SLASHVID: TRAINING-FREE EXTREME TOKEN COMPRESSION VIA SELECT-AND-AGGREGATE FOR VIDEO TEMPORAL GROUNDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Video Temporal Grounding (VTG) using Large Vision-Language Models (LVLMs) requires dense temporal sampling to preserve boundary-sensitive evidence, yet the resulting large number of visual tokens renders inference prohibitively expensive. At extreme compression ratios, pruning-only baselines can remove all tokens from some frames and discard subtle boundary cues. We propose **SlashVID**, a training-free framework that decomposes extreme token compression into two complementary paths: *aggregating* redundant tokens to preserve temporal context and *selecting* a small set of query-relevant tokens to sharpen boundary evidence. Specifically, SlashVID employs Adaptive Segment-wise Guidance to partition each video into segments and allocate a fixed budget across the two paths, Bidirectional Temporal Aggregation Clustering to consolidate the unselected context, and Dynamic Max-Min Importance Selection to add diverse, query-guided evidence. Experiments on Charades-STA and ActivityNet show that SlashVID retains 80% of full-token grounding performance at a 2% retention ratio while achieving 9.6$\times$ prefill speedup, improving relative performance retention over the strongest baseline by an absolute 4.55%.

1 INTRODUCTION

Large Vision-Language Models (LVLMs) (Comanici et al., 2025; Bai et al., 2025; Wang et al., 2025b) have demonstrated considerable potential in video understanding by encoding video frames as visual tokens and performing language-model-based reasoning over these tokens. However, this token-based interface introduces a critical computational bottleneck for Video Temporal Grounding (VTG), where the model is required to identify the start and end timestamps of the event specified by a natural language query (Gao et al., 2017; Guo et al., 2025). Unlike general video understanding (Xu et al., 2016; Wang et al., 2025a), for which semantic questions can often be answered using only a sparsely sampled set of frames, VTG requires denser temporal observations to preserve the temporal evolution and boundary evidence of the target moment. Consequently, the number of visual tokens scales rapidly with video length, rendering inference prohibitively expensive.

Existing visual token compression methods are primarily developed for image understanding (Zhang et al., 2025) or VideoQA tasks (Kang et al., 2026), where retaining globally salient objects or representative keyframes is often sufficient for coarse-grained semantic reasoning. However, this objective is fundamentally misaligned with the requirements of VTG, which relies on reasoning over a temporally continuous sequence. As illustrated in Fig. 1(a), when compression eliminates all tokens from certain frames, it introduces temporal holes that force the language model to hallucinate event boundaries from discontinuous evidence (Shen et al., 2025). Moreover, VTG is inherently query-dependent. Visually salient regions are not necessarily relevant to the queried event, whereas subtle object interactions or brief scene transitions may provide critical boundary cues that are discarded by generic saliency-based pruning methods (Yang et al., 2025). These issues are further exacerbated at extreme compression ratios. Global saliency-based pruning allows visually prominent background scenes to monopolize the video-level budget, creating temporal voids and starving subtle moments that actually contain the queried action. Conversely, uniform frame-level compression fails to adapt to varying visual complexity and redundancy across scenes. Furthermore, generic selection mechanisms lack query sensitivity, frequently wasting scarce tokens on redundant salient objects while

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.

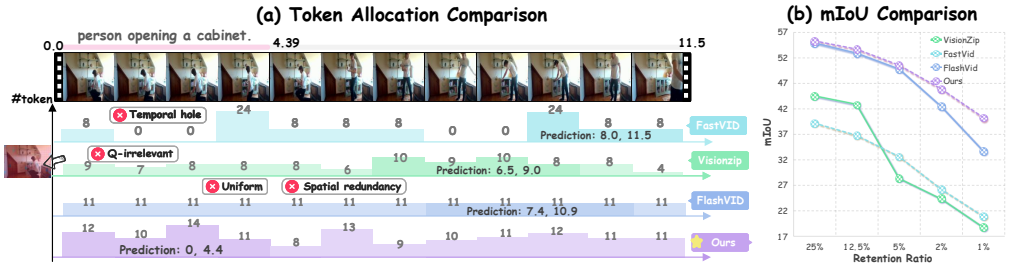


Figure 1: Motivation and performance comparison. (a) Token allocation comparison under 2% retention. (b) mIoU on Charades-STA under different visual token retention ratios.

failing to concentrate fine-grained visual evidence on query-relevant event boundaries (Fan et al., 2026).

Fig. 1(b) reports mIoU on Charades-STA (Sigurdsson et al., 2016) across retention ratios, showing that existing methods degrade sharply when the retention ratio falls below 5%, with the gap widening further as compression becomes more aggressive. These observations raise a central question: *How can visual tokens be compressed to extremely low retention ratios while preserving temporal continuity, prioritizing query-relevant cues, and minimizing redundancy?*

To this end, we propose **SlashVID**, **S**egment-**L**evel **A**daptive **S**election with **H**ierarchical aggregation for **VID**eo temporal grounding, a training-free framework applicable to off-the-shelf LVLMs. To address token redundancy and preserve query-relevant information under extremely constrained token budgets, SlashVID decomposes token compression across three structured levels. First, Adaptive Segment-wise Guidance (ASG) partitions the video into temporal segments and assigns a length-proportional total budget to each segment, preventing globally salient background scenes from consuming the entire video budget. ASG then adaptively modulates the branch budget split between selection and aggregation according to segment visual complexity. Second, the aggregation branch, powered by Bidirectional Temporal Aggregation Clustering (Bi-TAC), consolidates the remaining spatiotemporally redundant tokens into compact representatives via bidirectional merging and temporally uniform density-peak clustering. This branch preserves contextual and transitional evidence that pruning-only approaches would otherwise discard, and is the main source of robustness under extreme compression. Third, within the selection budget, Dynamic Max-Min Importance Selection (DMM-IS) adds diverse, query-relevant tokens around the target moment and its boundaries, while enforcing at least one token per frame to preserve temporal continuity. Thus, query guidance complements rather than replaces temporal aggregation.

Our contributions are summarized as follows:

- We propose SlashVID, a training-free select-and-aggregate framework for VTG that preserves temporal context under extreme token budgets.
- We introduce three complementary modules: ASG for segment-wise budget reservation and complexity-adaptive branch allocation, Bi-TAC for the primary redundancy-aware temporal aggregation path, and DMM-IS for complementary query-guided diverse selection.
- SlashVID preserves frame-level temporal coverage at the LLM input under ultra-low retention, retaining 80% of full-token grounding performance at a 2% retention ratio with a 9.6× prefill speedup.

2 Related Work

Video temporal grounding with LVLMs. Early VTG methods develop task-specific architectures trained with temporal annotations. UniVTG (Lin et al., 2023) unifies diverse grounding labels into a single formulation and enables temporal grounding pretraining from large-scale heterogeneous supervision. More recently, VTimeLLM (Huang et al., 2024) introduces boundary-aware multi-stage training to teach Video-LLMs fine-grained timestamp understanding. TimeLens (Zhang et al., 2026) systematically investigates data quality and algorithmic design for VTG-oriented MLLMs, achieving state-of-the-art through reinforcement learning with verifiable rewards. UniTime (Li et al., 2026b) interleaves timestamp tokens with visual tokens and uses adaptive frame scaling to handle

both short and long videos. While effective, all these methods require supervised fine-tuning with temporal annotations, limiting scalability.

A parallel line of work explores training-free VTG. TFVTG (Zheng et al., 2024) leverages frozen VLMs to score enumerated proposals with dynamic-static scoring functions, achieving competitive zero-shot grounding. Luo et al. (Luo et al., 2024) adapt frozen VLMs via conditional feature refinement and bottom-up proposal generation for zero-shot moment retrieval. HiTeA (Xu et al., 2026) introduces hierarchical temporal decomposition to align queries at event, scene, and action granularities. Whether trained or training-free, these VTG methods all operate on full visual token sequences, facing a fundamental efficiency bottleneck as video length grows. Visual token compression offers a principled solution to this problem.

Visual token compression for LVLMs. Token compression for vision Transformers (Dosovitskiy et al., 2020) originates from image understanding. ToMe (Bolya et al., 2023) merges similar tokens in ViTs based on bipartite matching, achieving $2\times$ throughput with minimal accuracy loss. FastV (Chen et al., 2024) identifies text-relevant visual tokens via early-layer attention and prunes the rest in deeper layers. PyramidDrop (Xing et al., 2024) progressively reduces tokens across LLM layers in a pyramid fashion. These image-centric methods exploit spatial redundancy but do not consider temporal structure. For video understanding, VisionZip (Yang et al., 2025) selects informative tokens via CLS attention statistics and generalizes to video QA, but treats frames independently without temporal modeling. FastVID (Shen et al., 2025) improves over uniform pruning with dynamic cross-frame density estimation, but remains query-agnostic and may discard boundary-critical tokens. FlashVID (Fan et al., 2026) proposes tree-based spatiotemporal token merging with attention-diversity selection. However, as a task-agnostic compressor, it does not condition budget allocation on the VTG query or explicitly optimize frame-level coverage for boundary localization. SemVID (Li et al., 2026a) is the only VTG-specific compression work, preserving semantic evidence chains via query-aware allocation over object, motion, and context tokens. However, its semantic-unit constraint prevents compression below 2%, and unselected tokens are discarded.

3 PRELIMINARIES

Video temporal grounding with LVLMs. Given an input video V and a language query q, video temporal grounding aims to predict a temporal segment (t_s, t_e) that semantically corresponds to the query. When an LVLM is used for this task, the video is first sampled into T frames and encoded into visual tokens. Let each frame contain M visual tokens after the vision encoder and projector. The total number of visual tokens is therefore given by:

$$N_v = TM. \tag{1}$$

Together with N_t textual tokens, the large language model (LLM) receives a sequence of length $N = N_t + N_v$. Since VTG usually requires dense temporal sampling to preserve timestamp resolution, T can be much larger than in VideoQA settings. Consequently, N_v often dominates the input length.

Efficiency bottleneck of visual tokens. The major inference cost lies in the language model backbone. For a Transformer layer with hidden size d and feed-forward network (FFN) intermediate size m, the computational cost can be roughly decomposed into linear projections, self-attention, and FFN operations. Across L layers, the cost scales as:

$$\mathcal{O}\left(L\left(Nd^2 + N^2d + Ndm\right)\right), \tag{2}$$

where the quadratic term comes from self-attention and the linear terms come from projections and FFNs. Although this expression hides architecture-dependent constants, it makes the key dependency clear: increasing the number of visual tokens enlarges the entire LLM sequence and affects both attention and FFN computation. In long videos, where $N_v \gg N_t$, reducing N_v is therefore one of the most direct ways to improve inference efficiency.

Minimum retention for frame-level temporal connectivity. Unlike VideoQA, VTG requires the compressed representation to remain temporally traceable. If some frames lose all visual tokens, the LLM must infer their contents from neighboring frames, which can introduce temporal holes and boundary misalignment. A natural lower bound for maintaining frame-level connectivity is to keep at least one token per frame. Under the original token count $N_v = TM$, this corresponds to the

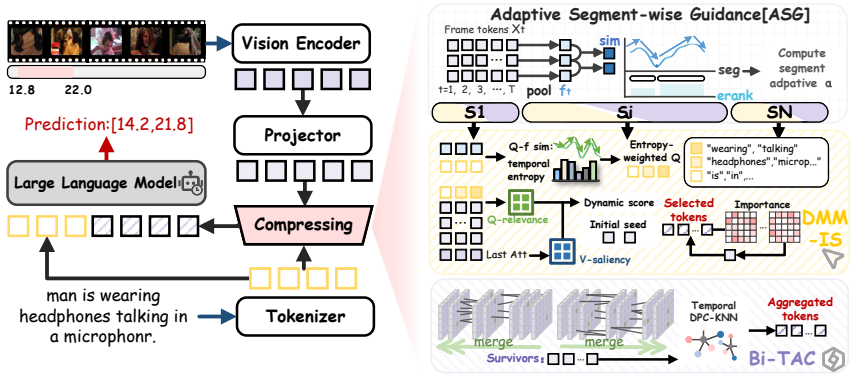


Figure 2: The pipeline of SlashVID. ASG partitions the video into temporal segments and allocates each segment's token budget between two paths: DMM-IS selects query-relevant, diverse tokens; Bi-TAC aggregates the rest via bidirectional merging and density-peak clustering.

minimum vision-side retention ratio:

$$\rho_{\min} = \frac{T}{TM} = \frac{1}{M}. \tag{3}$$

This bound is independent of video length but depends on per-frame token count M. For high-resolution videos, M is large and $\rho_{\min}$ can be very small; however, approaching this regime is challenging because each frame is represented by only a few tokens. In this work, we evaluate settings with $\rho_v \geq \rho_{\min}$; below this bound, exact target retention and frame-level connectivity cannot be simultaneously satisfied. If an LLM-side pruning stage keeps a ratio ρ_l of the already compressed visual tokens, the post-pruning token ratio in deeper layers becomes:

$$\rho_{\text{deep}} = \rho_v \rho_l, \tag{4}$$

where ρ_v is the vision-side retention ratio. Therefore, extreme compression for VTG must not only reduce the total number of tokens, but also carefully decide where the remaining tokens should be placed along the temporal axis.

4 Method

4.1 Overview

As shown in Fig. 2, we propose **SlashVID**, **S**egment-**L**evel **A**daptive **S**election with **H**ierarchical aggregation for **VID**eo temporal grounding. Given video tokens $\mathbf{X} \in \mathbb{R}^{T \times M \times C}$ and a target token budget before the LLM, SlashVID produces a compact sequence $\widetilde{\mathbf{X}}$ while maintaining frame-level temporal connectivity. The key idea is select-and-aggregate: aggregated tokens retain contextual cues from the otherwise discarded majority, while selected tokens add query-relevant fine-grained evidence around the target boundaries. SlashVID realizes this idea through Adaptive Segment-wise Guidance (ASG) for segment-wise budgeting, Dynamic Max-Min Importance Selection (DMM-IS) for diverse query-aware selection, and Bidirectional Temporal Aggregation Clustering (Bi-TAC) for temporal aggregation (see Appendix A, Algorithms 1–3 for complete algorithmic procedures).

4.2 Adaptive Segment-wise Guidance

Adaptive temporal segmentation. Compressing a whole video globally can cause visually salient segments to dominate the token budget, while weak but temporally important segments are underrepresented. To preserve the coarse temporal structure, SlashVID first divides the video into dynamically determined segments. We compute the frame-level representation by average pooling tokens within each frame, and measure adjacent frame similarity via cosine similarity:

$$\mathbf{f}_t = \frac{1}{M} \sum_{i=1}^{M} \mathbf{x}_{t,i}, \qquad s_t = \cos(\mathbf{f}_t, \mathbf{f}_{t+1}). \tag{5}$$

A temporal boundary is inserted when s_t falls below threshold τ_s, indicating a possible scene transition. To avoid degenerate cases where the whole video is treated as a single segment, we enforce a minimum number of segments by adding boundaries at low-similarity transitions when needed.

Segment-wise budget allocation. Given a retention ratio ρ_v, the per-frame budget is $B = \max(1, \lfloor \rho_v M \rfloor)$ and the segment total budget is $B_k = BT_k$. Crucially, this total segment budget is strictly length-proportional and query-agnostic; its purpose is to reserve temporal budget, guaranteeing that visually prominent background segments cannot exhaust the video-level budget and starve quieter temporal intervals where the target event occurs.

Within each segment, ASG modulates the branch budget allocation between fine-grained selection and contextual aggregation:

$$b_{\text{sel}} = \max(1, \lfloor \alpha_k B \rfloor), \quad b_{\text{agg}} = B - b_{\text{sel}}, \tag{6}$$

with selection budget $B_{\text{sel}} = b_{\text{sel}} T_k$ and aggregation budget $B_{\text{agg}} = b_{\text{agg}} T_k$; when B=1, all budget goes to DMM-IS for coverage. The allocation coefficient α_k is governed by segment visual complexity via effective rank, rather than textual queries. For frame t, we form $\mathbf{A}_t \in \mathbb{R}^{K_a \times C}$ by stacking the top-K_a L2-normalized tokens ranked by saliency $a_{t,i}$, and let $\{\sigma_{t,r}\}_{r=1}^{R_t}$ be its singular values with $p_{t,r} = \sigma_{t,r} / \sum_{r'} \sigma_{t,r'}$. We compute the segment-average effective rank:

$$\bar{e}_k = \frac{1}{T_k} \sum_{t \in \mathcal{S}_k} \frac{\exp(-\sum_{r=1}^{R_t} p_{t,r} \log p_{t,r})}{R_t}. \tag{7}$$

We map it to $\alpha_k = \alpha_{\min} + (\alpha_{\max} - \alpha_{\min}) \bar{e}_k$, where $\bar{e}_k \in [0, 1]$ and $[\alpha_{\min}, \alpha_{\max}] = [0.2, 0.8]$. Visually diverse segments with higher effective rank allocate more tokens to selection (B_{sel}), while redundant segments allocate more to aggregation (B_{agg}). Because $b_{\text{sel}} \geq 1$ and $b_{\text{sel}} + b_{\text{agg}} = B$, DMM-IS can seed at least one token per frame while the total segment budget remains exactly B_k. Query-adaptive token concentration is delegated to DMM-IS (Section 4.3), which prioritizes query-discriminative boundary cues within the allocated selection budget B_{sel}.

4.3 Dynamic Max-Min Importance Selection

Under extreme compression, selecting tokens by top-k visual attention is insufficient: it may keep redundant tokens from the same salient region and ignore query-relevant but less visually dominant evidence. DMM-IS addresses this by combining a dynamic importance score with diversity-aware farthest-point selection (FPS).

Query-adaptive importance. We first compute a query-adaptive importance score for each visual token. Let $\mathbf{q}_j$ denote the j-th text token embedding and $\mathbf{f}_t$ the frame-level mean feature. We compute the text-to-frame similarity matrix $\mathbf{S} \in \mathbb{R}^{Q \times T}$ as $S_{j,t} = \cos(\mathbf{q}_j, \mathbf{f}_t)$ and derive a per-text-token temporal distribution via $P_{j,t} = \text{softmax}(S_{j,\cdot}/\tau)_t$, where τ is a temperature. The temporal entropy of the j-th text token is given by:

$$H_j = -\sum_{t=1}^{T} P_{j,t} \log P_{j,t}. \tag{8}$$

Text tokens with lower entropy respond to fewer frames and thus correspond to temporally discriminative concepts (e.g., specific actions). We convert entropy to weight:

$$w_j = \frac{(1 - H_j / \log T)^\gamma}{\sum_{j'} (1 - H_{j'} / \log T)^\gamma}, \tag{9}$$

where $\gamma = 0.7$ controls the sharpness. The weighted query vector is $\mathbf{q}^* = \sum_j w_j \hat{\mathbf{q}}_j$ (with $\hat{\mathbf{q}}_j$ denoting L2-normalized text embeddings), and the per-token query relevance is $r_{t,i} = \max(\cos(\mathbf{x}_{t,i}, \mathbf{q}^*), 0)$. Combining this relevance with static saliency $a_{t,i}$ (last-layer vision self-attention mass, frame-normalized), with λ_q controlling the strength of query guidance, we compute:

$$I_{t,i} = a_{t,i} \cdot (1 + \lambda_q \cdot r_{t,i}). \tag{10}$$

Max-min diversity selection. Given the importance scores, we select tokens via greedy FPS on a candidate set. We first pre-filter each frame to retain its top-k tokens by $I_{t,i}$, forming a candidate

set $\mathcal{C}$. For each frame, an initial seed is chosen as $\arg\max_{i\in\mathcal{C}_t} I_{t,i}\cdot d_i^{\beta}$, where $d_i = 1/(\text{density}_i + 1)$ measures global diversity (density counts candidates with cosine similarity > 0.8) and β is a weighting exponent. Remaining tokens are then selected iteratively according to:

$$i^* = \arg\max_{i\in\mathcal{C}\setminus\mathcal{S}} \min_{j\in\mathcal{S}} D(i,j), \tag{11}$$

where the weighted distance is defined as:

$$D(i,j) = (1-\cos(\mathbf{x}_i,\mathbf{x}_j))\cdot\tilde{I}_i\cdot(1+\lambda_q\cdot r_i), \tag{12}$$

and $\tilde{I}_i$ is the normalized importance weight; r_i biases diversity toward query-related regions. This process continues until the selection budget B_{sel} is reached. We reserve at least one DMM-IS token per frame by replacing the last FPS-selected token with the highest-importance candidate from any uncovered frame, so for $\rho_v \geq 1/M$ the final sequence keeps frame-level coverage, while Bi-TAC only adds contextual aggregates.

4.4 Bidirectional Temporal Aggregation Clustering

DMM-IS retains a small set of informative tokens, yet the unselected majority still carries the contextual and transitional cues that are valuable for boundary localization. Bi-TAC therefore treats aggregation as the main context-preserving path.

Bidirectional token merging. For each unselected token $\mathbf{x}_{t,i}$, we search both forward and backward within a temporal window of radius W for the most similar anchor $\mathbf{x}_{t',i'}$ (from adjacent frames) satisfying $\cos(\mathbf{x}_{t,i},\mathbf{x}_{t',i'}) > \tau_m$, and merge by size-weighted averaging. The search radius is controlled by H through $W = \lceil T_k/(2H)\rceil$ for a segment of T_k frames. Operationally, each source uses one direction selected by the segment-midpoint router after both directional candidates are computed. Unlike one-directional merging that biases information toward earlier or later frames, this bidirectional strategy distributes active directions across the segment, reducing temporal bias near event boundaries.

Temporally uniform density-peak clustering. After merging, the surviving tokens may still exceed the aggregation budget B_{agg}. We select B_{agg} cluster centers via density-peak clustering with a temporal uniformity penalty. For each survivor token i, we estimate local density ρ_i from its k-nearest neighbors and compute separation δ_i as the distance to the nearest token with higher density:

$$\rho_i = \exp\Big(-\frac{1}{k}\sum_{j\in\text{kNN}(i)} c_{ij}^2\Big), \quad \delta_i = \min_{j:\rho_j>\rho_i} c_{ij}, \tag{13}$$

where $c_{ij} = 1-\cos(\mathbf{x}_i,\mathbf{x}_j)$ is the cosine distance, k=7, and the highest-density token takes the maximum survivor distance as δ_i. The center selection score incorporates a temporal uniformity penalty λ_u that discourages selecting multiple centers from the same frame:

$$score_i = \rho_i\cdot\delta_i\cdot(1-\lambda_u\cdot c_{t(i)}), \tag{14}$$

where $c_{t(i)}$ is the number of centers already selected from frame $t(i)$, normalized by the current total center count. We greedily select the top-scoring tokens as centers until the budget is reached; hard frame-level coverage is already provided by DMM-IS. Finally, each remaining survivor is assigned to its nearest center and merged by size-weighted averaging, producing temporally balanced aggregated tokens.

5 Experiments

5.1 Experiment Settings

Benchmarks. We evaluate SlashVID on two VTG benchmarks, Charades-STA (Sigurdsson et al., 2016) and ActivityNet-Captions (Caba Heilbron et al., 2015), covering short fine-grained actions and longer diverse events, respectively. We additionally test on two VideoQA benchmarks, LongVideoBench (Wu et al., 2024) and VideoMME (Fu et al., 2025), to verify generalization beyond VTG. For VTG, we report Recall@1 at IoU thresholds 0.5 and 0.7, as well as mean IoU

(mIoU). The relative mIoU percentage with respect to the full-token baseline quantifies retained grounding performance after compression.

Models and baselines. We conduct experiments with Qwen3-VL backbones, including 4B and 8B variants, to verify the generality of SlashVID across model scales. The full-token inference result is used as the 100% baseline. We compare SlashVID with representative training-free visual token compression methods, including VisionZip (Yang et al., 2025), FastVID (Shen et al., 2025), FlashVID (Fan et al., 2026), and SemVID (Li et al., 2026a). SemVID is a concurrent VTG-oriented compression method that allocates tokens based on semantic evidence chains; we evaluate it across its operational retention range (25%–2%), while noting that its semantic-unit formulation inherently requires a minimum token budget that precludes compression below 2%.

Implementation details. SlashVID is applied in a training-free and plug-and-play manner without updating any model parameters. We evaluate retention ratios of 25%, 5%, 2%, and 1%. We set τ_s=0.9 with at least 2 segments, K_a=$\min(64, M)$, λ_q=0.9, query temperature τ=0.9, β=0.7, τ_m=0.8, λ_u=1.0, and H=4; α is adaptive per segment (see Appendix A.3 for details). All methods use the same frame sampling, prompt, timestamp parser, greedy decoding with a maximum of 256 new tokens, and no task-specific fine-tuning.

Token budget definition. The reported retention ratio r denotes the target visual-token ratio at the LLM input. Let N_v be the original number of visual tokens and N_b the number entering the LLM. For before-LLM-only methods, N_b is obtained by applying the target ratio to each video's actual visual-token count. Hybrid methods use the same before-LLM target r and then prune the input sequence at layer K to $N_a = \rho_l N_b$ tokens for the remaining layers. Since token counts are discrete, the implementation applies integer floor rounding (together with the frame-coverage constraint where required), so the realized input ratio can be slightly below the nominal r. We do not expand the LLM input to compensate for subsequent inner-LLM pruning. We follow each baseline's original paradigm: VisionZip, FastVID, and SemVID are before-LLM-only. FlashVID and SlashVID are hybrid. SlashVID uses FlashVID's inner-LLM pruning setting (K=24, ρ_l=0.4), fixed across all datasets, backbones, and ratios. Appendix C.6 reports SlashVID without inner-LLM pruning to isolate the before-LLM compressor.

5.2 Main Results

Table 1 reports results on Charades-STA and ActivityNet for Qwen3-VL-4B and Qwen3-VL-8B. SlashVID retains 79.30% of full-token mIoU at a 2% retention ratio on Charades-STA (4B), and, at 1% (where SemVID cannot operate), outperforms FlashVID by 6.50 mIoU. On ActivityNet at 1%, SlashVID reaches 32.13 mIoU. The advantage grows as compression becomes more extreme: at 2% SlashVID surpasses FlashVID by 3.93 and SemVID by 2.63, while at moderate ratios (5%–25%) all methods perform reasonably with SlashVID maintaining a consistent lead. Qwen3-VL-8B exhibits the same trends, confirming that the gains are independent of model scale. The performance gap at extreme ratios reveals the failure modes of existing methods. VisionZip and FastVID degrade sharply below 5% because frame-independent pruning produces widespread temporal holes (99% of samples at 1%, Table 9 in Appendix B.6). FlashVID avoids holes through spatiotemporal aggregation but does not condition compression on the grounding query. SemVID adds query awareness but discards unselected tokens, losing transitional context that is critical for boundary localization. SlashVID makes aggregation the primary route for retaining this context and uses query-guided selection as a complementary refinement, while explicit frame coverage protects temporal traceability.

5.3 Generalization Across Tasks and Backbones

Although SlashVID is designed for VTG, its compression mechanism is task-agnostic at the vision-token level. To verify that the VTG-oriented design does not sacrifice general video understanding ability, we evaluate SlashVID on two VideoQA benchmarks with LLaVa-Video-7B (Zhang et al., 2024) and LLaVa-OneVision-7B (Li et al., 2025) under 10% token retention (Table 2). SlashVID achieves the highest or comparable scores across both models and benchmarks, retaining up to 99.3% of full-token accuracy. This confirms that preserving temporal structure and diversity benefits not only boundary-sensitive grounding but also general semantic comprehension. To further verify backbone generality, Table 3 reports results with Qwen3.5-4B on Charades-STA, showing that

Table 1: VTG results with Qwen3-VL at target LLM-input retention ratios r on Charades-STA and ActivityNet. Percentages for locally reproduced rows show relative mIoU against the corresponding full-token baseline. Italic SemVID entries are quoted from the original paper; their italic percentages use the original paper's evaluation baseline.

Ratio	Method	Qwen3-VL-4B								Qwen3-VL-8B							
		Charades-STA				ActivityNet				Charades-STA				ActivityNet			
		R@0.5	R@0.7	mIoU	(%)	R@0.5	R@0.7	mIoU	(%)	R@0.5	R@0.7	mIoU	(%)	R@0.5	R@0.7	mIoU	(%)
100%	Qwen3-VL	67.98	42.77	57.79	100.00	42.05	26.46	42.93	100.00	68.76	44.52	58.60	100.00	40.06	25.66	41.76	100.00
25%	VisionZip	47.37	21.59	44.41	76.85	32.71	20.27	35.54	82.79	57.80	29.92	51.08	87.17	36.90	23.13	38.98	93.34
	FastVID	42.28	20.89	39.11	67.68	32.41	19.07	35.22	82.04	41.21	21.29	38.71	66.06	27.80	16.60	31.63	75.74
	FlashVID	63.60	37.90	54.82	94.86	39.04	24.11	39.43	91.85	66.10	41.96	56.67	96.71	37.29	23.86	39.63	94.90
	SemVID	*61.08*	*33.17*	*52.31*	*93.30*	*35.68*	*22.51*	*39.06*	*96.90*	*60.11*	*34.55*	*52.66*	*93.00*	*33.49*	*21.50*	*37.54*	*96.00*
	SlashVID	**64.32**	**39.06**	**55.25**	**95.57**	**40.27**	**25.27**	**41.62**	**96.95**	**66.80**	**42.12**	**56.74**	**96.83**	**38.09**	**24.52**	**40.68**	**97.41**
5%	VisionZip	24.70	9.92	28.39	49.13	23.02	12.80	27.29	63.57	40.62	17.74	38.97	66.50	26.24	15.24	30.14	72.17
	FastVID	31.50	14.68	32.58	56.38	27.78	15.84	31.24	72.77	32.50	14.89	32.85	56.06	21.81	12.53	26.52	63.51
	FlashVID	56.69	30.10	49.76	86.10	35.20	20.85	37.32	86.93	53.82	29.38	47.73	81.45	32.47	19.18	35.60	85.25
	SemVID	56.32	29.76	49.40	85.48	34.68	**21.69**	37.24	86.75	55.62	30.75	49.24	84.03	30.53	18.69	33.90	81.17
	SlashVID	**57.55**	**31.45**	**50.49**	**87.37**	**36.23**	21.43	**38.27**	**89.15**	**58.49**	**32.17**	**50.91**	**86.88**	**33.96**	**20.15**	**36.34**	**87.02**
2%	VisionZip	18.49	6.80	24.37	42.17	18.10	9.72	22.93	53.41	23.39	9.17	27.21	46.43	19.35	10.65	24.31	58.22
	FastVID	24.03	9.76	26.18	45.30	24.25	13.04	27.86	64.90	23.17	9.54	25.57	43.63	16.29	8.93	21.03	50.36
	FlashVID	45.16	22.80	41.90	72.50	30.34	17.53	32.70	76.17	35.35	16.83	35.19	60.05	28.79	16.52	32.22	77.16
	SemVID	46.64	22.77	43.20	74.75	30.54	18.22	33.53	78.10	44.97	22.15	41.75	71.25	25.62	15.38	29.54	70.74
	SlashVID	**51.21**	**25.21**	**45.83**	**79.30**	**32.19**	**18.30**	**34.93**	**81.37**	**48.44**	**23.49**	**43.61**	**74.42**	**29.53**	**17.25**	**32.89**	**78.76**
1%	VisionZip	11.85	3.76	18.78	32.50	15.69	8.21	20.86	48.59	14.87	5.94	21.46	36.62	15.33	8.31	20.28	48.56
	FastVID	18.68	7.04	20.86	36.10	23.98	12.82	27.46	63.96	14.62	6.05	18.38	31.37	15.44	8.41	20.17	48.30
	FlashVID	33.06	15.83	33.71	58.33	26.31	14.15	29.70	69.18	27.72	11.83	28.68	48.94	23.88	13.20	27.79	66.55
	SlashVID	**42.90**	**20.22**	**40.21**	**69.58**	**29.16**	**15.93**	**32.13**	**74.84**	**37.15**	**16.99**	**36.12**	**61.64**	**25.66**	**14.29**	**29.52**	**70.69**

Table 2: VQA results on LongVideoBench (LVB) and VideoMME with LLaVa-OneVision-7B and LLaVa-Video-7B under 10% visual token retention.

Method	LVB		VideoMME				
	Value	(%)	Short	Medium	Long	Overall	(%)
LLaVa-Video-7B	59.5	100.0	77.0	62.1	53.3	64.2	100.0
FastV	53.6	90.1	64.3	53.8	49.2	55.8	86.9
VisionZip	54.5	91.6	69.4	57.9	51.2	59.5	92.7
FastVID	56.9	95.6	71.8	57.3	50.2	59.8	93.1
FlashVID	57.7	97.0	72.2	59.1	51.2	60.9	94.9
SlashVID	**58.3**	**98.0**	**72.8**	**60.0**	**51.1**	**61.3**	**95.5**
LLaVa-OneVision-7B	56.6	100.0	69.9	56.7	48.9	58.5	100.0
FastV	52.4	92.6	60.9	52.2	44.9	52.7	90.1
VisionZip	49.4	87.3	60.3	52.9	46.7	53.3	91.1
PruneVID	54.0	95.4	65.9	52.8	45.6	54.7	93.5
FastVID	55.7	98.4	68.1	55.7	47.8	57.2	97.8
FlashVID	56.5	99.8	67.3	**57.1**	49.0	57.8	98.8
SlashVID	**56.5**	**99.8**	**69.7**	56.3	**49.0**	**58.1**	**99.3**

Table 3: VTG results with Qwen3.5-4B under $r\%$ visual token retention on Charades-STA.

Ratio	Method	R@0.5	R@0.7	mIoU
100%	Baseline	68.15	44.27	58.80
5%	VisionZip	46.67	24.57	44.50
	FastVID	46.75	24.81	44.43
	FlashVID	52.31	30.38	48.02
	SlashVID	**53.71**	**31.94**	**49.02**
2%	VisionZip	37.07	17.63	36.90
	FastVID	37.10	17.66	37.00
	FlashVID	41.59	21.56	40.50
	SlashVID	**43.92**	**23.23**	**42.30**

SlashVID generalizes across model generations consistently outperforming all baselines at both 5% and 2% retention. An extended architectural comparison with FlashVID is provided in Appendix E.

5.4 Ablation Studies

We conduct ablation studies on Charades-STA with Qwen3-VL-4B at 2% visual token retention to examine the contribution of each module. Table 4 reports all non-trivial combinations of ASG, DMM-IS, and Bi-TAC. Several findings clarify the division of labor between the two paths. First, Bi-TAC alone achieves 44.95 mIoU, outperforming DMM-IS alone (41.27) by 3.68 points. This larger gain shows that preserving temporal context through aggregation is the main source of robustness under extreme compression. Second, ASG provides consistent gains when paired with either path: +0.93 for DMM-IS and +0.30 for Bi-TAC. This improvement arises from ASG's segment-wise budget reservation ($B_k = BT_k$), which prevents globally salient background scenes from consuming the entire video budget and starving quieter segments that contain the target action. Within these protected segment budgets, Bi-TAC preserves contextual evidence, while DMM-IS adds query-relevant boundary cues and diversity. Third, combining DMM-IS with Bi-TAC (45.18) surpasses either in isolation, showing that query-guided selection is complementary to, rather than a replacement for, temporal aggregation. Finally, the full system obtains the highest mIoU (45.83): Bi-TAC supplies

Table 4: Ablation study on Charades-STA (Qwen3-VL-4B, 2% visual token retention).

ASG	DMM-IS	Bi-TAC	R@0.5	R@0.7	mIoU
	√		44.70	21.10	41.27
		√	49.27	24.46	44.95
√	√		45.14	21.21	42.20
√		√	49.76	24.89	45.25
	√	√	50.68	24.81	45.18
√	√	√	**51.21**	**25.21**	**45.83**

Table 5: Efficiency comparison on ActivityNet with Qwen3-VL-4B at 2% visual token retention on a single NVIDIA A800 GPU. Prefilling time is reported in milliseconds (ms).

Method	TFLOPs	Prefilling Time			Score	
		Pruning	LLM Forward	Total	mIoU	(%)
Qwen3-VL-4B	51.3	-	1258.1	1258.1(1×)	42.93	100
VisionZip	0.7	703.2	181.5	884.7 (1.4×)	22.93	53.41
FastVID	0.5	**21.3**	182.4	203.7(6.2×)	27.86	64.90
FlashVID	0.5	47.0	93.8	140.8 (8.9×)	32.70	76.17
SemVID	0.7	46.3	181.8	228.1 (5.5×)	33.53	78.10
SlashVID	0.4	45.4	**85.8**	**131.2(9.6×)**	**34.93**	**81.37**

the dominant improvement and DMM-IS contributes a consistent additional gain. Fine-grained ablations of individual design choices (entropy weighting, diversity selection, token merging and direction, temporal uniformity, and LLM-side pruning layer) are provided in Appendix C.

5.5 Efficiency and Parameter Sensitivity Analysis

Although token compression reduces LLM computation, the compression itself introduces overhead. Table 5 reports an efficiency comparison on ActivityNet using Qwen3-VL-4B on a single NVIDIA A800 GPU at 2% visual token retention. SlashVID achieves a total prefilling time of 131.2 ms, yielding a 9.6× speedup over the uncompressed baseline, making it the fastest among all compared methods. Meanwhile, SlashVID retains 81.37% relative accuracy, substantially outperforming the second-fastest method, FlashVID (8.9×, 76.17%). This demonstrates that SlashVID provides the best accuracy-efficiency trade-off under extreme compression.

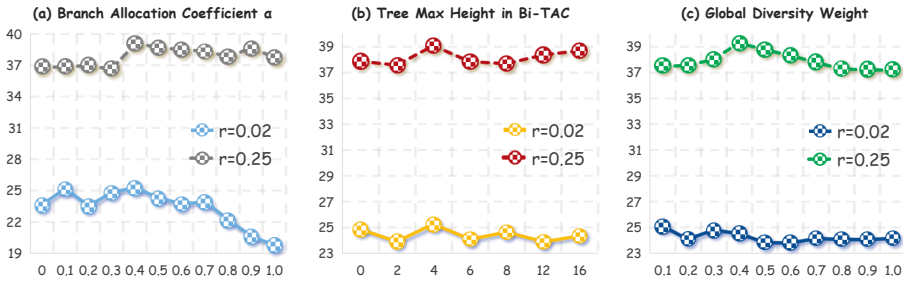


Figure 3: Parameter sensitivity.

Fig. 3(a) evaluates the adaptive α design by sweeping a fixed α across all segments. Performance peaks around 0.4 but degrades toward both extremes: selection-only compression loses the contextual evidence supplied by aggregation, whereas aggregation-only compression loses fine-grained boundary cues. This motivates our per-segment adaptive $\alpha \in [0.2, 0.8]$, which adjusts the balance based on each segment's visual complexity. The search-radius control H in Bi-TAC (Fig. 3(b)) has minimal impact, indicating that the aggregation path is stable across reasonable temporal neighborhoods. Fig. 3(c) reveals a clear sweet spot for the diversity weight β at 25% retention: too low causes redundant selections; too high sacrifices query relevance. At 2% retention the curve flattens, as extreme compression naturally enforces diversity. Overall, SlashVID is robust across reasonable parameter ranges, with defaults near the optimum in all cases. Additional sensitivity analyses are provided in Appendix D.

6 Conclusion

We presented SlashVID, a training-free framework that decomposes extreme visual token compression for VTG into complementary aggregation and selection paths. Segment-wise budgeting and Bi-TAC preserve temporal context from the majority of redundant tokens, while DMM-IS adds diverse query-relevant evidence for boundary refinement. SlashVID retains approximately 80% of full-token grounding performance at 2% retention with 9.6× prefill acceleration.

AI USE STATEMENT

Generative AI tools were used for language editing, LaTeX formatting, and checking the consistency of manuscript descriptions. They were not used to train models, execute benchmark experiments, or generate experimental measurements. The authors independently verified the implementation, experimental protocol, numerical results, citations, and scientific claims, and take responsibility for the final content of this work.

ETHICS STATEMENT

This research follows the ICLR Code of Ethics. We use publicly available academic benchmarks, including Charades-STA, ActivityNet-Captions, LongVideoBench, and Video-MME, together with pretrained vision-language models under their respective licenses and terms of use. SlashVID performs inference-time visual-token compression and temporal grounding; it does not involve human-subject recruitment, the collection of new personal data, or decisions about individuals. We acknowledge that public datasets and pretrained models may contain inherited biases or privacy-sensitive content, which this work does not remove or claim to resolve.

REPRODUCIBILITY STATEMENT

The SlashVID method is specified in Sections 4.2–4.4 and Appendix A, including the segment-wise budget allocation, query-guided selection, temporal aggregation, and pseudocode. Appendix A.3 provides the implementation conventions, model checkpoints, hyperparameters, runtime environment, and token-budget accounting, while Appendix B describes the datasets, preprocessing, prompts, decoding protocol, metrics, and baseline reproduction settings. Additional coverage, ablation, inner-LLM pruning, and sensitivity analyses are given in Appendices B.6, C, C.6, and D. The method uses no stochastic training and uses greedy decoding, so fixed inputs and backend settings yield reproducible results up to ordinary hardware and numerical variation. If this paper is accepted, we will release the source code and evaluation scripts.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

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

Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 961–970, 2015.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pp. 19–35. Springer, 2024.

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

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Ziyang Fan, Keyu Chen, Ruilong Xing, Yulin Li, Li Jiang, and Zhuotao Tian. FlashVID: Efficient video large language models via training-free tree-based spatiotemporal token merging. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=H6rDX4w6Al.

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

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*, pp. 5267–5275, 2017.

Yongxin Guo, Jingyu Liu, Mingda Li, Dingxin Cheng, Xiaoying Tang, Dianbo Sui, Qingbin Liu, Xi Chen, and Kevin Zhao. Vtg-llm: Integrating timestamp knowledge into video llms for enhanced video temporal grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 3302–3310, 2025.

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.

Minseok Kang, Minhyeok Lee, Jungho Lee, Minjung Kim, Donghyeong Kim, Dayeon Lee, Heeseung Choi, Ig-jae Kim, and Sangyoun Lee. Ott-vid: Optimal transport temporal token compression for video large language models. *arXiv preprint arXiv:2605.11803*, 2026.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856.

Jiaqi Li, Shuntian Zheng, Yixian Shen, Jia-Hong Huang, Xiaoman Lu, Minzhe Ni, and Yu Guan. Keeping the evidence chain: Semantic evidence allocation for training-free token pruning in video temporal grounding. In *European Conference on Computer Vision*, 2026a.

Zeqian Li, Shangzhe Di, Zhonghua Zhai, Weilin Huang, Yanfeng Wang, and Weidi Xie. Universal video temporal grounding with generative multi-modal large language models. *Advances in Neural Information Processing Systems*, 38:64426–64455, 2026b.

Kevin Qinghong Lin, Pengchuan Zhang, Joya Chen, Shraman Pramanick, Difei Gao, Alex Jinpeng Wang, Rui Yan, and Mike Zheng Shou. Univtg: Towards unified video-language temporal grounding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2794–2804, 2023.

Dezhao Luo, Jiabo Huang, Shaogang Gong, Hailin Jin, and Yang Liu. Zero-shot video moment retrieval from frozen vision-language models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 5464–5473, 2024.

Leqi Shen, Guoqiang Gong, Tao He, Yifeng Zhang, pengzhang liu, Sicheng Zhao, and Guiguang Ding. FastVID: Dynamic density pruning for fast video large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=2xS4VtpApy.

Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, pp. 510–526. Springer, 2016.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, et al. Lvbench: An extreme long video understanding benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22958–22967, 2025a.

Yi Wang, Xinhao Li, Ziang Yan, Yinan He, Jiashuo Yu, Xiangyu Zeng, Chenting Wang, Changlian Ma, Haian Huang, Jianfei Gao, Min Dou, Kai Chen, Wenhai Wang, Yu Qiao, Yali Wang, and Limin Wang. Internvideo2.5: Empowering video mllms with long and rich context modeling. *arXiv preprint arXiv:2501.12386*, 2025b.

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

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

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5288–5296, 2016.

Xinyi Xu, Hongsong Wang, Guo-Sen Xie, Caifeng Shan, and Fang Zhao. Hitea: Hierarchical temporal alignment for training-free long-video temporal grounding. In *The Fourteenth International Conference on Learning Representations*, 2026.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19792–19802, 2025.

Jun Zhang, Teng Wang, Yuying Ge, Yixiao Ge, Xinhao Li, and Limin Wang. Timelens: Rethinking video temporal grounding with multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10419–10429, 2026.

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

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

Minghang Zheng, Xinhao Cai, Qingchao Chen, Yuxin Peng, and Yang Liu. Training-free video temporal grounding using large-scale pre-trained models. In *European Conference on Computer Vision*, pp. 20–37. Springer, 2024.

A Implementation Details and Algorithm

We provide the complete algorithmic specification of SlashVID. The full pipeline consists of three stages executed sequentially: (1) Adaptive Segment-wise Guidance (ASG) partitions the video and allocates per-segment budgets, (2) Dynamic Max-Min Importance Selection (DMM-IS) selects the most informative tokens, and (3) Bidirectional Temporal Aggregation Clustering (Bi-TAC) aggregates the remaining tokens into compact representatives. Algorithm 1 presents the overall procedure and makes the data flow and segment-level budget accounting explicit.

A.1 Algorithm

Algorithm 1 SlashVID: Complete Pipeline

Require: Video features $\mathbf{V} \in \mathbb{R}^{T\times M\times C}$, visual saliency $\mathbf{A} \in \mathbb{R}^{T\times M}$, text embeddings $\mathbf{Q} \in \mathbb{R}^{L_q\times C}$, retention ratio ρ_v, branch range $[\alpha_{\min}, \alpha_{\max}]$

Ensure: Compressed tokens $\hat{\mathbf{V}} \in \mathbb{R}^{N'\times C}$ where $N' \approx \rho_v \cdot T \cdot M$

$s_t \leftarrow \cos(\bar{\mathbf{v}}_t, \bar{\mathbf{v}}_{t+1})$; partition at $s_t < \tau_s$ into $\{\mathcal{S}_k\}_{k=1}^{K}$
$B \leftarrow \max(1, \lfloor \rho_v M \rfloor)$
for each segment $\mathcal{S}_k$ **do**
  $B_k \leftarrow B|\mathcal{S}_k|$; compute α_k from mean effective rank
  $B_k^{\text{sel}} \leftarrow \max(1, \lfloor \alpha_k B \rfloor)|\mathcal{S}_k|$; $B_k^{\text{agg}} \leftarrow B_k - B_k^{\text{sel}}$
  $\mathcal{I}_k^{\text{sel}} \leftarrow$ DMM-IS$(\mathcal{S}_k, \mathbf{A}, \mathbf{Q}, B_k^{\text{sel}})$
  $\mathcal{U}_k \leftarrow \{(t,i) : t \in \mathcal{S}_k\} \setminus \mathcal{I}_k^{\text{sel}}$
  $(\widetilde{\mathcal{I}}_k^{\text{sel}}, \mathbf{R}_k) \leftarrow$ Bi-TAC$(\mathcal{U}_k, \mathcal{I}_k^{\text{sel}}, B_k^{\text{agg}})$
end for
$\hat{\mathbf{V}} \leftarrow$ TemporalSort$(\bigcup_k (\widetilde{\mathcal{I}}_k^{\text{sel}} \cup \mathbf{R}_k))$
Assign temporal/spatial MRoPE indices to $\hat{\mathbf{V}}$ **return** $\hat{\mathbf{V}}$

A.2 DMM-IS: Detailed Procedure

Algorithm 2 specifies the entropy-weighted query-guided selection with greedy Farthest Point Sampling (FPS). Query entropy weights are computed once over the full video and then reused in each segment, consistent with the formulation in the main paper. This separates video-level query characterization from segment-level budgeted selection.

Algorithm 2 DMM-IS: Dynamic Max-Min Importance Selection

Require: Full-video frame features $\{\bar{\mathbf{f}}_t\}_{t=1}^{T}$, segment features $\mathbf{V}_k \in \mathbb{R}^{T_k\times M\times C}$, visual saliency $\mathbf{A}_k$, text embeddings $\mathbf{Q}$, selection budget B^{sel}, per-frame budget B_{frame}, diversity weight β

Ensure: Selected indices $\mathcal{I}^{\text{sel}}$

$P_{j,t} \leftarrow \text{softmax}_t(\cos(\mathbf{q}_j, \bar{\mathbf{f}}_t)/\tau)$; $H_j \leftarrow -\sum_t P_{j,t} \log P_{j,t}$
$w_j \leftarrow (1 - H_j/\log T)^{0.7}$; normalize $\mathbf{w}$
$\hat{\mathbf{q}} \leftarrow \sum_j w_j \bar{\mathbf{q}}_j$
$\tilde{A}_{t,i} \leftarrow A_{t,i} \cdot (1 + \lambda_q \cdot \cos(\mathbf{v}_{t,i}, \hat{\mathbf{q}})^{+})$
Build seed-first $\mathcal{C}$ with $k_c = \min(M, \max(3B_{\text{frame}}, 16))$ and $|\mathcal{C}| \le \max(T_k, 2048)$
$\rho_i \leftarrow |\{j \neq i : \cos(\mathbf{c}_i, \mathbf{c}_j) > 0.8\}|$
for each frame t **do**
  $s_t^* \leftarrow \arg\max_{i\in\mathcal{C}_t} \tilde{A}_i \cdot (1/(\rho_i + 1))^{\beta}$
  $\mathcal{I}^{\text{sel}} \leftarrow \mathcal{I}^{\text{sel}} \cup \{s_t^*\}$
end for
$d(i,j) \leftarrow (1 - \cos(\mathbf{c}_i, \mathbf{c}_j))\tilde{I}_i(1 + \lambda_q r_i)$
Initialize $d_{\min}(i) \leftarrow \min_{s\in\mathcal{I}^{\text{sel}}} d(i,s)$
while $|\mathcal{I}^{\text{sel}}| < B^{\text{sel}}$ and candidates remain **do**
  $i^* \leftarrow \arg\max_{i\notin\mathcal{I}^{\text{sel}}} d_{\min}(i)$
  $\mathcal{I}^{\text{sel}} \leftarrow \mathcal{I}^{\text{sel}} \cup \{i^*\}$
  $d_{\min}(i) \leftarrow \min(d_{\min}(i), d(i,i^*))$
end whilereturn $\mathcal{I}^{\text{sel}}$

Note on query relevance in $D(i,j)$. The second relevance factor intentionally biases the FPS diversity step toward query-relevant candidates after the initial importance filtering. The normalized importance term $\tilde{I}_i$ combines the saliency-boosted attention score with candidate normalization and first ranks globally salient and query-relevant candidates, while the explicit $(1 + \lambda_q \cdot r_i)$ multiplier prevents the later FPS iterations from selecting geometrically distant but query-irrelevant tokens. This keeps the selection branch diverse without losing boundary-critical evidence. Accordingly, query relevance guides selection without replacing feature-space diversity.

Algorithm 3 specifies the bidirectional temporal aggregation with temporally-uniform density peak clustering. Its steps follow the implementation order from candidate routing to final clustering.

Algorithm 3 Bi-TAC: Bidirectional Temporal Aggregation Clustering

Require: Segment $\mathcal{S}_k$, unselected tokens $\mathcal{U} = \{(\mathbf{u}_{t,i}, t)\}$, protected DMM-IS anchors $\mathcal{I}^{\text{sel}}$, aggregation budget B^{agg}, per-frame allocations $n_{\text{sel}}, n_{\text{agg}}$, merge threshold τ_m, radius-control parameter H, uniformity weight λ_u

Ensure: Updated protected anchors $\widetilde{\mathcal{I}}^{\text{sel}}$ and aggregated representatives $\mathbf{R} \in \mathbb{R}^{\widehat{B}^{\text{agg}} \times C}$, where $\widehat{B}^{\text{agg}} \le B^{\text{agg}}$

if $B^{\text{agg}} = 0$ **then return** $\mathcal{I}^{\text{sel}}$ and $\emptyset$
end if
Set the temporal search radius $W \leftarrow \lceil T_k/(2H) \rceil$ and initialize each $u_i \in \mathcal{U}$ with mass $m_i = 1$ and time $\bar{t}_i = t_i$
Mark $\mathcal{I}^{\text{sel}}$ as protected anchors
Compute each source's best backward and forward candidates within W
Let m_k be the segment midpoint; use backward for $t \le m_k$ and forward for $t > m_k$
Build merge/retaining masks using τ_m and the search radius determined by H
Correct masks using $L_{\text{TAM}} = (n_{\text{sel}} + n_{\text{agg}})T_k$ and one TAM survivor per frame
Freeze masks; then accumulate merged sources into anchors by mass-weighted averaging
$\mathcal{P} \leftarrow \text{retaining_mask} \cap \mathcal{U}$
$c_{ij} \leftarrow 1 - \cos(\mathbf{p}_i, \mathbf{p}_j)$; $\rho_i \leftarrow \exp(-\frac{1}{k}\sum_{j \in \text{kNN}(i)} c_{ij}^2)$
$\delta_i \leftarrow \min_{j:\rho_j > \rho_i} c_{ij}$ (use $\max_j c_{ij}$ for the densest point); $\gamma_i \leftarrow \rho_i \delta_i$
Initialize $\mathbf{n} \leftarrow \mathbf{0}$ and $\mathcal{Z} \leftarrow \emptyset$
$\widehat{B}^{\text{agg}} \leftarrow \min(B^{\text{agg}}, |\mathcal{P}|)$
for $b = 1$ to $\widehat{B}^{\text{agg}}$ **do**
    $i^* \leftarrow \arg\max_{i \notin \mathcal{Z}} \gamma_i (1 - \lambda_u n_{f(i)} / \max(1, b-1))$
    $\mathcal{Z} \leftarrow \mathcal{Z} \cup \{i^*\}$; $n_{f(i^*)} \mathrel{+}= 1$
end for
Assign tokens to nearest centers and compute $\mathbf{r}_c = \frac{\sum_{i \in c} m_i \mathbf{p}_i}{\sum_{i \in c} m_i}$
return $\widetilde{\mathcal{I}}^{\text{sel}}$ and $\mathbf{R} = \{\mathbf{r}_1, \ldots, \mathbf{r}_{\widehat{B}^{\text{agg}}}\}$

A.3 Implementation Details

Video Encoding. Each input video is decoded at a fixed frame rate: 2 FPS for Charades-STA (Sigurdsson et al., 2016) and 1 FPS for ActivityNet-Captions (Caba Heilbron et al., 2015). Frames are resized to a shorter side of 480 pixels for Charades-STA and 256 pixels for ActivityNet-Captions. Because the two datasets use different input resolutions and Qwen3-VL (Bai et al., 2025) applies dynamic visual-token budgeting, the effective number of visual tokens per frame is dataset-dependent rather than fixed. Charades-STA typically produces about $M = 256$ visual tokens per effective frame after the encoder merge, while ActivityNet-Captions produces fewer tokens per frame on average under the shorter-side-256 setting. All retention ratios are computed relative to each video's actual visual-token count after Qwen3-VL preprocessing, so the compression budget adapts to the effective resolution of each input video. In the hybrid setting, the reported ratio r is the target LLM-input budget; the implementation applies this target before layer-24 pruning and rounds token counts to integers (with the required frame-coverage constraint).

Token Compression. All compression operates on the projected features (after the vision-language projector, dimension $C = 2560$ for Qwen3-VL-4B and $C = 4096$ for the official Qwen3-VL-8B). The experiments use the Hugging Face Qwen3-VL checkpoints loaded from the local mirror of `Qwen/Qwen3-VL-4B-Thinking` (thinking disabled at generation time) and the corresponding official 8B checkpoint for the 8B table; no model parameters are fine-tuned. The visual saliency weights are derived from the final ViT layer's self-attention matrix by averaging across all heads

and query positions, yielding a per-token "received attention" score. These scores are aligned with the projected visual tokens through the vision encoder's native token-merging correspondence and then used as the base importance signal for token selection.

ASG Configuration. Segmentation uses frame-mean features with cosine similarity. Boundaries are placed where adjacent-frame similarity drops below $\tau_s = 0.9$. A complementary segmentation strategy ensures at least 2 segments by selecting the lowest-similarity transition when no natural boundary exists. Budget allocation is uniform across segments and proportional to frame count. The branch allocation coefficient α_k is adaptively determined per segment using the effective rank (erank) of salient frame-level token sets, matching the main-paper algorithm: for each frame, we take the top-K_a L2-normalized tokens ranked by saliency, compute the singular-value entropy/effective rank of this $K_a \times C$ matrix, and average the normalized erank over frames in the segment. Segments with higher diversity allocate more budget to the selection branch, while segments with redundant content favor aggregation. We use $[\alpha_{\min}, \alpha_{\max}] = [0.2, 0.8]$, as specified in the main paper.

DMM-IS Configuration. The pre-filter first retains one top-$\tilde{A}$ seed per frame to guarantee frame coverage, then fills the remaining candidate pool using the per-frame top-k_c lists, where $k_c = \min(M, \max(3B_{\text{frame}}, 16))$. The mandatory seeds have priority; optional candidates are added until the total reaches 2048; if $T_k \geq 2048$, no optional candidates are added. Thus, the total candidate count is bounded by $\max(T_k, 2048)$. The attention normalization maps raw saliency scores to $[0.5, 1.5]$ as a distance multiplier. Greedy FPS computes exact cosine distances (no approximation) on the candidate set. The replacement operation described in the main paper is retained as a coverage safeguard and is normally inactive because the seed-first initialization already covers every frame.

Bi-TAC Configuration. Bidirectional merging uses a temporal search radius of $W = \lceil T_k/(2H) \rceil$ frames in a segment containing T_k frames, with the radius-control parameter set to $H = 4$. Operationally, the bidirectional search described in the main paper is implemented by computing the best anchor separately in the backward and forward windows of every unselected token, followed by a deterministic midpoint router: tokens in the first half of a segment use their backward candidate, whereas tokens in the second half use their forward candidate. Thus, H controls the local temporal neighborhood through W. This construction uses both temporal directions across the segment while assigning a single active direction to each source. DMM-IS selected tokens are protected anchors: they can absorb unselected tokens through mass-weighted accumulation but cannot themselves be absorbed, and they remain counted only in the selection budget. Before feature accumulation, the implementation applies the retained-token heuristic $L_{\text{TAM}} = (n_{\text{sel}} + n_{\text{agg}})T_k$, with n_{sel} corresponding to `num_attn_div_tokens` and n_{agg} corresponding to `num_sttm_tokens`, to limit over-merging and restores one TAM survivor in every frame that would otherwise have none. The heuristic acts on the path-specific token masks and should not be interpreted as the aggregation-only constraint $|\mathcal{P}| \geq B^{\text{agg}}$. Only after these mask corrections are finalized are merged sources accumulated into anchors, so a restored token is never also scatter-added to a former anchor. This TAM restoration is separate from DMM-IS's own one-selected-token-per-frame guarantee. Aggregated representatives inherit the retained center/anchor global visual-token index, are sorted together with the updated protected selection outputs by global visual-token index before entering the LLM, and reuse Qwen3-VL's existing temporal/spatial MRoPE position IDs through the retained-token index map. The DPC density estimator uses $k = 7$ nearest neighbors.

Runtime Environment. All experiments use PyTorch 2.6 and Transformers 4.52 on NVIDIA A6000 (48GB) or A800 (80GB) GPUs. Inference is single-GPU with `torch.float16` precision. No batching is used across videos; each sample is processed independently. Greedy decoding with `max_new_tokens=256` and thinking mode disabled.

Complexity Summary. Let $N = T \times M$ be the total visual tokens before compression.

- **ASG**: $O(TMC + \sum_t K_a^2 C)$ for frame pooling/similarity and the frame-wise effective-rank computation. The erank computation takes the top-K_a salient tokens in each frame and computes singular values of the resulting $K_a \times C$ matrix before averaging within each segment; with fixed $K_a \leq 64$, the per-frame singular value computation operates on small, bounded matrices, maintaining a highly scalable linear cost in the total number of frames T.

- **DMM-IS**: $O(NL_qC + C_{\text{cand}}^2 C + B^{\text{sel}} C_{\text{cand}})$; this includes query scoring over all tokens, pairwise similarity and density on the candidate set, and B^{sel} FPS iterations updating pre-computed candidate distances, where $C_{\text{cand}} \leq \max(T_k, 2048)$.
- **Bi-TAC**: $O(NWMC + \sum_k P_k^2 C)$ in the dense-anchor case; the backward and forward candidate searches each inspect up to W neighboring frames with up to M tokens per frame, midpoint routing and mask correction are linear in the number of sources, and DPC runs on the P_k restored survivors in each segment (where $P_k \ll T_k M$ after merging).

In practice, the entire compression pipeline takes 60–110ms per sample on A6000, dominated by the greedy FPS loop in DMM-IS.

A.4 Token Budget Accounting and Coverage

For SlashVID, the before-LLM target in this accounting is $\rho_v = r$, where r is the reported LLM-input retention ratio. The implementation rounds this target to integer per-frame budgets while preserving the required frame coverage. SlashVID then enforces a two-stage accounting mechanism that bounds the output token count while maintaining frame-level temporal continuity across all segments. The allocation and boundary enforcement operate as follows:

1. **Selection Budget Allocation**: Within segment k, the selection quota is $B_{\text{sel}} = b_{\text{sel}} T_k$ with $b_{\text{sel}} = \max(1, \lfloor \alpha_k B \rfloor) \geq 1$. To guarantee unbroken temporal coverage, DMM-IS first assigns exactly one seed token per frame (T_k tokens in total) to the highest-scoring candidate. The remaining $B_{\text{sel}} - T_k$ slots (when $b_{\text{sel}} > 1$) are dynamically allocated across the segment candidate pool via query-entropy-weighted farthest-point selection.
2. **Survivor Pool Consolidation**: Bi-TAC processes the unselected tokens via bidirectional hierarchical merging with threshold τ_m. To prevent local over-merging and retain temporal support across transitions, any temporary frame devoid of unselected survivors receives a restored contextual representative before clustering. These tokens are maintained in a disjoint set from selected anchors, precluding duplicate feature accounting.
3. **Aggregation Budget Capping**: From the consolidated survivor pool $\mathcal{P}$, TU-DPC extracts exactly $\widehat{B}_{\text{agg}} = \min(B_{\text{agg}}, |\mathcal{P}|)$ cluster centers, where $B_{\text{agg}} = (B - b_{\text{sel}}) T_k$. Protected DMM-IS selection outputs remain in the selection pool and are never double-counted in $\widehat{B}_{\text{agg}}$.

Through this decoupled two-stage partitioning, the total output tokens for segment k strictly satisfy:

$$|\widetilde{\mathbf{X}}_k| = B_{\text{sel}} + \widehat{B}_{\text{agg}} \leq B_{\text{sel}} + B_{\text{agg}} = B_k, \tag{15}$$

where $B_k = B \cdot T_k$. Across the entire video, the total number of retained tokens satisfies $\sum_k |\widetilde{\mathbf{X}}_k| \leq B \cdot T$, matching the target input ratio r subject to integer rounding and the availability of aggregation survivors. When $|\mathcal{P}| \geq B_{\text{agg}}$, the segment budget is filled exactly to B_k, ensuring maximal utilization of the allotted computational capacity.

Edge case: very short segments. When a segment has only 1–2 frames, B_{sel} may equal T_k (all slots consumed by per-frame seeds, no FPS). In this case, the selection branch naturally provides the highest-importance token per frame. When $B_{\text{agg}} > 0$, Bi-TAC compresses the remaining unselected tokens; when $B = 1$, the method falls back to pure per-frame selection to maintain coverage. This seamless transition guarantees complete temporal robustness.

A.5 Computational Efficiency and Scalability of Bi-TAC

Bi-TAC achieves high computational efficiency by systematically mitigating the quadratic complexity of density-peak clustering through a two-fold architectural design:

- **Hierarchical Merging as an Efficient Pre-filter**: Prior to clustering, bidirectional merging absorbs 70–85% of spatiotemporally redundant tokens into adjacent anchors at threshold τ_m=0.8. This step reduces the candidate pool by nearly an order of magnitude before any distance-matrix computation occurs.

Table 6: Statistics of evaluation benchmarks.

Benchmark	#Videos	#Queries	Video Len. (s)	Task
Charades-STA	1.3k	3.7k	29	VTG
ANet-Captions	4.9k	17.1k	118	VTG
Video-MME (Fu et al., 2025)	900	2.7k	1017.9	VideoQA
LongVideoBench (Wu et al., 2024)	3.8k	6.7k	473	VideoQA

- **Segment-wise Localized Clustering**: Rather than clustering over the entire video globally, TU-DPC executes independently within each temporal segment. For K segments, decomposing P total survivors into local segment pools $\{P_k\}_{k=1}^{K}$ reduces the clustering complexity from $O(P^2)$ to $\sum_{k=1}^{K} O(P_k^2) \approx O(P^2/K)$, delivering an approximately K-fold acceleration.

Implemented with batched FP16 matrix operations on GPUs, TU-DPC accounts for only 10–20% of the vision-side compression time (typically 8–18ms per video). As reported in Table 5, the entire vision-side compression of SlashVID takes only 45.4ms, achieving a 9.6× overall speedup in end-to-end prefilling while preserving full temporal coherence.

B Experimental Setup

B.1 Benchmarks

Table 6 summarizes the benchmarks used in our evaluation.

Charades-STA (Sigurdsson et al., 2016) focuses on short indoor activity clips (∼29s) with fine-grained temporal annotations (3.7k queries, 7.8s average moment); we use the standard test split (1,334 videos, 3,720 queries). **ActivityNet-Captions** (Caba Heilbron et al., 2015) provides untrimmed videos averaging 118s with 17.1k queries (40.2s moments), testing scalability to longer sequences; we report on the val_1 split (4,917 videos, 17,031 queries). **Video-MME** contains 900 videos with 2,700 QA pairs and an average duration of 1017.9s, testing global semantic understanding. **LongVideoBench** contains 3,763 videos with 6,678 hidden-test QA pairs and an average duration of 473s across 17 fine-grained categories.

B.2 Evaluation Protocol

Video Preprocessing. Charades-STA (Sigurdsson et al., 2016): 2 FPS with a shorter side of 480 pixels. ActivityNet-Captions (Caba Heilbron et al., 2015): 1 FPS with a shorter side of 256 pixels. Qwen3-VL (Bai et al., 2025) applies its default encoder merge and dynamic-resolution mechanism, so ActivityNet produces fewer effective tokens per frame under the lower-resolution setting. Retention budgets are computed from the actual post-encoder visual-token count of each video. Ratios in result tables refer to the target LLM-input ratio r; hybrid methods use the same before-LLM target, with integer token rounding in the implementation.

Generation Configuration. We suppress chain-of-thought by appending `</think>` after the prompt. We use greedy decoding with a maximum of 256 tokens.

Baseline Reproduction. All locally reproduced VTG runs share the same Qwen3-VL backbone, frame preprocessing, prompt, timestamp parser, and greedy decoding. The Qwen3-VL-4B runs use the local Hugging Face mirror of `Qwen/Qwen3-VL-4B-Thinking` with thinking disabled, and the 8B runs use the corresponding official Qwen3-VL-8B checkpoint; model weights are unchanged. VisionZip, FastVID, and FlashVID are reproduced by adapting their published token-selection/compression rules within the same Qwen3-VL evaluation harness. FlashVID uses a local snapshot of its official implementation. SemVID entries marked in italics in the main paper are quoted from the SemVID paper and therefore follow its original evaluation pipeline; non-italic SemVID entries are our local reproduction/adaptation where available. For VideoQA generalization, we follow the LLaVA-Video (Zhang et al., 2024) and LLaVA-OneVision (Li et al., 2025) settings reported in the main paper. Published hyperparameters are used where available.

Table 7: Hyperparameter settings used for SlashVID experiments.

Module	Parameter	Value
ASG	Segment threshold τ_s	0.9
	Min segments	2
	Budget coefficient α	adaptive
	α range $[\alpha_{\min}, \alpha_{\max}]$	[0.2, 0.8]
DMM-IS	Query entropy weight λ_q	0.9
	Entropy temperature τ	0.9
	Global diversity weight β	0.7
	Min tokens per frame	1
Bi-TAC	Merging threshold τ_m	0.8
	Temporal uniformity λ_u	1.0
	Search-radius control H	4
	Direction routing	midpoint-routed backward/forward
	TAM min survivors per frame	1
LLM Pruning	Pruning layer	24
	Retention ratio ρ_l	0.4

Table 8: Model architecture parameters for TFLOPs computation.

Model	L	D	D'	h_{kv}	d
Qwen3-VL-4B	36	2560	9728	8	128
Qwen3-VL-8B	36	4096	12288	8	128

Metrics. VTG: Recall@1 at IoU $\{0.5, 0.7\}$ and mIoU. VideoQA: accuracy on multiple-choice questions.

Reproducibility. SlashVID introduces no stochastic training or decoding. With fixed video frames, model weights, prompts, and backend settings, the evaluation protocol is reproducible up to ordinary hardware-level numerical variation.

B.3 Hyperparameter Summary

B.4 TFLOPs Estimation

Qwen3-VL uses GQA with gated SwiGLU FFN. We report prefill compute using a MAC-counting convention, i.e., one multiply-accumulate is counted as one operation. Under the alternative convention that counts multiply and add separately, the absolute numbers are approximately doubled, but relative comparisons are unchanged. Per-layer prefill compute for sequence length n:

$$\text{FLOPs}_{\text{layer}} = 2nD(h_{\text{kv}}d) + 2nD^2 + 2n^2D + 3nDD'. \tag{16}$$

Under the ActivityNet setting reported in the main paper, Qwen3-VL-4B prefill TFLOPs decrease from 51.3 to 0.4 at the 2% LLM-input budget, matching the efficiency comparison in Table 5 of the main paper.

B.5 Instruction Prompts

VTG:

```
Given a textual query: {query}
When does the described content occur in the video? Please
return the final timestamps [start_time, end_time] in seconds
directly in one sentence.
```

VideoQA:

```
Select the best answer to the following multiple-choice
question based on the video. Question: {question} Options:
A. ... B. ... C. ... D. ... Respond with only the letter
of the correct option. The best answer is:
```

Table 9: Temporal coverage and boundary errors on Charades-STA at 1% retention.

Method	ZF Samp. (%)↓	ZF Ratio↓	Start MAE↓	End MAE↓	Mean MAE↓
VisionZip	58.7	8.42	6.73	7.38	7.05
FastVID	99.1	65.33	9.50	8.35	8.93
FlashVID	0.0	0.00	6.09	6.59	6.34
SlashVID	**0.0**	**0.00**	**4.96**	**5.32**	**5.14**

Table 10: Zero-token-frame statistics of the before-LLM compressor output on Charades-STA at LLM-input retention ratios of 2% and 1%.

Ratio	Method	Total	ZF Samp. (%)↓	ZF Ratio↓
2%	VisionZip	3720	36.2	3.85
	FastVID	3720	92.4	48.71
	FlashVID	3720	0.0	0.00
	SemVID	3720	1.75	0.12
	SlashVID	3720	**0.0**	**0.00**
1%	VisionZip	3720	58.7	8.42
	FastVID	3720	99.1	65.33
	FlashVID	3720	0.0	0.00
	SlashVID	3720	**0.0**	**0.00**

B.6 Temporal Coverage and Boundary Error

Table 9 summarizes temporal coverage and boundary errors on Charades-STA at 1% retention, where SemVID cannot operate. VisionZip and FastVID produce widespread temporal holes (58.7% and 99.1% of samples), while SlashVID maintains complete frame coverage at the LLM input. Beyond coverage, SlashVID reduces mean boundary MAE by 19% relative to FlashVID (5.14 vs. 6.34s), consistent with the benefit of temporal aggregation together with query-guided boundary evidence.

We further report zero-frame statistics at both 2% and 1% LLM-input retention (Table 10), alongside fine-grained boundary error distributions at 1% retention (Table 11). Zero-frame metrics evaluate the visual token sequence entering the LLM, quantifying the frame-level temporal continuity preserved by the vision-side compression framework. "Total" denotes the number of Charades-STA evaluation queries. The zero-frame sample rate (ZF Samp.) and ratio (ZF Ratio) measure the percentage of samples and frames that suffer from complete token elimination prior to the LLM. Boundary error metrics (Start/End MAE, MedAE, and signed bias) are computed over the full evaluation set.

Zero-frame analysis. VisionZip and FastVID produce widespread temporal holes (FastVID: 92% at 2%, 99% at 1%), while FlashVID and SlashVID maintain complete frame coverage at the LLM input. SemVID achieves near-zero holes at 2% but cannot operate at 1% because its semantic-unit constraint imposes a higher minimum budget.

Boundary error at 1%. At the most aggressive setting, SlashVID reduces mean boundary MAE by 19% over FlashVID (5.14 vs. 6.34s), with smaller signed biases (+1.72/+0.85 vs. +2.37/+1.93) and lower median errors (2.60/3.00 vs. 3.60/4.30). Compared with pruning-only baselines, the gain is larger (7.05–8.93s to 5.14s), consistent with the benefit of retaining temporal context through aggregation and adding query-guided evidence under extreme compression.

Graceful degradation. Although all methods degrade from 2% to 1%, SlashVID maintains zero-token-frame coverage at the LLM input and the lowest boundary error among the compared 1% methods, indicating that the select-and-aggregate design degrades more gracefully near the vision-side frame-coverage lower bound.

C Fine-Grained Ablation Studies

All experiments are on Charades-STA with Qwen3-VL-4B at 2% retention unless noted.

Table 11: Boundary error on Charades-STA at 1% retention (seconds). SemVID is not shown because its semantic-unit constraint prevents operation at 1% retention.

Method	Start MAE↓	End MAE↓	Mean MAE↓	Start Bias	End Bias	Start MedAE↓	End MedAE↓
VisionZip	6.73	7.38	7.05	+2.13	−1.62	5.00	5.70
FastVID	9.50	8.35	8.93	+7.40	+5.41	7.90	6.30
FlashVID	6.09	6.59	6.34	+2.37	+1.93	3.60	4.30
SlashVID	**4.96**	**5.32**	**5.14**	**+1.72**	**+0.85**	**2.60**	**3.00**

Table 12: Effect of entropy weighting in DMM-IS query guidance.

Query Guidance	R@0.5	R@0.7	mIoU
No query guidance (λ_q=0)	48.60	24.01	44.43
Plain similarity (uniform w_j)	49.30	24.01	44.57
Entropy-weighted	**51.21**	**25.21**	**45.83**

C.1 Entropy Weighting vs. Plain Similarity

Entropy weighting improves R@0.7 by 1.20 over no guidance and raises mIoU by 1.26. Compared with plain similarity (uniform w_j), it gives a further 1.26 mIoU gain, showing that query guidance supplies a useful refinement to the aggregation-centered design. The result supports using temporally concentrated query evidence to identify action-discriminative concepts without hand-crafted linguistic heuristics.

C.2 Diversity Selection vs. Top-k

Full DMM-IS improves R@0.5 by 2.96 and mIoU by 1.87 over top-k. The progression shows that query guidance and diversity provide complementary refinements within the selection path: ADTS adds spatial diversity over top-k (+0.69 mIoU), DMM-IS without diversity seeding adds query guidance (+0.95 mIoU over top-k), and the full combination yields the largest gain (+1.87 mIoU). These gains are in addition to the larger improvement from the Bi-TAC aggregation path reported in the module ablation.

C.3 Bidirectional vs. Unidirectional Merging and Merging Isolation

Isolating the token merging operation. To isolate the effect of token merging while keeping the rest of the framework intact, the top part of Table 14 keeps the ASG branch budget split (α_k) and DMM-IS selection unchanged while comparing three aggregation variants: (1) *Direct TU-DPC (w/o token merging)*: unselected tokens bypass local anchor merging and are directly clustered by TU-DPC into B_{agg} representatives. Without local anchor consolidation, raw redundant tokens can interfere with density-peak center selection, degrading mIoU to 42.50 (consistent with the high-threshold regime τ_m=0.95 in Table 19). (2) *Backward-only merging*: merging unselected tokens into backward anchors improves mIoU to 44.44 (+1.94 over no merging). (3) *Bidirectional merging*: midpoint-routed bidirectional merging further improves mIoU to 45.83 (+1.39 over backward-only, with R@0.5 +2.10 and R@0.7 +1.07). Applying backward anchors in the first half and forward anchors in the second half avoids routing all visual evidence in a single temporal direction and produces a more balanced segment-level aggregation pattern.

Clarification of selection-only configurations (40.65 vs. 42.20). The bottom part of Table 14 clarifies the distinction between two selection-only baselines that completely omit the Bi-TAC aggregation branch: (1) *Hybrid schedule (42.20 mIoU)*: In the main paper's Table 4 (Row 3, ASG + DMM-IS), all configurations use the target 2% LLM-input budget with layer-24 inner-LLM pruning (ρ_l=0.4). The input sequence is pruned after layer 24, achieving 42.20 mIoU. (2) *Before-LLM-only baseline (40.65 mIoU)*: When evaluated strictly before the LLM at r=2.0% without inner-LLM pruning, selection-only performance drops from 42.20 to 40.65. This 1.55 mIoU gap is consistent with the general effect of the later pruning stage documented in Table 17 (where omitting inner-LLM pruning reduces the full SlashVID performance by 2.38 mIoU at 2% retention).

Table 13: Effect of diversity-aware selection in DMM-IS.

Selection Strategy	R@0.5	R@0.7	mIoU
Top-k by attention	48.25	24.46	43.96
ADTS (attention + spatial div.)	49.03	24.87	44.65
DMM-IS w/o diversity seed	50.16	24.84	44.91
DMM-IS full	**51.21**	**25.21**	**45.83**

Table 14: Ablation of token merging in Bi-TAC on Charades-STA (2% retention). Top: isolating the token merging operation while keeping branch budgets and DMM-IS selection strictly identical. Bottom: branch-level comparison with selection-only under different pruning schedules.

Configuration	R@0.5	R@0.7	mIoU
Isolated Merging Ablation (branch budgets and DMM-IS kept identical)			
Direct TU-DPC (w/o token merging)	47.18	22.85	42.50
Backward-only merging + TU-DPC	49.11	24.14	44.44
Bidirectional merging + TU-DPC	**51.21**	**25.21**	**45.83**
Branch-Level Ablation (w/o Bi-TAC aggregation branch)			
Selection-only (before-LLM only, r=2%)	43.58	20.94	40.65
Selection-only (aligned hybrid, Table 4 Row 3)	45.14	21.21	42.20

C.4 DPC WITH/WITHOUT TEMPORAL UNIFORMITY

Temporal uniformity improves mIoU by 2.74 (43.09→45.83) and R@0.5 by 4.17. Without the uniformity penalty, density-peak clustering can over-select centers from a few visually dense frames, leaving fewer aggregated representatives for other temporal regions. The uniformity term (λ_u=1.0) penalizes repeated center selection from the same frame and encourages a more balanced distribution of aggregated context. The mandatory DMM-IS seed still provides one input token per frame; this ablation concerns the additional contextual representatives produced by Bi-TAC.

C.5 LLM-SIDE PRUNING LAYER

The layer sweep shows that layer 24 gives the best performance under the fixed pruning ratio ρ_l=0.4. Layer 16 prunes too early and causes a clear mIoU drop, consistent with visual tokens having undergone less multimodal self-attention interaction before pruning. Layers 20 and 28 are more stable than layer 16 but still underperform layer 24, showing that the pruning point matters under extreme visual-token budgets. We select layer 24 because it applies the reduced visual-token sequence to the final 12 of 36 layers and provides the best measured accuracy–efficiency trade-off among the tested pruning layers.

C.6 EFFECT OF LLM-SIDE PRUNING

The retention ratios reported in the main tables denote the target visual-token ratio at the LLM input. For before-LLM-only methods, this is the only token budget. For hybrid methods with inner-LLM pruning, including FlashVID and SlashVID, the same input ratio r is used before layer 24, followed by pruning with ρ_l=0.4 for the remaining layers. Integer rounding and frame-level constraints determine the realized input count for each video. VisionZip, FastVID, and SemVID are evaluated as before-LLM-only methods following their original designs; FlashVID and SlashVID use the same layer-24 pruning protocol.

Before-LLM input budgets. For Qwen3-VL-4B and the official Qwen3-VL-8B (L=36 for both), hybrid methods use $N_b \approx rN_v$ before layer 24, with integer rounding applied per video and the inner-LLM ratio ρ_l=0.4 applied afterwards. Thus, the reported ratio directly describes the visual-token budget entering the LLM, while the later pruning step reduces the token count for the remaining layers.

Effect of inner-LLM pruning on SlashVID. Table 17 compares the complete hybrid schedule with a before-LLM-only SlashVID schedule across retention ratios on Charades-STA (Qwen3-VL-4B). Both configurations use the same target LLM-input ratio r; the hybrid schedule additionally applies

Table 15: Effect of temporal uniformity in density-peak clustering.

Clustering Strategy	R@0.5	R@0.7	mIoU
DPC without uniformity (λ_u=0)	47.04	22.74	43.09
DPC with uniformity (λ_u=1.0)	**51.21**	**25.21**	**45.83**

Table 16: Effect of LLM-side pruning layer and ratio.

Layer	LLM-input ratio	ρ_l	R@0.5	R@0.7	mIoU
–	r	1.0 (no pruning)	49.62	24.35	44.69
16	r	0.4	45.27	22.96	41.91
20	r	0.4	48.98	25.16	44.47
24	r	0.4	**51.21**	**25.21**	**45.83**
28	r	0.4	48.90	23.95	44.48

layer-24 pruning. This table therefore evaluates the complete schedule rather than isolating pruning as a single factor.

At 25% retention, the hybrid schedule differs from the before-LLM-only schedule by 0.58 mIoU. The gap increases at lower budgets (+1.21 at 5%, +2.38 at 2%, +2.59 at 1%), showing that applying attention-based pruning after the same LLM-input budget is increasingly useful under extreme compression. In the fixed-input comparison of Table 16, layer-24 pruning improves mIoU from 44.69 to 45.83 at 2%, providing a direct isolation of the pruning operation for that setting.

Crucially, even without inner-LLM pruning, SlashVID at 2% (43.45 mIoU) still surpasses FlashVID *with* inner-LLM pruning (41.90), confirming that the primary performance gains originate from vision-side compression quality rather than solely from the inner-LLM pruning mechanism.

D Parameter Sensitivity

D.1 Segment Threshold τ_s

Table 18: Sensitivity to τ_s on Charades-STA (2% retention).

τ_s	0.7	0.8	0.85	0.9	0.95
mIoU	44.30	44.63	44.33	**45.83**	44.58

All settings stay within a 1.53 mIoU band (44.30–45.83), demonstrating strong robustness. Because boundaries are inserted when $s_t < \tau_s$, too low a threshold (τ_s=0.7) creates fewer boundaries and can under-segment the video, merging semantically distinct scenes into one segment and allowing globally salient regions to dominate the budget. Too high a threshold (τ_s=0.95) creates more boundaries and can over-segment the video, producing many short segments where the per-segment budget becomes too small for effective selection/aggregation. The optimum at τ_s=0.9 balances these effects: it detects meaningful scene transitions in Charades-STA's indoor activity videos (average 29s) without fragmenting continuous actions.

D.2 Merging Threshold τ_m

Table 19: Sensitivity to τ_m on Charades-STA (2% retention).

τ_m	0.6	0.7	0.8	0.9	0.95
mIoU	42.18	43.74	**45.83**	44.12	42.67

The sweep has a clear optimum at τ_m=0.8. The merging threshold exhibits a 3.65 mIoU range (42.18–45.83), making it the most sensitive parameter among those tested. At τ_m=0.6, moderately similar tokens are merged aggressively, which can blur distinct visual evidence. At τ_m=0.95, more

Table 17: SlashVID with vs. without inner-LLM pruning on Charades-STA (Qwen3-VL-4B).

Ratio	Config	LLM-input ratio	R@0.5	R@0.7	mIoU
25%	w/ inner pruning	r	64.32	39.06	55.25
	w/o inner pruning	r	63.68	37.28	54.67
5%	w/ inner pruning	r	57.55	31.45	50.49
	w/o inner pruning	r	56.05	29.78	49.28
2%	w/ inner pruning	r	51.21	25.21	45.83
	w/o inner pruning	r	47.63	23.25	43.45
1%	w/ inner pruning	r	42.90	20.22	40.21
	w/o inner pruning	r	38.58	17.74	37.62

redundant tokens remain for the fixed-budget DPC stage to resolve. The default τ_m=0.8 provides the best empirical balance between preserving distinct evidence and reducing the survivor count before clustering.

D.3 Frame Rate Sensitivity

Table 20: Sensitivity to input frame rate on Charades-STA (2% retention).

FPS	0.5	1.0	2.0	3.0	4.0
R@0.5	49.52	49.33	**51.21**	50.38	48.95
R@0.7	24.30	23.92	**25.21**	24.44	23.23
mIoU	44.59	44.64	**45.83**	44.93	44.10

SlashVID exhibits consistent stability across varying input frame rates: mIoU remains within a narrow 1.73-point band (44.10–45.83) across sampling rates from 0.5 to 4.0 FPS. Lower FPS condenses action transitions while reducing sequence length; higher FPS provides richer temporal observations with denser evidence. Across all tested configurations, SlashVID's segment-wise allocation and one-token-per-frame guarantee consistently maintain unbroken temporal connectivity, demonstrating robust grounding performance irrespective of temporal sampling frequency.

For Charades-STA, the default 2.0 FPS achieves the best performance in our setting, providing dense enough temporal sampling for moments averaging 7.8s while keeping per-frame compression manageable. ActivityNet uses 1 FPS with shorter-side-256 inputs, as described in the evaluation protocol.

E Detailed Comparison with FlashVID

Table 21: Design comparison between FlashVID and SlashVID.

Design Choice	FlashVID	**SlashVID**
Token selection	Attention-diversity	DMM-IS (query+FPS)
Query guidance	None	Entropy-weighted
Diversity mechanism	Implicit (attn-div)	Explicit greedy FPS
Budget allocation	Global	Segment-wise (ASG)
Aggregation merging	Unidirectional	Bidirectional
Cluster selection	Plain DPC	TU-DPC (uniformity)
LLM-side pruning	Layer-24 pruning	Layer-24 pruning
VTG-specific design	No	Yes

FlashVID is the strongest general-purpose video token compression baseline among those evaluated in the main tables. Its attention-diversity selection encourages spatial spread but does not condition token selection on the grounding query. SlashVID's DMM-IS combines entropy-weighted query guidance with explicit greedy FPS, biasing selection toward query-relevant evidence while maintaining diversity.

FlashVID's unidirectional merging restricts every source to the same temporal direction. Bi-TAC precomputes candidates in both directions and uses midpoint routing to apply backward anchors in the first half of a segment and forward anchors in the second half; this variant improves mIoU by 1.39 over backward-only merging in Table 14. Additionally, FlashVID uses plain DPC for cluster-center selection, whereas TU-DPC penalizes repeated centers from the same frame; this design improves mIoU by 2.74 in Table 15. Finally, ASG partitions the video and reserves a length-proportional budget for each segment, preventing a small number of globally salient regions from consuming the entire video-level budget.

F Scope of VTG Evaluation on Long-Horizon Benchmarks

The long-horizon VTG datasets TACoS (**?**) and Ego4D-NLQ (**?**) contain minute-scale, untrimmed videos and substantially different visual domains from Charades-STA and ActivityNet-Captions. We do not include them in the quantitative compression comparison because a controlled study would require a separate specification of frame sampling, spatial resolution, context truncation, and token-budget alignment for these much longer sequences. Applying the short-video protocol directly would conflate compressor quality with dataset-specific preprocessing and context limits. We therefore restrict the controlled VTG comparison to Charades-STA and ActivityNet-Captions, where all methods use identical inputs, prompts, parsing, and decoding settings; extending the same comparison to long-horizon VTG is left for future work.