HTSC-AT: HIERARCHICAL TEMPORAL SEMANTIC COMPRESSION WITH ADAPTIVE TOKENIZATION FOR EFFICIENT LONG VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Long video understanding with multimodal large language models (MLLMs) is bottlenecked by quadratic attention complexity and severe temporal redundancy in visual tokens. We present HTSC-AT, a three-stage framework for efficient long-context reasoning on lightweight open-source MLLMs. Stage 1 introduces adaptive visual tokenization driven by motion-semantic change intensity, reducing token count by 66.3% while improving QA accuracy by +5.3%. Stage 2 performs event-aware temporal compression via dual-branch boundary detection and hierarchical intra-event clustering, achieving up to 720× compression while preserving semantically critical evidence. Stage 3 employs a sparse-dense hybrid decoder that maintains dense intra-event attention and structured sparse inter-event attention, reducing peak GPU memory from 28.7 GB to 10.4 GB. Across MLVU, VideoMME, LongVideoBench, and LVBench, HTSC-AT achieves state-of-the-art results among open-source small-scale models—outperforming VideoXL2-8B by 4.1 points on MLVU Test—while supporting over 10,000 frames on a single-node 8×H800 GPU setup.

1 INTRODUCTION

Multimodal large language models (MLLMs) have extended visual reasoning from images to video (Liu et al., 2023; Zhu et al., 2023; Alayrac et al., 2022; Li et al., 2023; Zhang et al., 2025). Long video understanding (LVU), however, requires processing thousands of frames while retaining evidence for reasoning across distant events (Qin et al., 2025; Shu et al., 2025; Zhou et al., 2024). Visual token counts grow with video duration, whereas full self-attention has quadratic cost. Temporal redundancy increases this cost without necessarily adding useful evidence; aggressive compression can instead remove details needed for temporal grounding and cross-segment question answering.

Existing approaches address different parts of this problem. Visual token compression reduces the input sequence through merging, pooling, or memory banks (Li et al., 2024b;a; Wang et al., 2024c; He et al., 2024a), but can lose fine-grained information. Sparse attention reduces computation (Li et al., 2025; Yuan et al., 2025; Lu et al., 2025; He et al., 2024b), although connectivity patterns may not follow video event structure. Chunking and streaming process long sequences incrementally (Shu et al., 2025; Wang et al., 2024b; Chen et al., 2025a; Xue et al., 2024); fixed partitions can fragment events, and historical KV dependencies constrain scalability. KV-cache methods retain selected context (Liu et al., 2025; Li et al., 2024c; Cai et al., 2024), motivating closer integration of compression and cross-event reasoning. Appendix D provides a detailed survey.

These challenges are particularly relevant to lightweight open-source MLLMs at the 7B/8B scale, where memory budgets constrain the amount of video evidence available to the language model. We propose **HTSC-AT**, a three-stage framework that combines adaptive visual token allocation, event-coherent compression, and structured attention with a frozen MLLM backbone. Its contributions are:

- An adaptive visual tokenizer that allocates tokens according to motion-semantic change intensity, retaining greater detail in dynamic regions.

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.

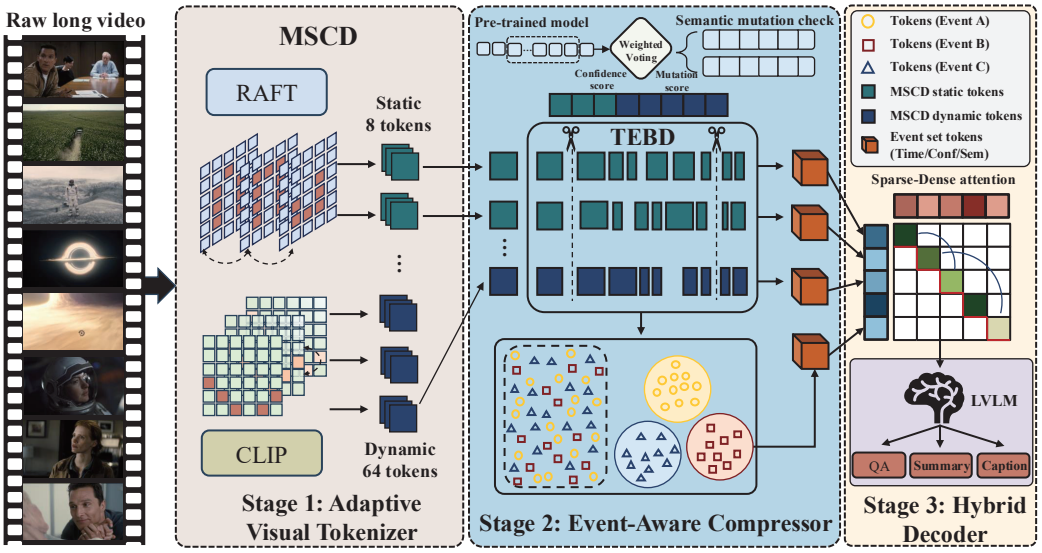

Figure 1: Overview of the proposed HTSC-AT framework.

- An event-aware temporal compressor that combines dual-branch boundary detection with hierarchical clustering to preserve event structure and distinctive evidence.
- A sparse-dense hybrid decoder that couples dense intra-event attention with sparse connections between adjacent and key events.

Experiments on MLVU, VideoMME, LongVideoBench, and LVBench show strong accuracy and efficiency among lightweight open-source models. HTSC-AT improves MLVU Test accuracy by 4.1 points over Video-XL2-8B and supports over 10,000 frames on a single-node 8×H800 setup. Ablations examine the contribution of each stage to token reduction, long-context understanding, and memory use.

2 Method

2.1 Overall Architecture

HTSC-AT (Hierarchical Temporal Semantic Compression with Adaptive Tokenization) is a trainable compression front end for a frozen LVLM. Uniform tokenization can over-represent static content, while aggressive downsampling can discard evidence needed for cross-event reasoning. The three stages in Fig. 1 address these limitations at complementary temporal scales.

Stage 1, the adaptive visual tokenizer $\mathcal{A}_\theta$, maps raw frames to a variable-length sequence $\mathcal{T}^{(1)}$, allocating more tokens where motion-semantic change is greater. **Stage 2**, the event-aware temporal compressor $\mathcal{C}_\psi$ (TEBD), partitions this sequence into semantically coherent events and applies hierarchical clustering to retain representative and distinctive summary tokens $\mathcal{T}^{(2)}$. **Stage 3** combines dense interactions within events with sparse token-level interactions between neighboring and anchor events. Temporal positional encodings preserve the location and order of retained evidence, and a lightweight adapter maps the contextualized tokens to the frozen LVLM. Together, the stages reduce redundancy while retaining local detail and routes for long-range interaction.

Formally, let an input video be $\mathcal{V} \in \mathbb{R}^{T\times H\times W\times 3}$, where T is the total number of sampled frames and $H \times W$ is the spatial resolution. The full pipeline computes:

$$\hat{y} = \mathcal{D}_\phi(\mathcal{C}_\psi(\mathcal{A}_\theta(\mathcal{V}))\,,\, q)\,, \tag{1}$$

where q is the text query. Only θ and ψ (plus a lightweight adapter) are updated during training; ϕ remains frozen.

For a fixed video, the compressed tokens, event partition, and sparse adjacency are independent of the question. Compression is content-adaptive and event-aware; query conditioning occurs during downstream answer generation by the frozen LVLM.

2.2 Stage 1: Adaptive Visual Tokenizer via Motion-Semantic Change Detection

Conventional tokenizers assign a fixed number of tokens per frame, leading to over-representation of near-static scenes and under-representation of semantically rich, high-motion events. We instead allocate tokens as a function of each frame's informational novelty relative to its predecessor.

2.2.1 Inter-Frame Semantic Change Intensity.

Let $\mathrm{CLIP}(I_t) \in \mathbb{R}^d$ denote the ℓ_2-normalized CLIP visual embedding of frame I_t, and let $\mathrm{Flow}(I_t, I_{t-1}) \in \mathbb{R}^{H' \times W' \times 2}$ denote the dense optical flow field estimated by a lightweight RAFT model. We define the **inter-frame semantic change intensity** as:

$$\Delta S_t = \left\|\mathrm{CLIP}(I_t) - \mathrm{CLIP}(I_{t-1})\right\|_2 + \beta \cdot \left\|\mathrm{Flow}(I_t, I_{t-1})\right\|_1, \tag{2}$$

where $\beta = 0.3$ balances static semantic displacement (CLIP embedding distance) against dynamic motion magnitude (optical flow ℓ_1 norm). The dual-stream design is critical: CLIP differences alone fail to detect high-frequency motion in semantically invariant scenes (*e.g.*, a camera pan over a static background), while optical flow alone cannot encode semantic scene transitions.

2.2.2 Semantic-Density-Driven Token Allocation.

Given $\{\Delta S_t\}_{t=1}^T$, we compute the per-frame token count:

$$N_t = N_{\min} + \alpha \cdot \mathrm{Softmax}(\Delta S_t)\,, \tag{3}$$

where the Softmax normalizes $\{\Delta S_t\}$ over the full temporal sequence to $[0, 1]$, $N_{\min} = 8$ is the minimum token budget for near-static frames, and α scales the dynamic range such that $N_t \leq 64$ for maximally dynamic frames. The resulting token sequence is:

$$\mathcal{T}^{(1)} = \left[\mathbf{z}_{1,1}, \ldots, \mathbf{z}_{1,N_1},\ \mathbf{z}_{2,1}, \ldots, \mathbf{z}_{T,N_T}\right] \in \mathbb{R}^{N^{(1)} \times d}, \tag{4}$$

where $N^{(1)} = \sum_{t=1}^T N_t$ and $\mathbf{z}_{t,i} \in \mathbb{R}^d$ is the i-th spatial token of frame I_t extracted by the CLIP visual encoder. This formulation achieves an average compression ratio exceeding 60% relative to uniform 32-token-per-frame baselines, while yielding a $+5.3\%$ QA accuracy improvement.

2.3 Stage 2: Event-Aware Temporal Compressor (TEBD)

Even after adaptive tokenization, $\mathcal{T}^{(1)}$ may still contain thousands of tokens for hour-long videos, far exceeding practical LVLM context limits. The TEBD module addresses this by (i) detecting event boundaries with semantic fidelity, and (ii) compressing each event into a compact summary token set via hierarchical clustering.

2.3.1 Dual-Branch Event Boundary Detection

Branch 1: Pre-trained Temporal Localization. We leverage a distilled ActionFormer (3M parameters) applied over sliding windows of $w = 30$ tokens with stride $s = 10$. ActionFormer outputs a per-position **boundary confidence score** $C_1(t) \in [0, 1]$ and an event-type label $\ell_t \in \{\text{dialogue, action, static, } \ldots\}$. For distillation and implementation details, see Appendix A.

Branch 2: Semantic Discontinuity Scoring. We compute the cosine similarity between adjacent token features:

$$\mathrm{sim}_t = \frac{\mathbf{feat}_t \cdot \mathbf{feat}_{t-1}}{\|\mathbf{feat}_t\|_2 \cdot \|\mathbf{feat}_{t-1}\|_2}, \quad \mathrm{MS}_t = 1 - \mathrm{sim}_t, \tag{5}$$

where $\mathbf{feat}_t \in \mathbb{R}^d$ is the CLIP feature of the t-th token in $\mathcal{T}^{(1)}$, and $\mathrm{MS}_t \in [0, 1]$ is the **semantic mutation score**. An adaptive boundary threshold is defined as:

$$\theta = \alpha_s \cdot \max_t \mathrm{MS}_t, \tag{6}$$

where α_s is a scene-adaptive coefficient ($\alpha_s = 0.65$ for lecture-style videos, $\alpha_s = 0.80$ for action films, default $\alpha_s = 0.75$). Positions with $\mathrm{MS}_t \geq \theta$ are marked as candidate boundaries; a minimum inter-boundary gap of 2 s and a maximum event duration of 30 s are enforced to prevent over- and under-segmentation.

Boundary Fusion. The two branches are fused via a weighted decision score:

$$S_b(t) = 0.6 \cdot C_1(t) + 0.4 \cdot \widetilde{\mathrm{MS}}_t, \tag{7}$$

where $\widetilde{\mathrm{MS}}_t = \frac{\mathrm{MS}_t - \min(\mathrm{MS})}{\max(\mathrm{MS}) - \min(\mathrm{MS})}$ is the min-max normalized mutation score. Positions with $S_b(t) \geq 0.5$ are confirmed as final event boundaries (with post-processing to merge sub-second boundaries and split segments >30 s). The output is an ordered event list $\mathcal{E} = \{E_e\}_{e=1}^{E}$, where each $E_e = [\mathrm{start}_e, \mathrm{end}_e, \ell_e, S_b^e]$.

2.3.2 Hierarchical Intra-Event Clustering.

For each event E_e, let $\mathcal{F}_e = \{\mathbf{feat}_t\}_{t \in E_e} \in \mathbb{R}^{N_{E_e} \times d}$ be the set of token features within that event.

Token Pre-filtering. We compute the semantic saliency score of each token relative to the event mean $\overline{\mathbf{feat}}_e$:

$$\mathrm{SS}_t = \cos\big(\mathbf{feat}_t, \overline{\mathbf{feat}}_e\big)\,. \tag{8}$$

Tokens with $\mathrm{SS}_t < 0.3$ are discarded as low-salience; tokens corresponding to high-motion frames (top-30% by ΔS_t) are retained unconditionally to preserve action-critical information.

Coarse Clustering. We apply Mini-Batch K-Means (batch size 64) with an entropy-adaptive cluster count:

$$H_e = -\sum_{i=1}^{n} p_i \log_2 p_i, \qquad K_e = \max(2,\, \min(\lfloor H_e \times 2 \rfloor,\, 5))\,, \tag{9}$$

where p_i is the fraction of tokens within event E_e belonging to semantic class i (from ActionFormer labels), and H_e is the Shannon entropy of the per-class distribution. This formulation assigns more clusters to semantically diverse events while conservatively compressing homogeneous ones. As a fallback for degenerate entropy values ($H_e \approx 0$), a duration-based formula $K_e = \max(2, \min(5, \lfloor \mathrm{Dur}_{E_e}/10 \rfloor))$ is employed.

Fine Clustering via Mahalanobis Outlier Selection. Within each coarse cluster C_k, we retain the top-1 or top-2 tokens with the highest Mahalanobis distance from the cluster mean to capture structurally distinct, semantically significant tokens. The final per-event summary merges cluster centroids with these outlier tokens, yielding $K'_e \leq 10$ tokens in practice. For videos exceeding one hour, TEBD activates a three-tier hierarchical clustering regime (Appendix C).

2.3.3 Cross-Event Alignment and Temporal Positional Encoding.

Redundancy Removal. Adjacent events with pairwise similarity

$$\mathrm{Sim}_{E_1,E_2} = \frac{1}{K'_{E_1} K'_{E_2}} \sum_{t_1 \in E_1} \sum_{t_2 \in E_2} \cos(\mathbf{feat}_{t_1}, \mathbf{feat}_{t_2}) \geq \tau_{\text{dup}} \tag{10}$$

(default $\tau_{\text{dup}} = 0.85$; scene-specific values are reported in Appendix A) are treated as semantically redundant; only the first event's full token set is retained, and subsequent duplicates are collapsed to a single highest-confidence token.

Temporal Positional Encoding (TPE). To encode the multi-scale temporal structure of each summary token, we define a three-dimensional positional feature:

$$T_{\text{abs}} = \frac{\mathrm{Start}_{E_e}}{\mathrm{Dur}_V}, \quad T_{\text{rel}} = \frac{\mathrm{Idx}_{E_e}}{N_V}, \quad T_{\text{lev}} = 0.4 \cdot \frac{\mathrm{Dur}_{E_e}}{\mathrm{Dur}_V} + 0.6 \cdot \frac{1}{N_{E_e}} \sum_{t \in E_e} \mathrm{SS}_t, \tag{11}$$

corresponding to absolute temporal position, relative sequence order, and event-level semantic importance, respectively. These scalars are projected to the token embedding space by an MLP, and the resulting positional vector is fused with each summary token feature:

$$\mathrm{TPE}_{E_e} = \mathrm{MLP}([T_{\text{abs}}, T_{\text{rel}}, T_{\text{lev}}])\,, \quad \mathbf{feat}'_t = \mathrm{LayerNorm}(\mathbf{feat}_t + \mathrm{TPE}_{E_e})\,. \tag{12}$$

The output of Stage 2 is the event-level token matrix $\mathcal{T}^{(2)} = [\boldsymbol{\tau}_{1,1}, \ldots, \boldsymbol{\tau}_{E,K'_E}] \in \mathbb{R}^{N^{(2)} \times d}$, where $N^{(2)} = \sum_{e=1}^{E} K'_e \ll N^{(1)}$.

2.4 Stage 3: Sparse-Dense Hybrid Attention Decoder

2.4.1 Motivation.

Even after Stage 2 compression, $\mathcal{T}^{(2)}$ may span hundreds of event segments for very long videos, rendering standard full self-attention ($\mathcal{O}(N^2)$ complexity) prohibitive. Sparse attention alternatives such as Longformer and BigBird apply structurally agnostic sparsity patterns that ignore the event hierarchy, fragmenting the causal chains that cross-event temporal reasoning demands. We therefore design a structured **Sparse-Dense Hybrid Attention** mechanism that explicitly models (i) *dense* intra-event interactions and (ii) *sparse* inter-event interactions restricted to structurally meaningful connections.

2.4.2 Intra-Event Dense Attention.

For event e with summary tokens $\{\boldsymbol{\tau}_{e,1}, \ldots, \boldsymbol{\tau}_{e,K'_e}\}$, we form query, key, and value matrices $\mathbf{Q}_e, \mathbf{K}_e, \mathbf{V}_e \in \mathbb{R}^{K'_e \times d}$ and apply full self-attention:

$$\mathrm{Attn}^{(e)}_{\text{dense}} = \mathrm{Softmax}\left(\frac{\mathbf{Q}_e \mathbf{K}_e^\top}{\sqrt{d}}\right)\mathbf{V}_e. \tag{13}$$

Since $K'_e \leq 10$ by design, the per-event computation is $\mathcal{O}(K'^2_e) = \mathcal{O}(100)$, making dense intra-event attention computationally negligible while fully capturing fine-grained local semantic interactions.

2.4.3 Inter-Event Sparse Attention.

Let E denote the total number of events. We define a binary inter-event adjacency matrix $\mathbf{A} \in \{0,1\}^{E \times E}$ based on structural priors:

$$A_{ij} = \begin{cases} 1, & \text{if } |i-j| \leq 1 \quad \text{(temporal adjacency)}, \\ 1, & \text{if } i \in \mathcal{K} \text{ or } j \in \mathcal{K} \quad \text{(key-event anchoring)}, \\ 0, & \text{otherwise}, \end{cases} \tag{14}$$

where $\mathcal{K}$ contains at most six distinct anchors. For the Table 4 setting, $m = |\mathcal{K}| = 6$: the first and last events provide structural anchors, and four content anchors are selected by stratified top-1 selection over four coarse temporal bins using TEBD boundary-confidence saliency. Ordinary events connect to all anchors, and anchors connect back to all events. This budget is independent of video length. Attention operates on all retained summary tokens of each connected event, preserving token-level evidence rather than pooling each event into one vector. Inter-event sparse attention for event i is then:

$$\mathrm{Attn}^{(i)}_{\text{sparse}} = \sum_{j:\, A_{ij}=1} \mathrm{Softmax}\left(\frac{\mathbf{Q}_i \mathbf{K}_j^\top}{\sqrt{d}}\right)\mathbf{V}_j. \tag{15}$$

Let $\bar{C} = E^{-1}\sum_{i,j} A_{ij}$ denote the average number of allowed event blocks per query event, including self blocks. The bounded anchor graph satisfies $\bar{C} \leq C = 2m + 3 = 15$; this is a conservative bound, since overlapping neighbor and anchor connections are counted only once in $\mathbf{A}$. For fixed feature dimension, the two attention branches require

$$\mathcal{O}\big(EK'^2 + CEK'^2\big) = \mathcal{O}\big((1+C)EK'^2\big), \tag{16}$$

which is linear in E for bounded m and K', compared with $\mathcal{O}(E^2K'^2)$ for full token attention. Each allowed event block retains all $K'_iK'_j$ token interactions. The bound assumes that masked-out blocks are skipped during computation. For $E = 300$ and $K' = 5$, the analytical score-entry budget is $(1+15) \times 300 \times 25 = 0.12$M per head per layer. Appendix G distinguishes this budget from exact mask counts.

The full hybrid attention output for event i is:

$$\mathbf{H}_i = \mathrm{Attn}^{(i)}_{\text{dense}} + \mathrm{Attn}^{(i)}_{\text{sparse}}. \tag{17}$$

The contextualized sequence $\{\mathbf{H}_e\}_{e=1}^{E}$ is projected by a lightweight adapter ($<$1M parameters) and concatenated with text query q before being forwarded to the frozen LVLM language head.

2.5 Training Objectives

HTSC-AT's trainable parameters—the MSCD detector, TEBD module, and the hybrid-attention adapter—are optimized end-to-end via a composite loss:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\mathrm{QA}} + \lambda_2 \mathcal{L}_{\mathrm{cap}} + \lambda_3 \mathcal{L}_{\mathrm{temp}}, \tag{18}$$

where $\mathcal{L}_{\mathrm{QA}}$ is answer-token cross-entropy, $\mathcal{L}_{\mathrm{cap}}$ is caption-token cross-entropy for global narrative preservation, and $\mathcal{L}_{\mathrm{temp}}$ is the temporal consistency term. Appendix F provides the full definitions. Only the MSCD, TEBD, and hybrid-attention adapter are optimized; the LVLM backbone remains frozen. The three-phase loss-weight schedule is specified in Appendix A.3. The current objective does not include explicit temporal localization supervision, motivating a dedicated grounding objective in future work.

3 Experiments

3.1 Experimental Setup

Datasets. We evaluate HTSC-AT on seven long-video understanding benchmarks spanning 5-minute to 2-hour content: MLVU (Zhou et al., 2024), VideoMME (Fu et al., 2024), LongVideoBench (LVB) (Wu et al., 2024), LVBench (Wang et al., 2024a), VideoEval-Pro (Ma et al., 2025), CharadesSTA (Gao et al., 2017), and V-STaR (Cheng et al., 2025). LongVideoBench serves as the primary benchmark for all ablation studies. Full dataset descriptions are provided in Appendix D.

Evaluation metrics. We report benchmark QA accuracy and, for the ablations, multi-hop QA accuracy, ROUGE-L, and temporal grounding recall (R@1, R@5), supplemented by the **Long-Context Understanding Score (LCUS)**:

$$\mathrm{LCUS} = \frac{1}{|\mathcal{Q}_{>3}|} \sum_{q \in \mathcal{Q}_{>3}} \mathbf{1}[\hat{a}_q = a_q^*], \tag{19}$$

where $\mathcal{Q}_{>3}$ is the nonempty subset of questions requiring information from more than three non-contiguous video segments, and $\hat{a}_q$ and a_q^* are the predicted and reference answers. LCUS is reported as a percentage and measures accuracy within this subset. Overall QA evaluates the broader question set, whereas multi-hop QA evaluates a designated reasoning subset. The evaluation design treats multi-hop questions as harder than the LCUS subset, and the broader QA set includes simpler questions; this motivates an expected accuracy ordering, not a mathematical constraint on measured scores.

Implementation. The backbone is a *frozen* Qwen2.5-VL-7B (Wang et al., 2025); only the MSCD tokenizer, TEBD compressor, and lightweight hybrid-attention adapter ($<$1M parameters) are trained. All experiments run on $8\times$H800 GPUs. Full training details are provided in Appendix A.

3.2 Comparison with State-of-the-Art

Table 1 compares representative open-source models at $\geq$30B and $\leq$8B parameters across eight evaluation splits. The large-model results provide context; the comparisons below concern the small-model group.

On **MLVU Test**, HTSC-AT achieves 56.3%, exceeding Video-XL2-8B by 4.1 points and VideoLLaMA3-8B by 8.6 points. On **VideoMME**, it ranks second at 71.2%, close to Qwen3-VL-8B (71.4%). On **LongVideoBench**, it reaches 65.2%, surpassing VideoChat-Flash-8B (64.7%). These results show competitive performance across the evaluated long-video QA benchmarks.

Table 1: Comparison with state-of-the-art methods. **Bold**: best among small models; underline: second best among small models. "–": not reported by the original work.

Model	MLVU Dev	MLVU Test	Video MME	Long Video	LVBench	VideoEval -Pro	Charades -STA	V-STaR
Open-source large models (≥30B parameters)								
Qwen3-VL-30B (Qwen Team, 2025)	81.3	–	74.5	64.6	62.5	–	63.5	–
Qwen2.5-VL-72B (Wang et al., 2025)	74.6	–	73.3	60.7	47.3	–	50.9	–
InternVL3-78B (Chen et al., 2025b)	79.5	–	72.7	65.7	–	–	–	–
LLaVA-Video-72B (Zhang et al., 2024)	74.4	–	70.6	61.9	–	–	–	–
Open-source small models (≤8B parameters)								
InternVL2.5-8B (Chen et al., 2024)	68.9	–	64.2	60.0	–	24.6	–	7.8
Qwen2.5-VL-7B (Wang et al., 2025)	70.2	–	65.1	56.0	45.3	27.7	43.6	7.6
InternVL3-8B (Chen et al., 2025b)	71.4	–	66.3	58.8	–	24.4	–	10.1
Qwen3-VL-8B (Qwen Team, 2025)	**78.1**	–	**71.4**	59.5	58.0	28.4	56.0	12.4
LLaVA-Video-7B (Zhang et al., 2024)	70.8	–	63.3	58.2	–	24.2	–	12.2
VideoLLaMA3-8B (Zhang et al., 2025)	73.0	47.7	66.2	59.8	45.3	–	**60.7**	**23.1**
Video-XL2-8B (Qin et al., 2025)	74.8	52.2	66.6	61.0	48.5	28.6	54.2	21.3
VideoChat-Flash-8B (Li et al., 2024a)	74.6	–	65.3	64.7	48.2	27.0	48.0	–
HTSC-AT (Ours, <8B)	78.0	**56.3**	71.2	**65.2**	**58.9**	**28.8**	58.3	17.7

Table 2: Ablation of the MSCD adaptive tokenizer on LongVideoBench. QA denotes multi-hop QA.

Method	Avg. Tokens	Reduction	QA Acc (%)	Inference (s)
Baseline A: Uniform (32 tok/frame)	1,840	–	58.4	92
HTSC-AT (Ours)	**620**	**66.3%**	**63.7**	**48**

HTSC-AT also leads the small-model group on **LVBench** (58.9%) and **VideoEval-Pro** (28.8%), compared with 58.0% for Qwen3-VL-8B and 28.6% for Video-XL2-8B, respectively. The smaller margins on these benchmarks complement the larger improvement on MLVU Test.

On **CharadesSTA**, HTSC-AT reaches 58.3%, trailing VideoLLaMA3-8B by 2.4 points. This suggests a potential benefit from explicit temporal grounding supervision, which is absent from our objective (Eq. 18); the result comparison alone does not establish the cause of this gap.

Inference efficiency. Processing time is at most $1/10$ of video duration for videos up to 60 minutes and $1/15$ for videos up to 2 hours.

3.3 Ablation: Adaptive Visual Tokenizer (MSCD)

Setup. To isolate the MSCD contribution, we compare HTSC-AT against **Baseline A**, which replaces the adaptive tokenizer with uniform sampling at a fixed budget of 32 tokens per frame, while sharing all other modules (TEBD, hybrid decoder, frozen backbone). The evaluation corpus consists of ≈560 LongVideoBench segments (≈12 min each, max 200 frames after pre-processing).

Results.

As shown in Table 2 and Figure 2, MSCD reduces total token count by **66.3%** (620 vs. 1,840 average tokens) while raising multi-hop QA accuracy from 58.4% to **63.7%** (+5.3 pp) and cutting inference time from 92 s to **48 s** (47.8% speedup). These results improve token count, accuracy, and inference time together. Appendix E provides the frame-redundancy analysis motivating adaptive allocation.

3.4 Ablation: Event-Aware Temporal Compressor (TEBD)

Setup. We compare against fixed 60-second windows (Baseline B) (Li et al., 2024a; Shu et al., 2025), random 30–90 s windows (Baseline C), and single-level global clustering (Baseline D). All strategies share the same MSCD front-end and hybrid decoder.

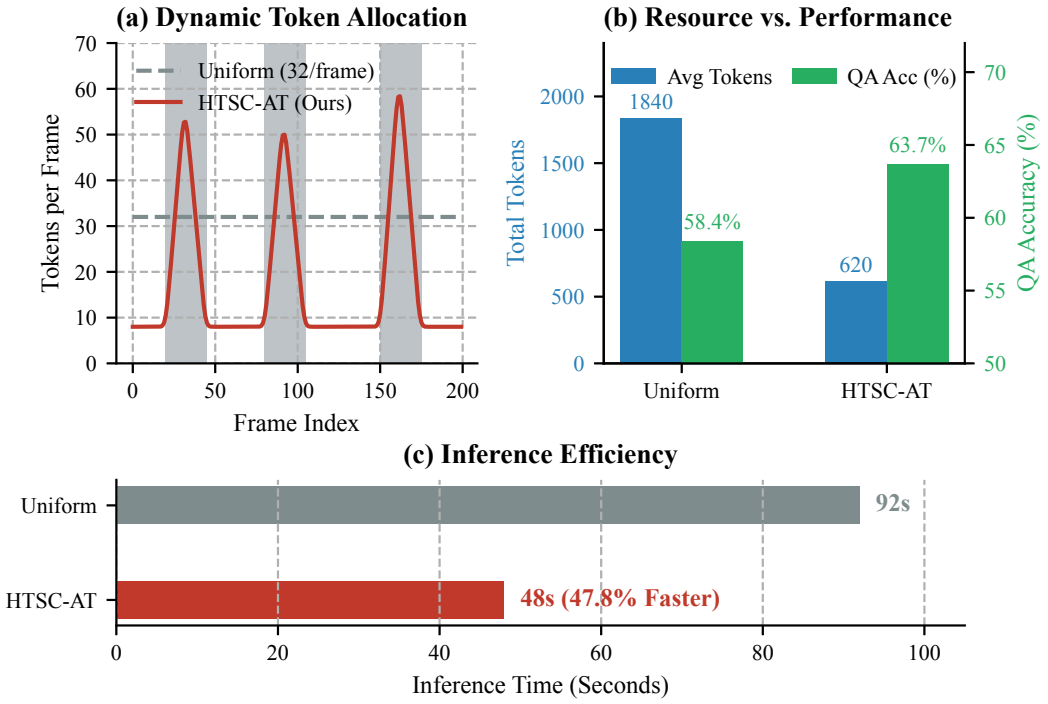

Figure 2: **HTSC-AT vs. Uniform Sampling on LongVideoBench. (a)** Per-frame token allocation of HTSC-AT against uniform baseline across a representative clip. **(b)** Token budget and multi-hop QA accuracy comparison between the two methods. **(c)** Inference time comparison, demonstrating the throughput gain of adaptive allocation.

Table 3: Ablation of video segmentation strategies on LongVideoBench.**Bold**: best result. QA denotes multi-hop QA.

Segmentation Strategy	LCUS (%)	QA Acc (%)	ROUGE-L (%)	R@1 (%)	R@5 (%)	Compression
Baseline B: Fixed 60s window	43.1	39.8	34.5	33.5	55.1	480×
Baseline C: Random window (30–90s)	38.5	35.1	30.1	29.8	49.7	450×
Baseline D: Single-level clustering	41.2	37.6	32.8	31.2	52.4	800×
Ours (TEBD + Hier. Clustering)	**50.8**	**46.2**	**40.2**	**38.7**	**62.3**	**720×**

Results. Table 3 shows that TEBD achieves 50.8% LCUS, exceeding fixed and random windows by 7.7 and 12.3 points, respectively, with the highest QA, ROUGE-L, R@1, and R@5 scores. Single-level clustering compresses more aggressively (800×) but achieves only 41.2% LCUS; TEBD retains better performance at 720× compression. These results favor event-aware segmentation over compression ratio alone. The contrast with single-level clustering is particularly informative: its shorter representation does not yield higher LCUS or multi-hop QA accuracy. The gains in R@1 and R@5 also support retaining event boundaries and temporal annotations alongside semantic summaries. Within this comparison, the highest compression ratio is therefore not the most effective operating point for downstream understanding.

3.5 Ablation: Sparse-Dense Hybrid Attention Decoder

3.5.1 Attention Mechanism Comparison.

Setup. We compare four attention mechanisms applied to an identical compressed token sequence ($N^{(2)} = 1{,}500$, $E = 300$, $K' = 5$): Full Self-Attention, Longformer (Beltagy et al., 2020), Big-

Table 4: Attention mechanism comparison on $N^{(2)} = 1{,}500$ compressed tokens ($\sim$1 h video). Score entries are analytical budgets per head per layer, not measured FLOPs; see Appendix G. ASE denotes Attention Score Entries. QA denotes multi-hop QA. **Bold**: best result.

Method	QA Acc (%)	LCUS (%)	ASE (M)	GPU Mem. (GB)	Max Context
Full Self-Attention	53.8	56.4	2.25	28.7	$\leq$2k
Longformer (Beltagy et al., 2020) (ω=128, g=64)	52.5	54.2	0.29	19.3	$\approx$8k
BigBird (Zaheer et al., 2020) (r=64, ω=128, g=64)	55.6	56.8	0.38	20.1	$\approx$10k
Ours (Sparse-Dense)	**57.4**	**59.3**	**0.12**	**10.4**	$>$10k

Table 5: Ablation of sparse connectivity strategies (Eq. 14). **Bold**: best result. QA denotes multi-hop QA.

Sparse Connectivity	QA Acc (%)	LCUS (%)
Adjacent events only	52.1	55.3
Key events only (structural + content anchors)	51.8	53.5
Random (3 events sampled per query event)	49.5	52.7
Adjacent + Key events (Ours)	**57.4**	**59.3**

Bird (Zaheer et al., 2020), and our Sparse-Dense Hybrid Attention. All conditions share the frozen Qwen2.5-VL-7B backbone.

Results. Table 4 reports attention score entries (QK-pair counts, in millions) as a hardware-independent analytical measure, not FLOPs. For $N = 1{,}500$, the tabulated budgets are $N^2 = 2.25$M for full attention, $(\omega + g)N \approx 0.29$M for Longformer, $(r + \omega + g)N \approx 0.38$M for BigBird, and $(1 + C)EK'^2 = 0.12$M for HTSC-AT with $m = 6$ and $C = 15$. These are analytical budgets rather than profiler measurements or exact deduplicated mask counts. Our budget is $18.75\times$ smaller than full attention, alongside the best multi-hop QA accuracy (57.4%) and LCUS (59.3%), a $2.8\times$ reduction in peak GPU memory (28.7 to 10.4 GB), and context support beyond 10k tokens. The structured mask aligns token interactions with event boundaries, temporal neighbors, and content-selected global anchors, rather than relying on event-agnostic sparse patterns.

3.5.2 Sparse Connectivity Strategy.

Table 5 compares adjacency rules. Combining neighbors and anchors improves multi-hop QA by 5.3 points over neighbors alone and 5.6 points over anchors alone. The random baseline reaches 49.5%, but samples only three events per query event; it is not degree-matched to the six-anchor design and does not isolate structure from connectivity budget.

4 Conclusion

We presented HTSC-AT, a task-aware long video understanding framework for lightweight open-source MLLMs. The method combines adaptive motion-semantic token allocation, event-boundary-guided hierarchical temporal compression, and sparse-dense hybrid attention decoding to jointly optimize accuracy, memory use, and inference efficiency. Experimental results show that adaptive tokenization reduces the token count by 66.3%, while the hybrid decoder reduces peak memory usage from 28.7 to 10.4,GB. HTSC-AT achieves state-of-the-art performance among lightweight open-source MLLMs while maintaining practical long-context scalability. The framework supports over 10,000 frames on a single node equipped with eight H800 GPUs, showing strong potential for real-world long-form video analysis.

AI USE STATEMENT

No generative AI assistance was employed in conductin the research or preparing this manuscript.

REPRODUCIBILITY STATEMENT

Implementation details, training data, schedules, and hyperparameters are provided in Appendix A. Dataset descriptions appear in Appendix D, and the ultra-long-video extension is detailed in Appendix C. The manuscript's code-release plan is stated in Appendix A.6.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, et al. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, et al. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

Zihan Cai, Yu Zhang, Bo Gao, et al. Pyramidkv: Dynamic kv cache compression based on pyramidal information funneling. *arXiv preprint arXiv:2406.02069*, 2024.

Guiming Chen, Zhi Li, Shi Wang, et al. Eagle 2.5: Boosting long-context post-training for frontier vision-language models. *arXiv preprint arXiv:2504.15271*, 2025a.

Zhe Chen et al. Internvl 2.5: Advancing open-source multimodal large language models. *arXiv preprint arXiv:2412.09616*, 2024.

Zhe Chen et al. Internvl3: Advanced multimodal large language model. *arXiv preprint*, 2025b.

Zhi Cheng, Jian Hu, Zhuang Liu, et al. V-star: Benchmarking video-llms on video spatio-temporal reasoning. *arXiv preprint arXiv:2503.11495*, 2025.

Chaoyou Fu, Yuhan Dai, Youtian Luo, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*, 2024.

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 5267–5275, 2017.

Kristen Grauman, Andrew Westbury, Eugene Byrne, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18995–19012, 2022.

Bo He, Heng Li, Young Kyun Jang, et al. Ma-lmm: Memory-augmented large multimodal model for long-term video understanding. *arXiv preprint arXiv:2404.05726*, 2024a.

Yuchen He, Feng Chen, Jian Liu, et al. Zipvl: Efficient large vision-language models with dynamic token sparsification and kv cache compression. *arXiv preprint arXiv:2410.08584*, 2024b.

Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. TVQA+: Spatio-temporal grounding for video question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

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 *ICML*, 2023.

Xinhao Li, Yi Wang, Jiashuo Yu, et al. Videochat-flash: Hierarchical compression for long-context video modeling. *arXiv preprint arXiv:2501.00574*, 2024a.

Yang Li, Hao Jiang, Chen Zhang, et al. Mminference: Accelerating pre-filling for long-context vlms via modality-aware permutation sparse attention. *arXiv preprint arXiv:2504.16083*, 2025.

Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, pp. 323–340. Springer, 2024b.

Yuhong Li, Yiran Huang, Biao Yang, et al. Snapkv: Llm knows what you are looking for before generation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 22947–22970, 2024c.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023.

Zhuang Liu, Lin Zhu, Baochen Shi, et al. Nvila: Efficient frontier visual language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 4122–4134, 2025.

En Lu, Zhen Jiang, Jian Liu, et al. Moba: Mixture of block attention for long-context llms. *arXiv preprint arXiv:2502.13189*, 2025.

Wentao Ma, Weiming Ren, Yiming Jia, Zhuofeng Li, Ping Nie, Ge Zhang, and Wenhu Chen. Videoeval-pro: Robust and realistic long video understanding evaluation. *arXiv preprint arXiv:2505.14640*, 2025.

Ming Qin, Xiang Liu, Zhi Liang, et al. Video-xl-2: Towards very long-video understanding through task-aware kv sparsification. *arXiv preprint arXiv:2506.19225*, 2025.

Qwen Team. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Yuchen Shu, Zhuang Liu, Peng Zhang, et al. Video-xl: Extra-long vision language model for hour-scale video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 26160–26169, 2025.

Peng Wang, Jinze Bai, et al. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Wei Wang, Zhi He, Wen Hong, et al. Lvbench: An extreme long video understanding benchmark. *arXiv preprint arXiv:2406.08035*, 2024a.

Xin Wang, Qing Si, Jian Wu, et al. Retake: Reducing temporal and knowledge redundancy for long video understanding. *arXiv preprint arXiv:2412.20504*, 2024b.

Yi Wang, Kunchang Li, Xinhao Li, et al. Internvideo2: Scaling foundation models for multimodal video understanding. In *European Conference on Computer Vision*, pp. 396–416. Springer, 2024c.

Hao Wu, Dong Li, Bo Chen, et al. Longvideobench: A benchmark for long-context interleaved video-language understanding. *arXiv preprint arXiv:2407.15754*, 2024.

Guangxuan Xiao, Yuandong facility Tian, Beidi Chen, et al. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.

Feng Xue, Yan Chen, Dong Li, et al. Longvila: Scaling long-context visual language models for long videos. *arXiv preprint arXiv:2408.10188*, 2024.

Jian Yuan, Hao Gao, Dong Dai, et al. Native sparse attention: Hardware-aligned and natively trainable sparse attention. *arXiv preprint arXiv:2502.11089*, 2025.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, et al. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 17283–17297, 2020.

Bo Zhang, Kai Li, Zhi Cheng, et al. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Llava-video: Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 34661–34710, 2023.

Jun Zhou, Yuchen Shu, Bing Zhao, et al. Mlvu: A comprehensive benchmark for multi-task long video understanding. *arXiv preprint arXiv:2406.04264*, 2024.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

A Implementation Details

A.1 Training Data

HTSC-AT is trained on two data sources: (1) a proprietary corpus of ≈420 GB long-form video content covering documentaries, lectures, movies, and live streams, annotated with coarse event-level labels; and (2) the publicly available LLaVA-Video-178K dataset, which provides diverse VideoQA and captioning pairs. The proprietary subset covers diverse domains including instructional, documentary, and surveillance content, providing broad coverage of real-world long-form video distributions. The two corpora are combined and used jointly across all three training phases.

A.2 Training Schedule

We follow the three-phase curriculum defined by the composite loss $\mathcal{L}$ in Eq. 18. Phase 1 emphasizes global narrative alignment, Phase 2 shifts emphasis to QA capability, and Phase 3 jointly optimizes all trainable modules end-to-end with the full composite loss to strengthen temporal robustness.

Training proceeds in three phases, each adjusting $(\lambda_1, \lambda_2, \lambda_3)$ in Eq. 18 to shift the optimization emphasis:

- **Phase I** (λ_1=0.5, λ_2=1.0, λ_3=0.1): Emphasizes global narrative alignment ($\mathcal{L}_{\text{cap}}$) to bootstrap meaningful compression representations before QA supervision dominates.
- **Phase II** (λ_1=1.0, λ_2=0.5, λ_3=0.3): Shifts emphasis to QA performance ($\mathcal{L}_{\text{QA}}$), aligning the adapter with the frozen language model's prediction head.
- **Phase III** (λ_1=1.0, λ_2=0.5, λ_3=0.5): Reinforces temporal consistency ($\mathcal{L}_{\text{temp}}$), improving robustness on multi-hop temporal reasoning tasks.

Throughout all phases, only the MSCD detector, TEBD module, and hybrid-attention adapter are updated; the LVLM backbone $\mathcal{D}_\phi$ remains frozen.

A.3 Hyperparameters

All trainable modules are optimized with AdamW (β_1=0.9, β_2=0.999, weight decay 10^{-2}). Learning rates are 5×10^{-4} for MSCD and TEBD and 1×10^{-4} for the hybrid-attention adapter, each with cosine decay. Training runs for 3 epochs per phase on 8×H800 GPUs with gradient checkpointing and bf16 mixed precision; the Qwen2.5-VL-7B backbone remains fully frozen throughout.

The three-phase weight schedule $(\lambda_1, \lambda_2, \lambda_3)$ follows Sec. 2.5: Phase I (0.5, 1.0, 0.1), Phase II (1.0, 0.5, 0.3), Phase III (1.0, 0.5, 0.5). The MSCD optical flow weight $\beta = 0.3$ (Eq. 2), minimum token count $N_{\min} = 8$, and maximum $N_t = 64$ (Eq. 3) are fixed throughout. TEBD boundary fusion weights are 0.6 (ActionFormer) / 0.4 (semantic mutation), consistent with Eq. 7. Mini-Batch K-Means batch size is 64 for all clustering stages.

A.4 ActionFormer Distillation

The ActionFormer backbone used in TEBD Branch 1 is obtained via knowledge distillation from a 12M-parameter teacher to a 3M-parameter student, retaining $\geq 92\%$ of the teacher's boundary detection accuracy on the ActivityNet-v1.3 validation set. Standard response-based distillation is applied with a temperature of $\tau = 4.0$ and a distillation loss weight of $\lambda = 0.7$.

A.5 Scene-Specific Duplication Threshold

The cross-event redundancy threshold τ_{dup} (Eq. 10) is set to 0.85 by default. For first-person video (egocentric datasets), a lower threshold of 0.75 is used to account for the higher visual repetition inherent in continuous point-of-view recordings; for live-stream content, a stricter threshold of 0.90 is adopted to avoid merging visually similar but temporally distinct commentary segments.

Table 6: Cross-domain scene adaptability of the TEBD segmentation strategy on Ego4D and TVQA+. HTSC-AT's LCUS advantage over fixed-window segmentation (Baseline B) remains consistent across first-person video and episodic drama domains, demonstrating the domain generality of event-boundary-guided compression.

Dataset	Segmentation Strategy	LCUS	Multi-hop QA	Compression
Ego4D	Ours (TEBD + Hier. Clustering)	**48.5**	**43.7**	**680**×
	Baseline B (Fixed 60s window)	40.2	36.5	450×
TVQA+	Ours (TEBD + Hier. Clustering)	**52.3**	**47.8**	**750**×
	Baseline B (Fixed 60s window)	44.6	41.2	500×

A.6 Reproducibility

Code, model weights, and evaluation scripts will be released upon acceptance. All reported results are averages over three independent runs; standard deviations were $<0.3\%$ across all metrics, indicating stable training dynamics.

B Experimental Analysis

B.1 Scene Adaptability across Video Domains

To verify that the TEBD segmentation strategy generalizes beyond the movie/lecture/live-stream distribution of LongVideoBench, we evaluate on two complementary out-of-distribution benchmarks: **Ego4D** (Grauman et al., 2022) (first-person long video) and **TVQA+** (Lei et al., 2020) (episodic drama). As shown in Table 6, our method maintains a consistent LCUS advantage of 8–10 points over the fixed-window baseline (Baseline B) across both datasets, while preserving higher compression ratios.

On Ego4D, the compression ratio of 680× exceeds the fixed-window baseline (450×) because TEBD's dual-branch boundary detector accurately identifies first-person action events (*e.g.*, "picking up an object", "opening a door"), avoiding the fragmentation of continuous actions that fixed-window segmentation incurs. On TVQA+, the 750× compression ratio is achieved by leveraging the denser semantic boundary structure of scripted drama, where dialogue-scene transitions are more pronounced.

Ego4D shows modestly lower absolute LCUS (48.5%) compared to TVQA+ (52.3%), which we attribute to frequent motion blur, view occlusion, and the absence of sharp visual scene boundaries in egocentric footage. Nonetheless, the relative improvement over the fixed-window baseline is consistent (8.3 vs. 7.7 points on LVB), confirming that TEBD's event-boundary-guided design is domain-agnostic.

B.2 Event Boundary Detection Accuracy

Table 7 reports event boundary detection accuracy, false-positive rate, and miss rate of the TEBD dual-branch detector across three video domains. On our proprietary long-video corpus, TEBD achieves a boundary detection accuracy of **91.67%**, with a false-positive rate $<5.6\%$ and miss rate $<7.4\%$, indicating reliable event segmentation in controlled conditions.

On public benchmarks, accuracy remains above 82% across all tested domains. Domain-specific degradations are attributable to known structural challenges: on Ego4D, frequent motion blur and view occlusion in first-person footage blur the visual boundary signal, while the absence of sharp scene-level transitions inflates the miss rate; on live-stream video, repetitive semantic patterns (*e.g.*, recurring audience reactions, similar consecutive interaction shots) lead to false positives; on LVB (movies/lectures), long static shots and gradual cross-scene fades cause occasional boundary misdetection. Despite these conditions, TEBD consistently maintains a precision–recall operating point suitable for downstream clustering, as evidenced by the LCUS results in Sec. 3.4.

Table 7: Event boundary detection performance of the TEBD dual-branch detector across video domains. Performance on the proprietary dataset is higher because it is drawn from the same distribution as the training annotation. Degradations on public benchmarks are attributable to known domain-specific challenges discussed in the text.

Video Domain	**Detection Acc. (%)**	**False-Positive Rate (%)**	**Miss Rate (%)**
Proprietary dataset (ours)	$\geq$91.7	$<$5.6	$<$7.4
LVB (movies / lectures) (Wu et al., 2024)	$\geq$85.0	$\leq$8.0	$\leq$10.0
Ego4D (first-person) (Grauman et al., 2022)	$\geq$82.0	$\leq$10.0	$\leq$12.0
Live-stream video	$\geq$83.0	$\leq$12.0	$\leq$10.0

C Ultra-Long Video Processing ($>$1 Hour)

For videos exceeding one hour, TEBD activates a three-tier hierarchical clustering regime (Sec. 2.3): global event-cluster grouping precedes intra-event coarse and fine clustering, enabling the model to manage videos with hundreds of events without exceeding the context window.

Each event E_e is first represented by the mean of its coarse centroids:

$$\mathbf{feat}_{E,\text{global}} = \frac{1}{K_e} \sum_{c \in E_e} \mathbf{feat}_c. \tag{20}$$

Mini-Batch K-Means is then applied with a global cluster count:

$$K' = \max(5,\, \min(\lfloor N_V/50 \rfloor\,,\, 30))\,, \tag{21}$$

where N_V is the total number of events, ensuring graceful scaling. Semantically related events are grouped into K' global clusters, yielding a three-tier hierarchy: *global cluster* $\rightarrow$ *event* $\rightarrow$ *intra-cluster token*. Core events (mean intra-event saliency $\overline{SS}_e \geq 0.7$) are allocated more clusters ($K_e \leq 6$), while non-core events are more aggressively compressed ($K_e \leq 4$), and both are annotated with three-tier timestamps (global cluster $\rightarrow$ event $\rightarrow$ intra-cluster), supporting full temporal traceability.

End-to-end inference for a 120-minute video requires $\leq$8 minutes on a single $8\times$H800 node, corresponding to a real-time factor better than $1/15$. This extension incurs negligible additional latency ($<$2% of total pipeline runtime) and maintains $N^{(2)} \ll N^{(1)}$ for videos up to 4 hours in our experiments.

On our internal ultra-long video evaluation set (videos in the 60–120 min range), the three-tier mode retains $>$95% of the LCUS performance of the two-tier mode applied to the same content truncated to 60 minutes, confirming that the global clustering stage successfully identifies and preserves the most semantically significant event chains.

C.1 Mahalanobis Outlier Selection Detail

Within each coarse cluster C_k, the Mahalanobis distance for token t is computed as:

$$\text{MD}_t = \sqrt{\left(\mathbf{feat}_t - \boldsymbol{\mu}_{C_k}\right)^\top \boldsymbol{\Sigma}_{C_k}^{-1} \left(\mathbf{feat}_t - \boldsymbol{\mu}_{C_k}\right)}, \tag{22}$$

where $\boldsymbol{\mu}_{C_k}$ and $\boldsymbol{\Sigma}_{C_k}$ denote the mean vector and covariance matrix of cluster C_k, respectively. The top-1 or top-2 tokens with the largest MD_t are retained as outliers, and the final summary set satisfies $K'_e = K_e + 1$–2, with $K'_e \leq 10$ enforced in practice.

D Datasets and Survey of Related Technical Directions

D.1 Dataset Descriptions

MLVU (Zhou et al., 2024) is a multi-task long-video understanding benchmark covering videos up to one hour, with tasks spanning needle-in-a-haystack retrieval, multi-detail summarization, and causal reasoning across non-contiguous segments.

VideoMME (Fu et al., 2024) provides short, medium, and long video subsets with diverse QA styles including open-ended, multiple-choice, and temporal localization questions, enabling fine-grained evaluation of length-dependent performance degradation.

LongVideoBench (LVB) (Wu et al., 2024) contains 5–60 min videos annotated with multi-hop, temporal-grounding, and summarization questions. Its explicit multi-fragment dependency structure makes it the primary benchmark for all ablation studies in this work.

LVBench (Wang et al., 2024a) targets extreme-length video understanding, with content exceeding one hour, stressing models' ability to maintain coherent reasoning over very long temporal horizons.

VideoEval-Pro (Ma et al., 2025) focuses on fine-grained perception and step-level reasoning, requiring precise temporal localization and compositional understanding beyond standard QA accuracy.

CharadesSTA (Gao et al., 2017) and **V-STaR** (Cheng et al., 2025) target temporal sentence grounding and step-level activity recognition, respectively, providing complementary evaluation of fine-grained temporal alignment. Ablation results on **Ego4D** (Grauman et al., 2022) and **TVQA+** (Lei et al., 2020) are also reported in Appendix B.

D.2 Detailed Survey of Technical Directions for Long Video Understanding

This section provides a comprehensive review of the four core technical directions for Long Video Understanding (LVU) summarized in Section 1, including representative models, core designs, and their limitations.

D.2.1 Visual Token Compression

Visual token compression reduces the number of tokens fed into the LLM through compression, fusion, downsampling, or key information filtering, directly alleviating memory and computational pressure (Qin et al., 2025; He et al., 2024a).

Llama-VID (Li et al., 2024b) compresses visual features into fixed query embeddings via cross-attention, achieving efficient encoding and significantly reducing the number of visual tokens while retaining basic visual information.

VideoChat-Flash (Li et al., 2024a) adopts the Token Merging (ToMe) method (Bolya et al., 2022) to hierarchically fuse similar visual tokens in long videos, and designs hierarchical compression strategies for different video segments to balance token compression and information retention.

InternVideo2 (Wang et al., 2024c) introduces spatiotemporal convolution to downsample visual tokens, effectively capturing temporal motion patterns while compressing the token count.

MA-LMM (He et al., 2024a) builds a memory bank to filter and store key visual information of long videos, discarding redundant tokens and retaining only core content semantically relevant to the video.

Despite their efficiency gains, token compression methods fundamentally cannot resolve the quadratic growth of self-attention computation with total token count (Qin et al., 2025). Extreme compression inevitably causes fine-grained spatiotemporal information loss, degrading performance on complex LVU tasks such as temporal grounding and fine-grained visual question answering (He et al., 2024b; Zhou et al., 2024).

D.2.2 Attention Sparsification

Inspired by research on long-context LLMs (Li et al., 2024c; Xiao et al., 2023; Zhang et al., 2023), attention sparsification techniques optimize computational costs by designing sparse attention patterns to avoid full attention computation among all visual tokens (Li et al., 2025; Yuan et al., 2025; Lu et al., 2025).

MMInference (Li et al., 2025) proposes modality-aware permutation sparse attention, designing predefined sparse patterns for different attention heads of MLLMs to adaptively reduce computation during the pre-filling stage, achieving significant efficiency improvements without noticeable performance loss.

ZipVL (He et al., 2024b) combines dynamic token sparsification with KV cache compression, enabling plug-and-play inference acceleration and effectively reducing memory overhead for long video processing.

Native Sparse Attention (Yuan et al., 2025) designs a hardware-aligned and natively trainable sparse attention mechanism that supports end-to-end training, making it more suitable for LVU tasks.

MoBA (Lu et al., 2025) proposes a Mixture of Block Attention mechanism, dividing long visual token sequences into multiple blocks and designing adaptive sparse attention patterns between blocks, balancing fine-grained intra-block modeling and inter-block long-distance dependency capture.

Despite these advancements, most sparse patterns are manually designed and lack adaptability to diverse LVU tasks (Qin et al., 2025; Li et al., 2025). Sparsifying inter-block attention also leads to the loss of cross-block semantic correlations, making it difficult to capture long-range temporal dependencies (Shu et al., 2025; Wang et al., 2024b).

D.2.3 Chunking and Streaming

Chunking processing divides the long visual token sequence into small chunks and processes them sequentially, fundamentally avoiding end-to-end processing of massive tokens (Shu et al., 2025; Wang et al., 2024b; Chen et al., 2025a).

Video-XL (Shu et al., 2025) first proposed a chunk-based streaming framework for hour-level LVU. By inserting special tokens into visual tokens, it generates sparse Key-Value (KV) pairs for each chunk and uses these sparse KV pairs to model inter-chunk dependencies, drastically reducing computational overhead.

ReTaKe (Wang et al., 2024b) retains full KV pairs for keyframes (filtered via attention scores) within each chunk and discards KV pairs for non-keyframes at a fixed ratio, balancing fine-grained information retention and memory efficiency.

LongViLA (Xue et al., 2024) migrates the hybrid architecture of long-context LLMs to the video domain, achieving efficient encoding of videos with thousands of frames through a combination of local and global attention.

Eagle 2.5 (Chen et al., 2025a) further improves chunking processing through long-context post-training, enhancing cross-modal alignment capability for chunked long video sequences.

However, existing chunking methods face a critical KV dependency explosion problem: the number of KV pairs required for current chunks to attend to historical chunks grows rapidly as processed chunk count increases, limiting scalability to extremely long videos (Qin et al., 2025; Wang et al., 2024b). Furthermore, fixed chunking partitions and inter-chunk interaction strategies lack adaptability to different LVU tasks (Qin et al., 2025).

D.2.4 KV Cache Optimization

The Key-Value (KV) cache stores KV pairs generated during the pre-filling stage to support iterative decoding, and is the primary source of memory overhead during MLLM inference (Li et al., 2024c; Cai et al., 2024). Recent research focuses on reducing memory footprints through compression, filtering, or dynamic adjustment of KV pairs (Liu et al., 2025; Li et al., 2024c; He et al., 2024b; Cai et al., 2024).

SnapKV (Li et al., 2024c) proposes a pre-generation KV filtering mechanism, allowing the model to identify and retain only text-query-relevant KV pairs before decoding, significantly reducing the KV cache size for cross-modal tasks.

PyramidKV (Cai et al., 2024) designs a dynamic KV cache compression method based on pyramidal information funneling, progressively refining KV pairs from bottom to top and retaining core semantic information only in higher-level KV pairs.

VideoChat-Flash (Li et al., 2024a) attempts to accelerate pre-filling and decoding stages by discarding KV pairs with low text-query relevance in specific model layers.

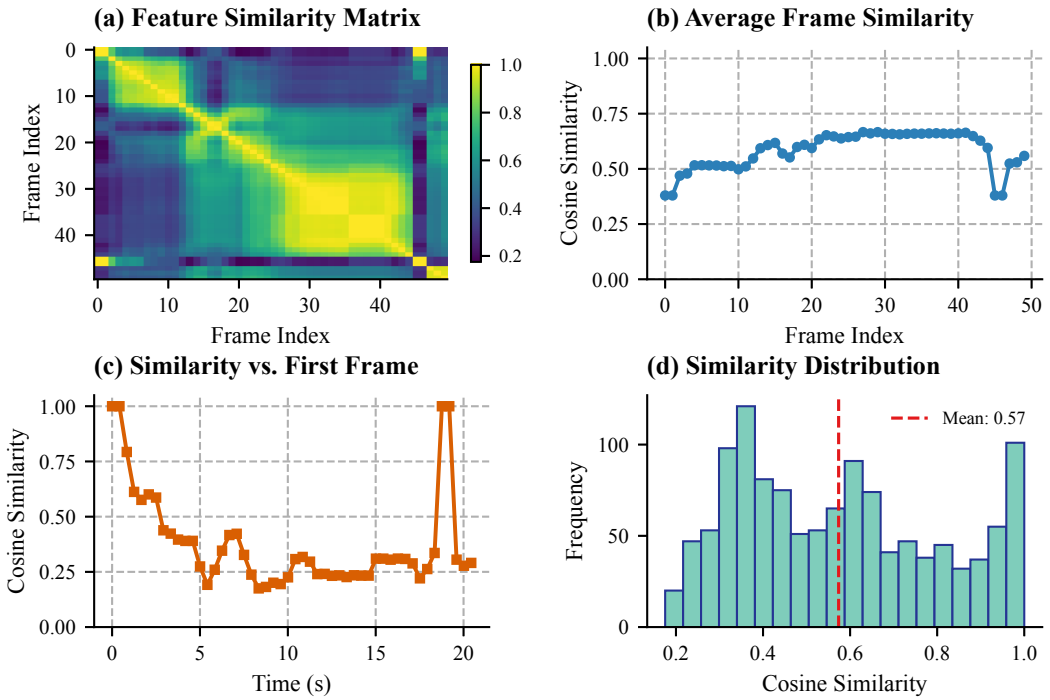

Figure 3: Redundancy analysis of CLIP-extracted video tokens on a representative LongVideoBench clip.*Top-left*: Inter-frame cosine similarity matrix; the pronounced diagonal band and warm off-diagonal regions indicate high similarity among both adjacent and non-adjacent frames. *Top-right*: Per-frame average similarity, stable in $[0.60,\ 0.75]$, confirming that redundancy is a global sequence property. *Bottom-left*: Per-frame similarity to the first frame; despite gradual decay, values remain above 0.6 throughout the clip, revealing persistent temporal correlation across the full timeline. *Bottom-right*: Pairwise similarity histogram (global mean 0.57), showing that most frame pairs are near-interchangeable under full attention.

Although these methods achieve meaningful memory optimization, most adopt single-level KV sparsification or compression strategies and lack task-aware dynamic KV management: they cannot adaptively adjust KV pair granularity (dense/sparse) based on the relevance between video chunks and text queries (Qin et al., 2025). This creates a dilemma—retaining dense KV pairs for the entire video causes excessive memory overhead, while over-compressing KV pairs leads to fine-grained information loss (Qin et al., 2025).

E Additional Token Redundancy Analysis

This superiority is grounded in the structural properties of video tokens (Figure 3): inter-frame cosine similarity exhibits pronounced temporal locality, per-frame redundancy is consistently high throughout the sequence, and the global pairwise similarity mean of **0.57** establishes token redundancy as a *global*, non-episodic property. Uniformly distributing a fixed token budget therefore wastes capacity on semantically redundant regions; the ΔS_t-driven allocation of MSCD (Eqs. 2–3) addresses this directly by concentrating tokens where semantic density is highest.

F Detailed Training Objectives

Task Alignment Loss ($\mathcal{L}_{\mathrm{QA}}$). Cross-entropy loss over VideoQA instances:

$$\mathcal{L}_{\mathrm{QA}} = -\sum_{t=1}^{T_{\mathrm{ans}}} \log P\Big(a_t \mid q,\ \mathcal{T}^{(2)},\ a_{<t}\Big)\,, \tag{23}$$

where a_t denotes the t-th ground-truth answer token and $P(\cdot)$ is the frozen LVLM language head's output distribution.

Global Narrative Preservation Loss ($\mathcal{L}_{\text{cap}}$). This captioning objective prevents over-compression of semantically coherent content:

$$\mathcal{L}_{\text{cap}} = -\sum_{t=1}^{T_{\text{cap}}} \log P\Big(c_t \mid \mathcal{T}^{(2)},\, c_{<t}\Big)\,, \tag{24}$$

where c_t is the t-th ground-truth caption token.

Temporal Consistency Regularization ($\mathcal{L}_{\text{temp}}$). The temporal regularizer encourages adjacent aligned-token representations to have greater cosine similarity than representations separated by one position:

$$\mathcal{L}_{\text{temp}} = \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} \max(0, \cos(\boldsymbol{\tau}_i, \boldsymbol{\tau}_{i+2}) - \cos(\boldsymbol{\tau}_i, \boldsymbol{\tau}_{i+1}) + \delta)\,, \tag{25}$$

where $\mathcal{I}$ contains valid triplet start indices in the compressed aligned token sequence, and $\delta = 0.1$ is the margin. For an active triplet, the loss encourages adjacent similarity to exceed the two-step similarity by this margin. This regularizes local temporal similarity; it does not by itself guarantee causal order preservation. The composite $\mathcal{L}$ (Eq. 18) is applied solely to the MSCD, TEBD, and adapter parameters; the LVLM backbone $\mathcal{D}_\phi$ remains frozen throughout. For details, see Appendix A.3.

Overall, the training design supports the three core goals of HTSC-AT: compression efficiency, cross-event reasoning, and long-context scalability. At the same time, the current composite objective $\mathcal{L}$ (Eq. 18) does not include explicit temporal localization supervision. This may limit temporal grounding performance, motivating a future extension that incorporates a dedicated grounding objective.

G Attention Score Counting Convention

The decoder uses token-level attention between connected events with a bounded anchor set. Define the token mask $M_{(i,u),(j,v)} = A_{ij}$ for all retained tokens u and v in events i and j. Thus every allowed event block is dense at the token level. For two separately evaluated dense and sparse branches in Eq. 17, the exact score-entry count per head per layer is

$$S_{\text{branches}} = \sum_i K_i'^2 + \sum_{i,j} A_{ij} K_i' K_j'. \tag{26}$$

Because Eq. 14 includes $A_{ii} = 1$, self blocks occur in both branches. Their repeated evaluation counts twice here; the number of distinct QK pairs in the union mask instead counts them once. These two conventions must not be interchanged.

For uniform K', let u be the number of consecutive event pairs whose endpoints are both non-anchors. With m distinct anchors,

$$|A| = 2mE - m^2 + (E - m) + 2u \le (2m + 3)E. \tag{27}$$

The first term counts ordered pairs incident to anchors, the second adds non-anchor self blocks, and the last adds remaining temporal-neighbor pairs. Table 4 uses the conservative budget $S_{\text{budget}} = (1 + C)EK'^2$ with $C = 2m + 3 = 15$, giving 0.12M for $m = 6$, $E = 300$, and $K' = 5$. It is not an exact graph-average degree or a measured count. Exact counts require the selected anchor indices and retained token counts. The Longformer and BigBird entries similarly use the stated nominal window/global/random budgets, without claiming exact deduplicated masks. Score counts exclude projections, feed-forward layers, and other operations, and cannot be interpreted as end-to-end FLOPs or runtime speedups.