.org/abs/2511.14349.

Ye Wang, Ziheng Wang, Boshen Xu, Yang Du, Kejun Lin, Zihan Xiao, Zihao Yue, Jianzhong Ju, Liang Zhang, Dingyi Yang, Xiangnan Fang, Zewen He, Zhenbo Luo, Wenxuan Wang, Junqi Lin, Jian Luan, and Qin Jin. Time-r1: Post-training large vision language model for temporal video grounding. In *Advances in Neural Information Processing Systems*, volume 38, 2025c. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/7801b29c93b599b8d0c44138596bdeed-Paper-Conference.pdf.

Zuxuan Wu, Caiming Xiong, Chih-Yao Ma, Richard Socher, and Larry S. Davis. AdaFrame: Adaptive frame selection for fast video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1278–1287, 2019.

Mingze Xu, Mingfei Gao, Zhe Gan, Hong-You Chen, Zhengfeng Lai, Haiming Gang, Kai Kang, and Afshin Dehghan. SlowFast-LLaVA: A strong training-free baseline for video large language models. *arXiv preprint arXiv:2407.15841*, 2024.

Antoine Yang, Arsha Nagrani, Ivan Laptev, Josef Sivic, and Cordelia Schmid. VidChapters-7M: Video chapters at scale. In *Advances in Neural Information Processing Systems*, volume 36, 2023a. Datasets and Benchmarks Track.

Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, and Cordelia Schmid. Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10714–10726, 2023b.

Shoubin Yu, Jaemin Cho, Prateek Yadav, and Mohit Bansal. Self-chained image-language model for video localization and question answering. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-3354. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/f22a9af8dbb348952b08bd58d4734b50-Abstract-Conference.html.

Sicheng Yu, Chengkai Jin, Huanyu Wang, Zhenghao Chen, Sheng Jin, Zhongrong Zuo, Xiaolei Xu, Zhenbang Sun, Bingni Zhang, Jiawei Wu, Hao Zhang, and Qianru Sun. Frame-voyager: Learning to query frames for video large language models. In *International Conference on Learning Representations*, 2025.

Xiangyu Zeng, Kunchang Li, Chenting Wang, Xinhao Li, Tianxiang Jiang, Ziang Yan, Songze Li, Yansong Shi, Zhengrong Yue, Yi Wang, Yali Wang, Yu Qiao, and Limin Wang. Timesuite: Improving mllms for long video understanding via grounded tuning. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/5e8309c9ca683e11672e3dbcd4b87776-Abstract-Conference.html.

Jun Zhang, Teng Wang, Yuying Ge, Yixiao Ge, Xinhao Li, Ying Shan, and Limin Wang. TimeLens: Rethinking video temporal grounding with multimodal LLMs. *arXiv preprint arXiv:2512.14698*, 2025a. URL https://arxiv.org/abs/2512.14698.

Shaojie Zhang, Jiahui Yang, Jianqin Yin, Zhenbo Luo, and Jian Luan. Q-frame: Query-aware frame selection and multi-resolution adaptation for video-llms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22056–22065, 2025b. URL https://openaccess.thecvf.com/content/ICCV2025/html/Zhang_Q-Frame_Query-aware_Frame_Selection_and_Multi-Resolution_Adaptation_for_Video-LLMs_ICCV_2025_paper.html.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.

A Additional Experiments

A.1 Full tables for the motivating analysis

Tables 5, 6, and 7 contain the motivating-analysis values used in Figure 2 and Section 2.

Table 5: Within-bin association between boundary proximity and Boundary@5s recall on VidChapters. A ground-truth boundary is a hit if its nearest sampled frame is within 0.5 s. Only bins with both hit and miss boundaries are shown. The unit of analysis is a ground-truth boundary, so the table reports recall, not F1. These observational comparisons control actual sampling density only; they do not establish a causal effect.

Interval	# Hit	# Miss	Recall hit	Recall miss	Uplift
1.0–1.5 s	10,243	2,052	48.25	47.22	+1.03
1.5–2.0 s	2,717	1,977	45.23	43.70	+1.53
>2.0 s	3,917	7,359	34.57	31.61	+2.96

Table 6: Boundary@5s performance of the native uniform-sampling Qwen3.5-9B baseline on VidChapters, stratified by video duration. The 768-frame cap binds beyond 384 seconds: the effective sampling rate falls and the mean sampling interval grows, while boundary precision, recall, and F1 degrade. This is sampling-density context; duration also covaries with video complexity and is not a causal treatment comparison.

Duration	#Videos	Eff. FPS	Interval	Prec.	Rec.	F1
$\leq$384 s	1,528	2.00	0.50 s	48.43	61.60	54.23
384–600 s	1,448	1.58	0.64 s	44.33	54.73	48.99
600–900 s	1,918	1.06	0.96 s	40.08	52.32	45.39
900–1200 s	962	0.75	1.35 s	32.68	45.95	38.20
1200–1800 s	771	0.52	1.94 s	29.07	43.81	34.95
$>$1800 s	892	0.31	3.36 s	21.10	30.30	24.87

Table 7: Moment-retrieval quality of the Qwen3.5-9B baseline on QVHighlights as a function of the uniform sampling rate. All videos are approximately 150 seconds, so the frame caps never bind and the visual-token cost scales linearly with the rate.

Sampling rate	Frames	mIoU	R@1@0.5	R@1@0.7	Rel. tokens
1 FPS	$\sim$150	61.49	62.91	50.45	1$\times$
2 FPS	$\sim$300	64.19	66.86	53.96	2$\times$
4 FPS	$\sim$600	63.61	68.00	58.00	4$\times$