TIDE: TEMPORAL INFORMATION-DENSE EVIDENCE COMPRESSION FOR VIDEO TEMPORAL GROUNDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Video temporal grounding (VTG) usually requires Large Vision-Language Models (LVLMs) to localize query-relevant events and accurately predict their start and end timestamps in untrimmed videos, incurring substantial computational and memory costs. Existing token compression methods, mostly developed for general video understanding, reduce visual redundancy by pruning or merging tokens based on feature similarity or visual saliency. However, these appearance-based criteria are not well aligned with temporal grounding: precise localization relies on subtle state transitions and coherent entity evolutions across frames rather than visual appearance alone. Pruning or merging tokens based solely on visual appearance can fragment entity trajectories and attenuate temporal contrasts between adjacent states. To address this problem, we propose TIDE, a token compression framework that shifts the objective from removing appearance-based redundancy to preserving temporal evidence and maintaining structural continuity. TIDE identifies piecewise-stable states by tracking cross-frame token correspondences, allocates token budgets based on query-conditioned transition signals, and recovers peripheral context through soft state-compatible routing. At a visual token retention rate of 12.5% on ActivityNet-Captions, TIDE outperforms the full-token baseline in mIoU with Qwen3-VL-8B and achieves a 4.7× speedup for compression plus LLM prefilling with Qwen3-VL-4B. The code is available at `https://anonymous.4open.science/r/TIDE-1125`.

1 INTRODUCTION

Large Vision-Language Models (LVLMs) (Comanici et al., 2025; Bai et al., 2025; Wang et al., 2025) increasingly use densely sampled video frames to localize events in untrimmed videos. Processing these frames can involve tens of thousands of visual tokens, making LLM prefilling computationally expensive and memory-intensive. Existing token compression methods (Huang et al., 2025; Fan et al., 2026) reduce these costs by pruning, selecting, or merging tokens based on visual similarity, diversity, or attention. These criteria help identify appearance-based redundancy, but video temporal grounding (VTG) (Gao et al., 2017) also relies on temporal relationships among tokens.

VTG requires models to predict the start and end timestamps of a query-relevant event. Recognizing the event alone is therefore insufficient: the model must distinguish the states observed during the event from those immediately preceding and following it. Two visually similar tokens from adjacent frames can lie on opposite sides of an event boundary and provide complementary evidence for localizing that boundary. By contrast, a visually salient token may reflect camera motion, illumination changes, or background variations that are unrelated to the query. A token's value for grounding therefore depends in part on its contribution to identifying query-relevant state transitions; appearance-based importance scores may not fully capture this contribution.

This distinction is important for both token selection and fusion. Under a limited token budget, discarding unselected tokens can remove context needed to interpret the retained evidence (Li et al., 2026a). Merging them into selected anchors recovers context (Yang et al., 2025; Shen et al., 2025; Fan et al., 2026), but appearance similarity can favor combinations across an event boundary: the actor and background may remain visually similar even as the queried action starts or ends. Averaging across these states can weaken the temporal contrast needed to localize the transition (Figure 1(a)). Effective compression should recover peripheral context while preserving the temporal distinctions

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.

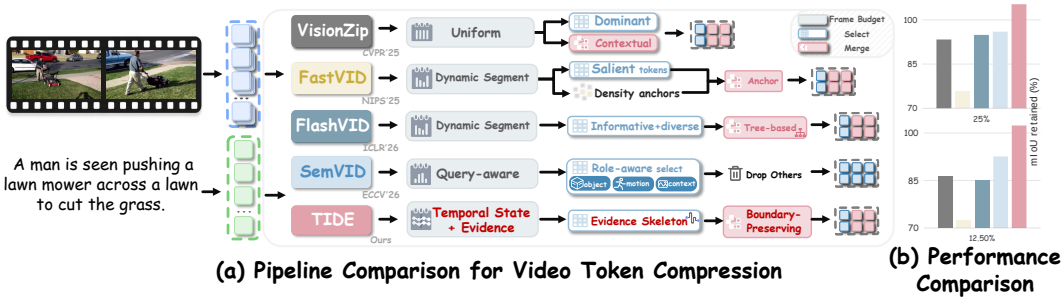

Figure 1: (a) Pipeline comparison between existing token compressors and TIDE which preserves transition evidence while recovering peripheral context through a shared temporal reference. (b) Relative mIoU (%) on ActivityNet-Captions with Qwen3-VL-8B-Thinking at 25% and 12.5% retention (Table 1). Colors follow (a); SemVID percentages use its reported full-token baseline.

that make the event localizable. This calls for a common temporal reference to guide both token allocation and information fusion.

We propose Temporal Information-Dense Evidence Compression (TIDE), a training-free token compression framework that preserves temporal evidence through a shared representation of temporal dynamics. The Temporal State Field (TSF) tracks token correspondences across frames to measure transitions and identify piecewise-stable states. Evidence Density Allocation (EDA) uses these transitions, query relevance, and spectral diversity to distribute each temporal unit's fixed token budget into per-frame quotas. Within these per-frame quotas, Evidence Skeleton Selection (ESS) selects diverse anchors and discounts repeated content using the same transition signal. Boundary-Preserving Fusion (BPF) recovers context from non-anchor tokens, using the shared state representation to penalize cross-state routing of query-relevant transition evidence. Allocation, selection, and fusion therefore act on a common description of how the video changes over time.

Our main contributions are as follows:

- We formulate visual token compression for VTG as a temporal evidence preservation problem under a fixed token budget, accounting for state transitions relevant to event boundary localization.
- We introduce TIDE, a training-free token compression framework that uses a shared temporal state representation to coordinate query-conditioned budget allocation, diverse anchor selection, and state-compatible context fusion.
- TIDE obtains the highest mIoU among the evaluated token compression methods on both VTG benchmarks at both tested token retention ratios. At a visual token retention ratio of 12.5% on ActivityNet-Captions, it achieves 98.12% and 102.37% of full-token mIoU with Qwen3-VL-4B and Qwen3-VL-8B, respectively. TIDE also achieves a $4.7\times$ speedup for compression plus LLM prefilling with Qwen3-VL-4B.

2 Related Work

Video temporal grounding with LVLMs. Supervised VTG methods learn temporal localization from annotated videos (Lin et al., 2023; Zhang et al., 2026; Li et al., 2026b). Training-free approaches use frozen VLMs with proposal scoring or hierarchical temporal decomposition (Zheng et al., 2024; Luo et al., 2024; Xu et al., 2026). Dense frame sampling supports timestamp resolution, but passing the resulting visual sequence to an LVLM incurs self-attention cost quadratic in the complete input length. Token compression addresses this inference cost while retaining the temporal evidence needed for localization.

Visual token compression for LVLMs. Token compression originated in image Transformers via spatial pruning and merging (Bolya et al., 2023; Chen et al., 2024; Xing et al., 2024). For video understanding, recent compressors explore spatiotemporal redundancy reduction. VisionZip (Yang et al., 2025) uses visual attention to select informative tokens without query conditioning. FastVID (Shen et al., 2025) uses temporal segmentation and density-based pruning, with

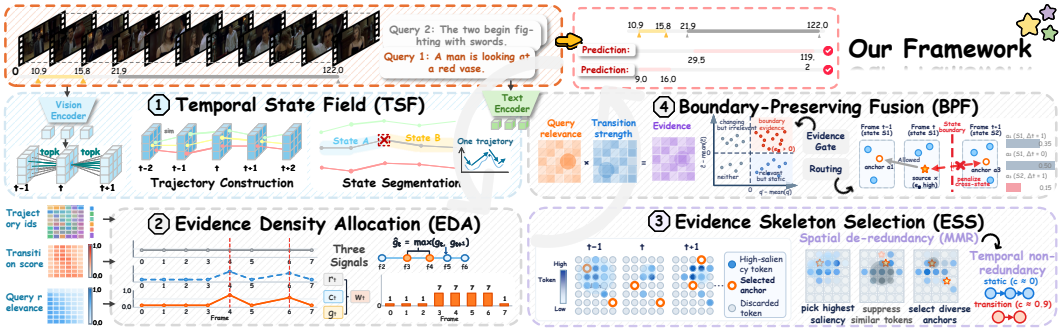

Figure 2: Overview of TIDE. 1) TSF tracks token correspondences across the full video to identify transitions and piecewise-stable states. Within each temporal unit of up to L frames, 2) EDA uses these transitions, query relevance, and spectral diversity to distribute the unit token budget into per-frame quotas. 3) ESS fills these quotas with diverse anchors while discounting temporal repetition. 4) BPF then recovers peripheral context through soft routing, using the shared states to discourage cross-state mixing.

frame-level differences that can respond to both camera motion and action changes. FlashVID (Fan et al., 2026) combines attention and diversity-based selection with tree-based spatiotemporal merging. Similarity-driven cross-frame merging can mix tokens on opposite sides of an event boundary. SemVID (Li et al., 2026a), the closest task-specific baseline, preserves semantic evidence chains for VTG through query-aware budgeting and token pruning. Our TIDE investigates a complementary design: token correspondence states guide allocation and softly constrain context fusion.

3 Preliminaries

Task formulation and visual token representations. Given an untrimmed video V and a natural language query q, video temporal grounding (VTG) aims to predict a temporal segment (t_s, t_e) that corresponds semantically to the queried event. In Large Vision-Language Models (LVLMs), the video is uniformly sampled into T frames and encoded by a vision encoder and projector into visual tokens $\mathbf{H} = \{\mathbf{h}_{t,i} \in \mathbb{R}^d \mid 1 \le t \le T, 1 \le i \le M\}$, where M denotes the number of tokens per frame and $N_v = TM$ is the total visual sequence length. Together with N_t retained prompt and formatting tokens, the sequence fed to the language model has length $N = N_v + N_t$. Dense frame sampling supports fine temporal resolution. As T increases, the visual sequence can reach tens of thousands of tokens and account for most of the input ($N_v \gg N_t$).

Visual token compression. Training-free token compression seeks a mapping Φ that converts the full visual representation $\mathbf{H}$ into a compact sequence $\tilde{\mathbf{H}} = \Phi(\mathbf{H}, q, r) \in \mathbb{R}^{K \times d}$ before LLM prefilling, where $r \in (0, 1]$ is the target retention ratio. TIDE assigns integer token budgets B_u to temporal units, giving $K = \sum_u B_u \approx rN_v$. For a Transformer with L_{LM} layers and full attention, compression changes the quadratic attention cost from $\mathcal{O}(L_{\text{LM}} N^2 d)$ to $\mathcal{O}(L_{\text{LM}} (N')^2 d)$, where $N' = K + N_t$. For VTG, the compressed representation must preserve the evidence that distinguishes adjacent temporal states.

4 Methods

4.1 Overview

TIDE compresses visual token sequences prior to LLM prefilling without modifying model weights (Figure 2).

Visual token representation and unit partitioning. Let $\mathbf{H} = \{\mathbf{h}_{t,i} \in \mathbb{R}^d\}_{t=1,i=1}^{T,M}$ denote projected visual tokens. Visual saliency $s_{t,i}$ averages the attention received by each patch over heads and query positions in the final vision block, then pools these scores onto the output token grid. Query

relevance uses prompt embeddings $\mathbf{e}_v$ in the shared LLM input space:

$$q_{t,i} = \max_{v \in \mathcal{T}} \cos(\mathbf{h}_{t,i}, \mathbf{e}_v), \qquad \hat{q}_{t,i} = \mathcal{N}_u(q)_{t,i}, \qquad \hat{s}_{t,i} = \mathcal{N}_u(s)_{t,i}. \tag{1}$$

Here $\mathcal{N}_u$ denotes min–max normalization over tokens in temporal unit u, with constant inputs mapped to zero. Prompt embeddings come from the backbone's input-embedding layer after chat formatting and removal of vision boundary tokens. Appendix E.3 specifies attention pooling and numerical conventions.

The sequence is partitioned into disjoint temporal units $\mathcal{U}_u = \{t \mid (u-1)L < t \le \min(uL, T)\}$ with reference length $L = 8$. Each unit receives an exact token budget $B_u = \mathrm{clip}(\lfloor r\,|\mathcal{U}_u|\,M \rceil, 1, |\mathcal{U}_u|M)$. TSF runs once on the full video to establish continuous trajectories and states. EDA, ESS, and BPF operate within each unit and preserve $\sum_{t \in \mathcal{U}_u} b_t = B_u$, bounding local computation while maintaining a global temporal reference. The four stages are coordinated through a shared temporal reference: TSF identifies state transitions and piecewise-stable states across frames, EDA distributes each unit token budget into per-frame quotas, ESS selects anchors within these quotas, and BPF recovers peripheral context while discouraging cross-state mixing.

4.2 Temporal State Field (TSF)

Video temporal grounding benefits from distinguishing action-related changes from other visual fluctuations. TSF constructs a temporal reference by linking token representations across adjacent frames using spatial-semantic affinities to form token trajectories, then partitioning these trajectories into piecewise-stable feature states.

Spatio-temporal message passing. Tokens are first contextualized using correspondences between adjacent frames. For frames $(t, t{+}1)$, we compute an affinity matrix $A^{(t)} \in \mathbb{R}^{M \times M}$ balancing appearance similarity with proximity on a normalized token-index grid:

$$A^{(t)}_{ij} = (1 - \lambda_s)\cos(\mathbf{h}_{t,i}, \mathbf{h}_{t+1,j}) + \lambda_s \left(1 - \frac{\|\mathbf{p}_i - \mathbf{p}_j\|_2}{D_p + \epsilon}\right), \tag{2}$$

where $\mathbf{p}_i \in [0,1]^2$ denotes normalized token-index coordinates (Appendix E.3), $D_p = \max_{i,j} \|\mathbf{p}_i - \mathbf{p}_j\|_2$, $\epsilon = 10^{-6}$, and $\lambda_s = 0.3$. Tokens aggregate their top-$\min(K_a, M)$ neighbors ($K_a = 8$) via one-hop message passing:

$$\mathbf{m}_{t,i} = \sum_{j \in \mathcal{N}_K(t,i)} \frac{\exp(A^{(t)}_{ij}/\tau_a)}{\sum_{j' \in \mathcal{N}_K(t,i)} \exp(A^{(t)}_{ij'}/\tau_a)} \mathbf{h}_{t+1,j}, \qquad \tilde{\mathbf{h}}_{t,i} = \frac{1}{2}(\mathbf{h}_{t,i} + \bar{\mathbf{m}}_{t,i}). \tag{3}$$

Here $\tau_a = 0.1$. The backward message uses the transposed affinity with its own top-K_a softmax. The mean message $\bar{\mathbf{m}}_{t,i}$ averages available forward and backward messages; endpoints use their single neighbor. For $T = 1$, message passing is skipped. Appendix E.3 gives the full definition. For $T \ge 2$, we write $\mathbf{z}_{t,i} = \tilde{\mathbf{h}}_{t,i}$; for $T = 1$, $\mathbf{z} = \mathbf{h}$. These contextualized features are used in matching, allocation, selection, and fusion.

Trajectory linking and state partitioning. We recompute spatial-semantic affinities $\widetilde{A}$ from $\mathbf{z}$ and greedily link each token to $\pi(t,i) = \arg\max_j \widetilde{A}^{(t-1)}_{ji}$. Trajectory identities follow $\mathrm{tr}(t,i) = \mathrm{tr}(t-1, \pi(t,i))$, with $\mathrm{tr}(1,i) = i$. We initialize state identities with $\mathrm{st}(1,i) = i$. These links follow displaced content without an external tracking model. Transition strength measures change between matched features:

$$c_{t,i} = \Big(1 - \cos\big(\mathbf{z}_{t,i},\, \mathbf{z}_{t-1,\pi(t,i)}\big)\Big)_+ \quad (t \ge 2), \qquad c_{1,i} = 0. \tag{4}$$

The score $c_{t,i}$ measures feature change after correspondence matching. Matching displaced content helps suppress spatial displacement-induced responses, whereas changes in the tracked representation yield higher scores. This correspondence-compensated transition signal is shared by EDA, ESS, and BPF. To obtain a discrete reference for transition-aware allocation and routing, we partition token trajectories into piecewise-stable correspondence states:

$$\mathrm{st}(t,i) = \begin{cases} \text{new state ID}, & c_{t,i} \ge \theta \wedge c_{t,i} > \epsilon, \\ \mathrm{st}\big(t-1, \pi(t,i)\big), & \text{otherwise}, \end{cases} \tag{5}$$

where $\theta = \mathrm{Quantile}_{\kappa}\big(\{c_{t',i'}\}_{t'\geq 2}\big)$ is a video-adaptive quantile threshold at level $\kappa = 0.8$, and $\epsilon = 10^{-6}$ safeguards against over-segmenting static scenes. Tokens sharing the same state ID belong to the same visual phase, while state changes indicate candidate event transitions.

EDA aggregates these transition strengths at the frame level ($c_t = M^{-1}\sum_i c_{t,i}$), ESS uses them to discount repeated content, and BPF uses the trajectory and state identities to guide fusion. The same temporal reference thus connects all three compression decisions.

4.3 Evidence Density Allocation (EDA)

EDA distributes the fixed unit token budget B_u among frames as $\{b_t\}_{t\in\mathcal{U}_u}$. Spectral diversity estimates the capacity needed to represent each frame, while query-conditioned transitions assign greater weight to changes relevant to the query.

Representational diversity via effective rank. Frames with complex visual compositions require more tokens than visually redundant scenes. We quantify representational diversity via the effective rank r_t of the contextualized frame matrix $\mathbf{Z}_t \in \mathbb{R}^{M\times d}$:

$$r_t \;=\; \exp\left(-\sum_{j=1}^{M} \bar{\sigma}_j \log(\bar{\sigma}_j + 10^{-12})\right), \qquad \bar{\sigma}_j \;=\; \frac{\sigma_j}{\max(\sum_{j'=1}^{M}\sigma_{j'},\epsilon)}, \tag{6}$$

where σ_j are the singular values of $\mathbf{Z}_t$. Effective rank is small when the spectrum is concentrated and approaches M when it is uniform, providing a feature-based estimate of representational capacity. Batched Gram-matrix spectra supply these singular values (Appendix D.4).

Query-conditioned transition mass and two-sided boundary protection. We compute the trajectory change as $c_t = M^{-1}\sum_i c_{t,i}$ and the query-weighted transition mass as $g_t = (\sum_i \hat{q}_{t,i}c_{t,i})/\max(\sum_i \hat{q}_{t,i},\epsilon)$. EDA applies $\tilde{g}_t = \max(g_t, g_{t+1})$ and $\tilde{c}_t = \max(c_t, c_{t+1})$ within each unit, giving both frames adjacent to a transition access to its signal. At the last frame of the unit, each signal retains its current value.

Composite frame quota allocation. We normalize each $x \in \{r, \tilde{c}, \tilde{g}\}$ by $\hat{x}_t = x_t/\max(\bar{x}_u,\epsilon)$. EDA combines the signals as $v_t = (1-\beta_1)\hat{r}_t + \beta_1\hat{c}_t$ and $w_t = (1-\beta_2)v_t + \beta_2\hat{g}_t$, with reference weights $\beta_1 = 0.4$, $\beta_2 = 0.8$. Capacity-constrained largest-remainder rounding gives:

$$b_t \;=\; \left\lfloor B_u \frac{w_t}{\max(\sum_{t'\in\mathcal{U}_u} w_{t'},\epsilon)} \right\rceil_{\mathrm{LR},M}, \tag{7}$$

with $0 \leq b_t \leq M$ and $\sum_{t\in\mathcal{U}_u} b_t = B_u$. Appendix E.3 defines ties, residual redistribution, and zero-budget frames.

4.4 Evidence Skeleton Selection (ESS)

Given frame quotas b_t, ESS selects anchor tokens $\mathcal{A}_t \subset \{1,\ldots,M\}$ with $|\mathcal{A}_t| = b_t$. Static objects can remain visually salient across many frames, leading saliency-based selection to retain repeated content. ESS therefore considers both temporal change and spatial diversity when selecting anchors.

Temporally discounted saliency. ESS discounts saliency by matched-predecessor similarity: $a_{t,i} = \hat{s}_{t,i} - \gamma\rho_{t,i}$, where $\rho_{t,i} = (1-c_{t,i})_+$ for $t \geq 2$, $\rho_{1,i} = 0$, and $\gamma = 0.3$. Frame-wise min-max normalization yields $\hat{s}^{\mathrm{tmp}}_{t,i} = \mathcal{N}_t(a)_{t,i}$ for MMR. This discount shifts selection toward changing content, and normalization makes its scale comparable to the spatial diversity term.

Intra-frame Maximal Marginal Relevance (MMR). To prevent spatial clustering among selected anchors, ESS selects tokens greedily via MMR, balancing temporal evidence against cosine similarity to previously selected anchors:

$$i^{\star} \;=\; \arg\max_{i\notin\mathcal{A}_t} \Big[\lambda\,\hat{s}^{\mathrm{tmp}}_{t,i} \;-\; (1-\lambda)\max_{j\in\mathcal{A}_t} \cos\big(\mathbf{z}_{t,i},\mathbf{z}_{t,j}\big)_+\Big], \tag{8}$$

where $\lambda = 0.7$ and the redundancy term is zero for the first pick. Selection stops at $|\mathcal{A}_t| = b_t$, giving the unit skeleton $\mathcal{A}_u = \{(t,i) \mid t\in\mathcal{U}_u,\ i\in\mathcal{A}_t\}$ with $|\mathcal{A}_u| = B_u$.

4.5 BOUNDARY-PRESERVING FUSION (BPF)

Let $\mathcal{X}_u = \{(t,i) \mid t \in \mathcal{U}_u, i \notin \mathcal{A}_t\}$ denote the non-anchor tokens. These tokens can contain useful peripheral context. BPF recovers this context by routing them into selected anchors, using temporal proximity and state compatibility to discourage mixing across transitions.

Query-gated boundary evidence and soft compatibility penalty. For non-anchor token $x = (t_x, i_x) \in \mathcal{X}_u$, candidate anchors lie in a local temporal neighborhood $\mathcal{V}(x) = \{a = (t_a, i_a) \in \mathcal{A}_u \mid |t_a - t_x| \le W\}$ ($W = 2$). We define raw boundary evidence as $e_x^{\text{raw}} = (\hat{q}_x - \bar{\hat{q}}_u)_+ \cdot (\hat{c}_x - \bar{\hat{c}}_u)_+$, where $\hat{q}_u$ and $\hat{c}_u$ are unit means of min-max normalized query relevance and transition strength. This score is positive when token x exceeds both means. Unit-level min-max normalization yields the gate e_x, which sets the strength of a soft state-compatibility penalty:

$$\tilde{z}_{xa} \;=\; \cos(\mathbf{z}_x, \mathbf{z}_a) \;-\; \alpha\, e_x \cdot \mathbf{1}\big[t_a \neq t_x \;\wedge\; \mathrm{st}(a) \neq \mathrm{st}(x)\big], \tag{9}$$

where $\alpha = 0.5$. The penalty concentrates protection on query-relevant transitions while allowing low-evidence context to be shared across states. Its soft form keeps cross-state destinations available when a state has few surviving anchors.

Soft transport weights and importance-weighted scatter accumulation. For each source x, we set $\mathcal{K}(x) = \emptyset$ and discard x if $\mathcal{V}(x) = \emptyset$ or its best penalized similarity is below $\tau_m = 0.6$. Otherwise, take $k_x = \min(k, |\mathcal{V}(x)|)$ with $k = 4$ and $\mathcal{K}(x) = \text{top-}k_{x\, a \in \mathcal{V}(x)}\, \tilde{z}_{xa}$. Admitted sources distribute their contribution across these destinations:

$$\omega_{xa} \;=\; \frac{\exp(\psi_{xa})}{\sum_{a' \in \mathcal{K}(x)} \exp(\psi_{xa'})}, \qquad \psi_{xa} \;=\; \frac{\tilde{z}_{xa}}{\lambda_m} \;-\; \frac{(t_a - t_x)^2}{2\sigma^2} \;+\; \eta\, \mathbf{1}[\mathrm{tr}(a) = \mathrm{tr}(x)], \tag{10}$$

where $\lambda_m = 0.5$ is the transport temperature, the Gaussian term enforces temporal proximity ($\sigma = 1.0$), and $\eta = 0.5$ rewards tokens sharing the same token trajectory. Anchors are updated via importance-weighted scatter accumulation:

$$\mathbf{h}_a^{\text{out}} \;=\; \frac{\mathbf{z}_a + \sum_{x:\, a \in \mathcal{K}(x)} \omega_{xa}\, \nu_x\, \mathbf{z}_x}{1 + \sum_{x:\, a \in \mathcal{K}(x)} \omega_{xa}\, \nu_x}, \tag{11}$$

where $\nu_x = \frac{1}{2} + \hat{s}_x$ weights source contribution by saliency, and the unit base weight on $\mathbf{z}_a$ retains an explicit anchor contribution.

Updated anchors retain their original sequence order. Qwen uses the retained native three-dimensional position indices; LLaVA uses contiguous positions in the packed sequence (Appendix E.4). The complete procedure is given in the supplementary material (Appendix D), specifically in Algorithms 1 and 2.

5 EXPERIMENTS

5.1 EXPERIMENT SETTINGS

Benchmarks and Metrics. We evaluate TIDE on Charades-STA (Sigurdsson et al., 2016), which contains short indoor clips with fine-grained actions, and ActivityNet-Captions citecaba2015activitynet, which contains open-domain untrimmed videos. LongVideoBench (Wu et al., 2024) and Video-MME (Fu et al., 2025) test transfer to VideoQA. Appendix E describes the benchmarks and evaluation splits (Table 10). For VTG, we report Recall@1 at IoU thresholds $\{0.3, 0.5, 0.7\}$ and mean IoU (mIoU). For VideoQA, we report multiple-choice accuracy (%). We also report each score as a percentage of its corresponding full-token baseline.

Models and baselines. We evaluate Qwen3-VL-4B-Thinking, Qwen3-VL-8B-Thinking, and Qwen3.5-4B for temporal grounding, and LLaVA-OneVision-7B for VideoQA. Table headings omit the Thinking suffix; thinking output is disabled. Uncompressed full-token inference serves as the 100% baseline. We compare with VisionZip (Yang et al., 2025), FastVID (Shen et al., 2025), FlashVID (Fan et al., 2026), and SemVID (Li et al., 2026a). SemVID's Qwen3-VL VTG scores and relative percentages are quoted from its original paper and marked in italics; these percentages use its full-token baselines. Its Qwen3.5 results are reproduced in our pipeline.

Table 1: VTG results on ActivityNet-Captions. Italic SemVID scores and percentages are quoted from its original paper and use its full-token baselines. Other results are our reproductions, with percentages relative to our same-backbone full-token baseline.

Ratio	Method	Qwen3-VL-4B					Qwen3-VL-8B					Qwen3.5-4B				
		R@0.3	R@0.5	R@0.7	mIoU	(%)	R@0.3	R@0.5	R@0.7	mIoU	(%)	R@0.3	R@0.5	R@0.7	mIoU	(%)
100%	–	58.99	42.05	26.46	42.93	100.00	56.88	40.06	25.66	41.76	100.00	65.49	49.03	32.90	48.64	100.00
25%	VisionZip	47.99	32.71	20.27	35.54	82.79	52.60	36.90	23.13	38.98	93.34	48.39	33.81	22.72	35.50	72.99
	FastVID	48.40	32.41	19.07	35.22	82.04	42.57	27.80	16.60	31.63	75.74	46.88	32.12	19.16	34.19	70.29
	FlashVID	55.46	39.04	24.11	39.43	91.85	54.37	37.29	23.86	39.63	94.90	63.88	47.69	31.85	47.16	96.96
	SemVID	*52.84*	*35.68*	*22.51*	*39.06*	*96.90*	*50.30*	*33.49*	*21.50*	*37.54*	*96.00*	41.22	29.59	20.17	30.32	62.34
	TIDE	**59.97**	**42.48**	**27.02**	**43.74**	**101.89**	**59.77**	**42.70**	**27.56**	**43.98**	**105.32**	**65.10**	**48.09**	**32.19**	**48.01**	**98.70**
12.5%	VisionZip	43.69	28.80	17.28	32.37	75.40	49.13	32.98	20.45	36.04	86.30	46.97	32.82	20.57	34.07	70.04
	FastVID	46.69	30.81	18.11	33.94	79.06	40.43	26.12	15.24	30.17	72.25	45.78	31.03	18.23	33.35	68.56
	FlashVID	49.86	34.97	21.79	36.77	85.65	47.73	33.23	20.93	35.47	84.94	54.48	38.91	25.40	40.24	82.73
	SemVID	*51.96*	*34.73*	*22.18*	*38.49*	*95.40*	*48.32*	*31.93*	*20.47*	*36.21*	*92.60*	38.79	27.23	18.16	29.47	60.59
	TIDE	**57.96**	**40.43**	**25.83**	**42.13**	**98.12**	**58.15**	**41.38**	**26.33**	**42.75**	**102.37**	**61.71**	**44.49**	**28.35**	**45.28**	**93.09**

Table 2: VTG results on Charades-STA. Italic SemVID scores and percentages are quoted from its original paper and use its full-token baselines. Other results are our reproductions, with percentages relative to our same-backbone full-token baseline.

Ratio	Method	Qwen3-VL-4B					Qwen3-VL-8B					Qwen3.5-4B				
		R@0.3	R@0.5	R@0.7	mIoU	(%)	R@0.3	R@0.5	R@0.7	mIoU	(%)	R@0.3	R@0.5	R@0.7	mIoU	(%)
100%	–	81.59	67.98	42.77	57.79	100.00	81.40	68.76	44.52	58.60	100.00	82.10	68.15	44.27	58.80	100
25%	VisionZip	69.49	47.37	21.59	44.41	76.85	76.59	57.80	29.92	51.08	87.17	64.76	48.09	25.73	44.14	75.07
	FastVID	58.49	42.28	20.89	39.11	67.68	57.26	41.21	21.29	38.71	66.06	63.55	43.04	22.42	41.97	71.38
	FlashVID	78.36	63.60	37.90	54.82	94.86	79.41	66.10	**41.96**	56.67	96.71	77.34	61.94	40.00	54.83	93.25
	SemVID	*76.91*	*61.08*	*33.17*	*52.31*	*93.30*	*76.30*	*60.11*	*34.55*	*52.66*	*93.00*	79.38	62.93	38.92	55.46	94.32
	TIDE	**79.54**	**64.95**	**38.95**	**55.62**	**96.24**	**80.65**	**67.04**	41.51	**56.93**	**97.15**	**81.42**	**65.03**	**40.07**	**57.09**	**97.09**
12.5%	VisionZip	67.37	44.68	19.73	42.76	73.99	75.03	55.65	28.01	49.70	84.81	59.30	42.74	22.55	40.25	68.45
	FastVID	55.38	37.82	18.17	36.71	63.52	56.40	39.14	19.52	37.62	64.20	60.83	39.92	20.83	39.92	67.89
	FlashVID	77.66	61.85	36.29	52.82	91.40	78.17	63.82	38.82	54.92	93.72	74.52	58.60	36.80	52.38	89.08
	SemVID	*74.17*	*56.91*	*30.67*	*49.89*	*89.00*	*73.45*	*56.24*	*31.11*	*49.93*	*88.10*	74.30	54.81	31.53	50.28	85.51
	TIDE	**77.92**	**62.15**	**36.88**	**53.69**	**92.91**	**78.85**	**64.44**	**39.27**	**55.34**	**94.44**	**78.71**	**60.97**	**36.87**	**54.75**	**93.11**

Implementation details. TIDE operates without parameter updates. We evaluate retention ratios $r \in \{25\%, 12.5\%\}$; reference hyperparameters and evaluation configurations appear in Appendix E.6. Charades-STA is sampled at 2 FPS with frame shorter side resized to 480 pixels, and ActivityNet-Captions at 1 FPS with shorter side resized to 256 pixels. We use greedy decoding with at most 256 new tokens and fixed seeds. Backbone-native chat templates and a JSON timestamp prompt define the evaluation inputs (Appendix F).

5.2 Main Results

Tables 1 and 2 compare grounding performance across backbones and retention ratios.

Accuracy under compression. At $r = 12.5\%$, TIDE achieves the highest mIoU among the compared compressors in all six backbone–dataset settings. On ActivityNet-Captions, TIDE improves mIoU over FlashVID by 5.36 points with Qwen3-VL-4B, 7.28 points with Qwen3-VL-8B, and 5.04 points with Qwen3.5-4B. The Qwen3-VL-8B result reaches 42.75 mIoU, or 102.37% of the full-token reference, while the Qwen3.5-4B result reaches 45.28 mIoU. TIDE also achieves the highest mIoU at $r = 25\%$, maintaining its lead across both tested retention ratios. Halving retention from 25% to 12.5% on ActivityNet-Captions reduces TIDE's mIoU by 1.23–2.73 points across the three backbones, compared with 2.66–6.92 for FlashVID. On both Qwen3-VL backbones, TIDE at 12.5% also outperforms FlashVID at 25%, preserving more accurate temporal localization with half as many retained visual tokens.

The gains over full-token inference suggest that the frozen model can benefit from a more selective representation of temporal evidence. Query-conditioned allocation emphasizes relevant changes,

Table 3: Design ablations on ActivityNet-Captions ($r = 12.5\%$, Qwen3-VL-4B). Checks denote active TSF, EDA, ESS, and BPF. See Appendix E.5 for variant definitions.

Variant	R@0.3	R@0.5	R@0.7	mIoU
w/o TSF	56.42	38.90	23.94	40.61
Rank-only allocation	56.97	39.29	24.23	41.12
w/o Temporal Discount	56.54	39.31	24.21	40.96
w/o Context Fusion	55.67	38.31	23.33	40.19
Full TIDE	**57.96**	**40.43**	**25.83**	**42.13**

Table 4: TSF variants and fusion controls on ActivityNet-Captions ($r = 12.5\%$, Qwen3-VL-4B). Protocols match Table 3.

Variant	R@0.3	R@0.5	R@0.7	mIoU
TSF: w/o Message Passing	56.33	39.00	24.06	40.71
TSF: w/o Trajectory Signals	56.79	39.16	24.07	40.97
Frame-boundary fusion	56.91	39.25	24.28	41.11
BPF: w/o Temporal Prior	56.85	39.15	24.12	41.01
BPF: w/o Trajectory Reward	56.93	39.33	24.34	41.13
Full TIDE	**57.96**	**40.43**	**25.83**	**42.13**

while fusion supplies anchors with compatible context. This combination can help preserve the distinctions needed for localization (Appendix B.3).

Localization at a strict overlap threshold. At $r = 12.5\%$ on ActivityNet-Captions, Qwen3-VL-8B obtains R@0.7 of 26.33 with TIDE, compared with 25.66 with full tokens and 20.93 with FlashVID. The gain at this strict overlap threshold supports TIDE's ability to retain evidence for precise interval localization.

Comparison with task-specific compression. TIDE achieves higher mIoU than the SemVID scores reported in its original paper for Qwen3-VL and those reproduced in our pipeline for Qwen3.5. The Qwen3.5 comparison evaluates both methods under the same inference setup.

Consistency across benchmarks and backbones. At $r = 25\%$, TIDE retains over 96% of full-token mIoU in all six main-table settings. At $r = 12.5\%$, it remains the top compressed method in every setting, with relative mIoU from 93.09% to 102.37% on ActivityNet-Captions and from 92.91% to 94.44% on Charades-STA. On Charades-STA, TIDE also improves over FlashVID for Qwen3-VL-8B by 0.26 and 0.42 mIoU points at the two ratios.

5.3 Ablation Study

Allocation, selection, and context recovery. Table 3 tests the design choices that connect TIDE's stages. Full TIDE reaches 42.13 mIoU. Removing TSF gives 40.61; using effective-rank-only allocation gives 41.12; and removing the temporal discount while retaining MMR gives 40.96. These comparisons support using temporal evidence both to distribute capacity and to choose anchors within each frame quota. Returning the selected anchors without context fusion gives 40.19 mIoU and lowers R@0.7 from 25.83 to 23.33. With allocation and selection retained, this 1.94-point mIoU gain shows the value of recovering peripheral context after anchor selection.

Temporal representation and fusion controls. Table 4 examines feature contextualization and correspondence signals within TSF. Removing message passing or trajectory signals gives 40.71 or 40.97 mIoU, respectively, compared with 40.61 when both are removed. Replacing BPF with frame-boundary fusion gives 41.11, compared with 42.13 for TIDE's query-gated state routing. Within BPF, removing the Gaussian temporal prior or trajectory reward gives 41.01 and 41.13 mIoU, respectively, supporting the contribution of both components to grounding accuracy in this setting. All three fusion variants outperform the selected-anchor baseline (40.19), while full BPF reaches 42.13. Appendix E.5 specifies the signal dependencies and retained operations for every variant.

BPF's benefit increases with the localization threshold: adding context fusion improves R@0.3 by 4.1% and R@0.7 by 10.7% relative to the selected-anchor variant. Both variants use identical anchors and position indices, linking the gain to recovered feature context. Together with the routing controls, this pattern supports using temporal compatibility to enrich the sparse anchor sequence with context that helps distinguish event boundaries.

5.4 Generalization to Video QA and Efficiency

Video question answering. At $r = 25\%$ on LLaVA-OneVision-7B (Table 5), TIDE obtains 57.7 accuracy on LongVideoBench and 59.3 on Video-MME, exceeding the full-token references of 56.6 and 58.5. It gives the highest overall score among the compared compressors and the strongest

Table 5: Video QA results on Video-MME and LongVideoBench (LVB) with LLaVA-OneVision-7B at 25% visual token retention.

Method	LVB		Video-MME				
	Value	(%)	Short	Medium	Long	Overall	(%)
100%	56.6	100.0	69.9	56.7	48.9	58.5	100.0
VisionZip	56.4	99.7	68.8	**57.3**	48.2	58.1	99.3
PruneVID	55.4	97.9	67.3	54.8	47.2	56.4	96.4
FastVID	55.9	98.8	69.9	56.3	47.4	57.9	99.0
FlashVID	56.8	100.4	**71.2**	57.0	49.3	59.2	101.2
SemVID	56.4	99.7	68.7	57.0	48.7	58.1	99.4
TIDE	**57.7**	**101.9**	70.7	56.6	**50.6**	**59.3**	**101.4**

Table 6: Qwen3-VL-4B efficiency at 12.5% retention. Times in milliseconds (ms) cover compression and LLM prefilling, excluding video decoding and vision encoding.

Method	TFLOPs	Latency (ms)			Score	
		Compress.	Prefill	Total	mIoU	(%)
Qwen3-VL-4B	59.4	-	1263.4	1263.4(1x)	42.93	100.00
VisionZip	4.8	710.9	184.3	895.2(1.4×)	32.37	75.40
FastVID	5.4	23.8	185.9	209.7(6.0×)	33.94	79.06
SemVID	4.8	33.1	184.6	217.7(5.8×)	*38.49*	*95.40*
TIDE	5.9	83.9	184.2	268.1(4.7×)	42.13	98.12

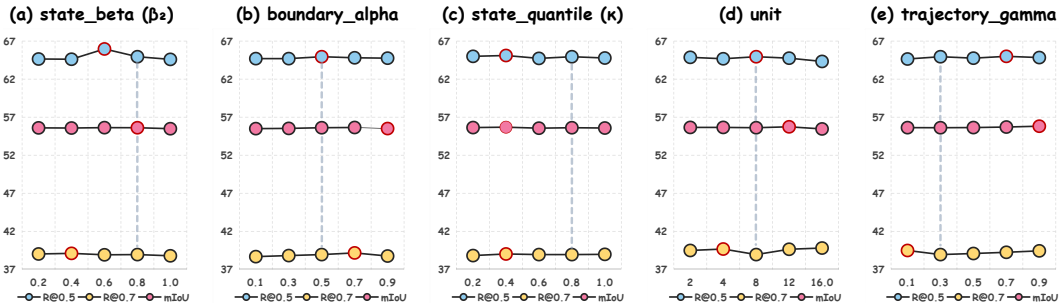

Figure 3: Sensitivity of mIoU to five key hyperparameters on Charades-STA (r=25%, Qwen3-VL-4B). Red-bordered markers indicate the best value per subplot, gray vertical lines mark our default.

score on the Video-MME Long subset (50.6). On these two benchmarks, TIDE preserves VideoQA accuracy with LLaVA-OneVision-7B at 25% retention.

Compression and prefilling efficiency. Table 6 reports compression overhead and LLM prefilling latency measured in our environment at $r = 12.5\%$ on Qwen3-VL-4B with an NVIDIA A800. The measured scope covers visual-token compression plus LLM prefilling, excluding video decoding and vision encoding. TIDE reduces prefilling from $1,263.4$ to 184.2 ms and adds 83.9 ms of compression, for a total of 268.1 ms and a $4.7\times$ speedup over the full-token prefill, while reaching 42.13 mIoU. All compared compressors have similar prefilling latency (184.2–185.9 ms) at this retention ratio. TIDE improves mIoU over FastVID by 8.19 points for 60.1 ms of additional compression, providing a practical option when accurate localization must be retained under a small token budget.

5.5 Parameter Sensitivity Analysis

Figure 3 reports single-factor sweeps on Charades-STA with Qwen3-VL-4B at $r = 25\%$. Across all reported sweeps, mIoU varies by less than 0.3 points per parameter, indicating stable performance around the reference configuration. Increasing the repetition penalty to $\gamma = 0.9$ gives 55.82 mIoU, compared with 55.62 at the reference $\gamma = 0.3$. Varying the temporal unit length from 2 to 16 frames gives a range of 0.28 mIoU points, indicating low sensitivity to the local partition size. Appendix G.2 provides the complete numerical results.

6 Conclusion

We presented TIDE, a training-free visual token compressor for video temporal grounding. Its shared temporal state representation connects frame allocation, anchor selection, and context fusion to preserve evidence for event localization. At 12.5% retention on ActivityNet-Captions, TIDE improves mIoU over FlashVID by 5.36 and 7.28 points on Qwen3-VL-4B and Qwen3-VL-8B, while retaining 98.12% and 102.37% of the corresponding full-token scores. TIDE achieves the highest mIoU among the compared compressors across both VTG benchmarks and both tested ratios. Compression plus LLM prefilling is $4.7\times$ faster in the Qwen3-VL-4B benchmark, and the VideoQA results show that the approach also supports general video understanding.

AI USE STATEMENT

Generative AI tools were used for language editing and checking the consistency of manuscript descriptions. They were not used to train models, execute benchmark experiments, or generate experimental measurements. The authors are responsible for the methods, reported results, interpretations, and final manuscript.

ETHICS STATEMENT

This research follows the ICLR Code of Ethics. We use publicly available academic video benchmarks (Charades-STA, ActivityNet-Captions, LongVideoBench, and Video-MME) under their research licenses and terms of use. TIDE compresses visual tokens at inference time without training the underlying models or collecting additional data from human participants.

REPRODUCIBILITY STATEMENT

Mathematical definitions and pseudocode are provided in Appendices C and D. Dataset descriptions, hyperparameters, and evaluation prompts are provided in Appendices E and F. The anonymous implementation is linked in the abstract. Tables distinguish reproduced results from SemVID results quoted from its original paper, and the timing analysis specifies the measured stages.

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.

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.

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

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

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.

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*, 2025.

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.

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.

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.

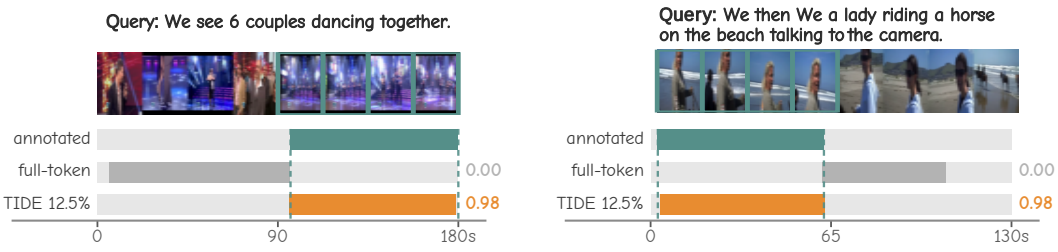

Figure 4: Grounding under 12.5% retention. Keyframes outlined in green fall inside the annotated event. The colors mark the annotated interval in green, the uncompressed full-token prediction in grey, and TIDE in orange. In both samples the full-token baseline returns an interval of reasonable duration attached to the wrong side of the transition, while TIDE localizes the queried event from one eighth of the visual tokens.

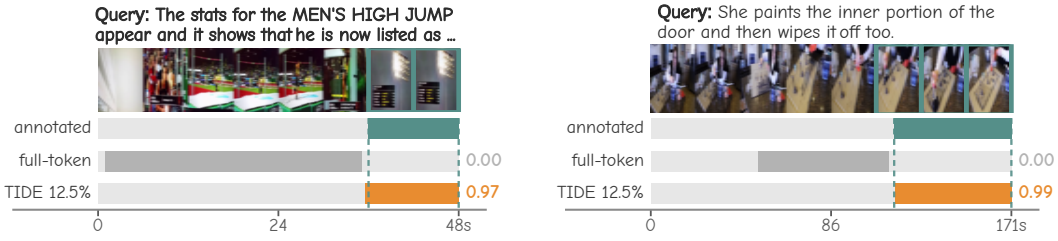

Figure 5: Grounding under 12.5% retention, continued. Both queries name a state that occupies a minority of the video. The full-token baseline spans a long early interval in each case, whereas TIDE isolates the late segment in which the queried state actually holds.

A VISUALIZATIONS

This appendix illustrates TIDE's grounding predictions and the operation of its four modules on ActivityNet-Captions. We use Qwen3-VL-8B-Thinking at $r = 12.5\%$ retention, matching the configuration in Table 1.

Sections A.2 to A.5 follow one video through the pipeline: 180 frames at 1 fps, $M = 66$ merged visual tokens per frame, and 23 temporal units of $L = 8$ frames, with 11880 visual tokens compressed to 1485. Tokens are displayed on a 6×11 grid; TSF uses the 66×1 index coordinates defined in Appendix E.3. The transition field in Figure 7 informs EDA's allocation in Figure 9. ESS selects anchors within the resulting quotas b_t (Figure 11), and BPF routes non-anchor tokens into them (Figure 13). Unit statistics cover all 23 units of this video; window statistics use rolling means over 15 frames.

A.1 GROUNDING RESULTS UNDER AGGRESSIVE COMPRESSION

Figures 4 to 6 compare TIDE at 12.5% retention against uncompressed full-token inference on six ActivityNet samples. Each panel shows a uniformly sampled keyframe strip with the frames inside the annotated event outlined, the annotated interval and the two predictions on a shared time axis, and the text query.

In these examples, the full-token baseline predicts plausible-duration intervals at the wrong temporal locations, whereas TIDE obtains IoUs between 0.97 and 0.99. These selected examples illustrate accurate localization of event position and extent with one eighth of the visual tokens; Tables 1 and 2 report performance across the full evaluation sets.

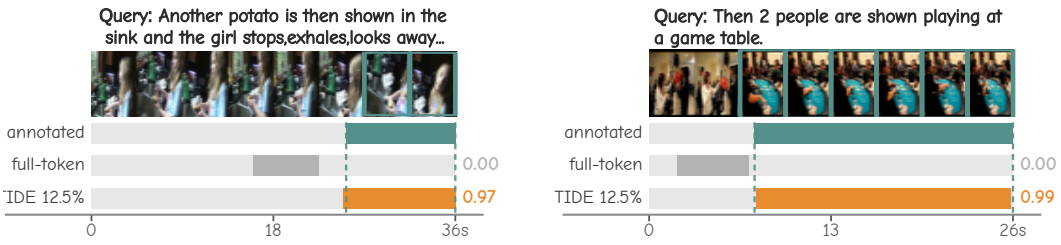

Figure 6: Grounding under 12.5% retention, continued. On short videos the full-token baseline collapses onto a narrow early window, while TIDE preserves the extent as well as the position of the annotated event.

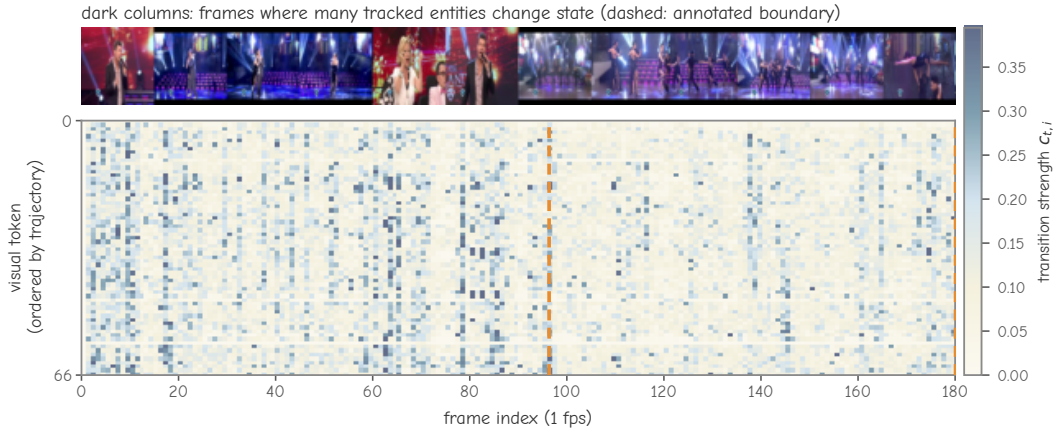

Figure 7: Transition field $c_{t,i}$ (TSF). Rows are visual tokens ordered by trajectory identity, columns are frames. Dark columns mark frames at which many tracked entities diverge from their matched predecessor. The dashed line marks the annotated event boundary.

A.2 Temporal State Field

Figure 7 shows the transition field $c_{t,i}$ defined in Eq. 4, with tokens ordered by trajectory identity. Dark columns indicate simultaneous changes across many matched tokens. This field enters EDA through the transition mass g_t and ESS through the temporal discount $\gamma(1 - c_{t,i})_+$; thresholding it gives the discrete states in Eq. 5 used by BPF.

In the video shown in Figure 8, trajectory matching raises the annotated boundary frame's rank from 23 to 7 out of 180 frames. This comparison uses the per-frame aggregate $\bar{c}_t$ and naive inter-frame cosine drift computed on the same tokens. In this example, the naive signal's largest responses coincide with stage lighting changes and camera cuts, whereas trajectory matching follows displaced content across frames. EDA combines this transition signal with query relevance and spectral diversity to determine the final integer frame quota.

The two views connect the token-level computation to its use in allocation. A concentrated column in the transition field indicates changes across several correspondences, which contribute to the frame-level signal. Matching changes the reference against which each token is compared: displaced content is compared with its estimated predecessor instead of the patch at a fixed spatial position. EDA subsequently weights these changes by query relevance, distinguishing the measurement of visual change from its importance to the query.

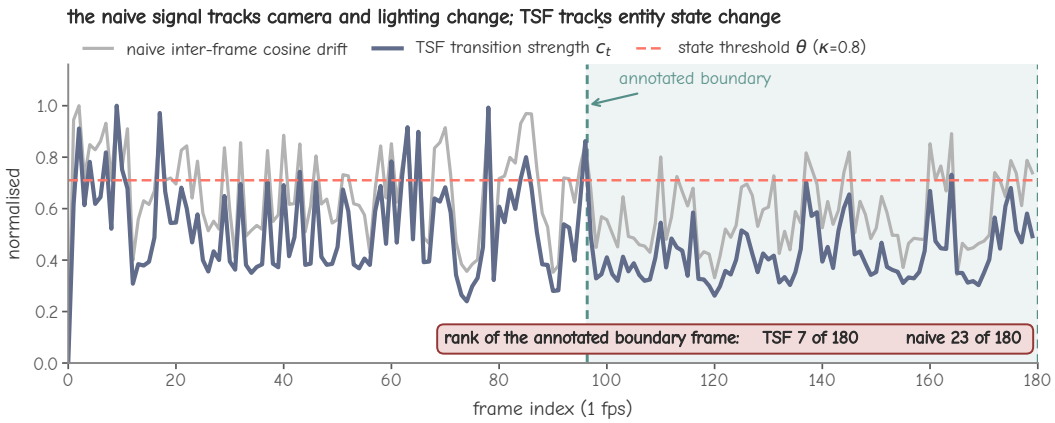

Figure 8: Trajectory-matched transition strength against naive frame differencing. Both signals are normalized to their maxima. The annotated boundary frame ranks 7 of 180 under the TSF signal $\bar{c}_t$ and 23 of 180 under naive inter-frame cosine drift, whose peaks instead track lighting and camera change.

A.3 Evidence Density Allocation

EDA redistributes each fixed unit token budget B_u among its frames. Figure 9 shows the difference between b_t and a uniform split of the same unit budget. Positive and negative deviations balance within each unit.

Figure 10 shows frame quotas around an annotated transition in the same video. Two-sided protection uses g_{t+1} when both frames belong to the same unit; adjacent units have independent token budgets. The right panel compares the per-unit correlation between frame quotas and transition mass for EDA and an effective-rank-only allocator, illustrating how the transition term changes local allocation.

The effective-rank-only comparison keeps the unit token budget fixed and changes the signal used to distribute it. It therefore examines where the available tokens are placed within the unit. The final bars reflect the combined allocation weights and integer rounding: a larger transition component competes with the representational diversity of other frames. Reading the allocation and transition panels together shows how a continuous importance signal becomes discrete frame quotas.

A.4 Evidence Skeleton Selection

Figure 11 shows how each frame quota is spent, comparing the ESS skeleton $\mathcal{A}_t$ against a saliency-only baseline that takes the top b_t tokens before temporal discounting. This baseline exposes the combined effect of temporal discounting and spatial diversity. On the frame outside the annotated event, saliency-only selection repeats all 7 previous patches, while ESS retains 40% and reallocates the remaining quota to new evidence. ESS therefore avoids repeatedly encoding persistent high-contrast background structure.

Figure 12 summarizes selection behavior using 15-frame rolling windows. Mean frame-to-frame reuse is 0.496 for saliency-only selection and 0.415 for ESS, with lower reuse under ESS in 157 of 166 windows. Mean transition mass rises from 0.107 to 0.128, with ESS higher in all 166 windows. Together, temporal discounting and spatial diversity shift selection toward tokens with stronger state changes in this video.

Both selectors receive the same frame quotas, so the comparison concerns selection within those quotas. The baseline ranks tokens before the temporal discount, whereas ESS also considers similarity to anchors already chosen in the current frame. The two plotted statistics capture different aspects of selection: reuse measures persistence across frames, and selected transition mass mea-

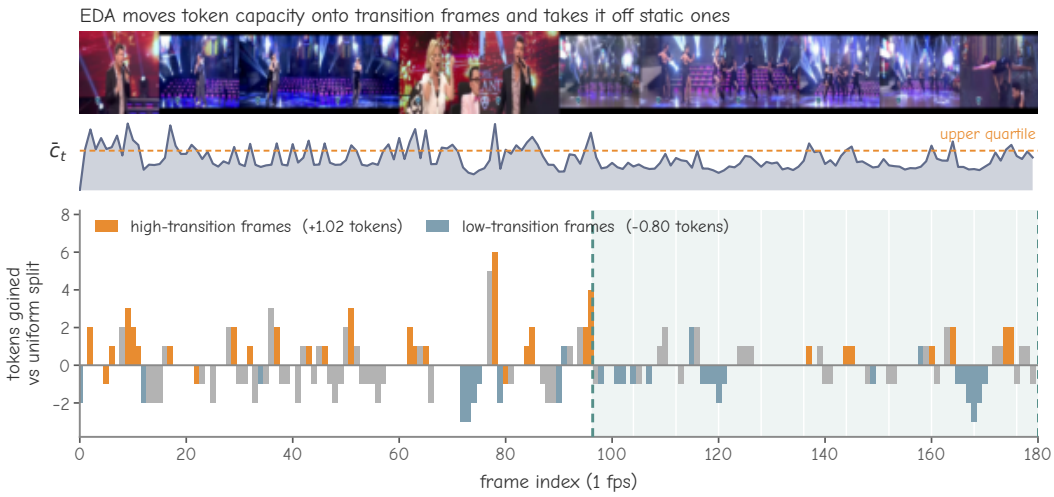

Figure 9: Within-unit reallocation (EDA), single-video illustration. Bars show b_t relative to a uniform split of the same unit budget; the middle row shows $\bar{c}_t$. Positive and negative deviations balance within each unit because its total budget is fixed.

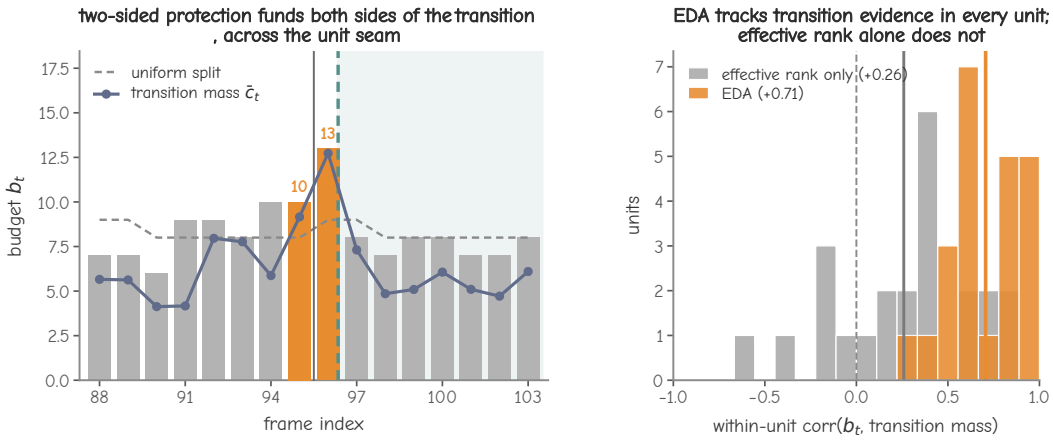

Figure 10: Local allocation behavior (EDA), single-video illustration. Left: frame quotas around an annotated transition, spanning two independently budgeted units. Right: per-unit correlation between b_t and transition mass for EDA and an effective-rank-only allocator. These panels illustrate the allocation rule on one selected video.

sures the amount of changing content retained. Their joint behavior connects reduced repetition to the temporal evidence carried by the selected tokens.

A.5 Boundary-Preserving Fusion

Figure 13 shows the transport weights from Eq. 10 for the unit beginning at the annotated boundary. Non-anchor sources appear on the left and selected anchors on the right; edge width is proportional to ω_{xa}, and color marks cross-state routing. The figure displays the sources with the highest boundary evidence from a plan containing 2766 edges. For these sources, the soft penalty in Eq. 9 reduces cross-state transported mass from 40.5% under top-k cosine merging to 25.5%. We reconstruct the plan from the recorded fusion inputs. Across all 23 units, the reconstructed fused embeddings have a minimum cosine similarity of 0.9914 to the implementation outputs.



Figure 11: Spatial anchor selection (ESS). Selected patches are outlined on the 6×11 token display grid, tinted patches are discarded. Left column: top-b_t by the undiscounted selection score. Right column: the ESS skeleton. On the frame outside the annotated event the saliency-only arm re-selects its entire previous selection, while ESS turns over 60% of it.



Figure 12: Redundancy and evidence capture (ESS). Rolling means over 15 frames. Left: overlap between consecutive frames' selections, 0.415 for ESS against 0.496 for saliency-only, lower across 157 of 166 windows within the analyzed video. Right: transition mass of the selected tokens, 0.128 against 0.107, higher across all 166 windows. The shaded region marks the annotated event.

Figure 14 reports routing statistics by whether the source carries boundary evidence. The penalty is $\alpha\, e_x\, \mathbf{1}[\cdot]$, with $e_x^{\text{raw}} = (\hat{q}_x - \bar{\hat{q}}_u)_+ (\hat{c}_x - \bar{\hat{c}}_u)_+$ min–max normalized within each unit. The gate is active for the 17.2% of tokens that exceed the unit means in both query relevance and transition strength. Averaged across units, BPF reduces cross-state transported mass for these sources from 45.9% to 36.3% relative to top-k cosine merging, a 20.8% relative reduction. Cross-state mass is lower in all 23 units. The soft penalty favors compatible states while retaining local fallback routes, and the trajectory reward favors continuity along matched tokens.

Separating sources by evidence makes the effect of the gate explicit. A source with zero boundary evidence receives no state penalty, so its routing is governed by similarity, temporal proximity, and the trajectory reward. For a source with positive evidence, the additional penalty changes the relative preference for cross-state destinations. The graph shows this redistribution for selected sources, while the per-unit plot shows its consistency within the video. These routing measurements describe the operation tested by the BPF component ablations in Table 4.

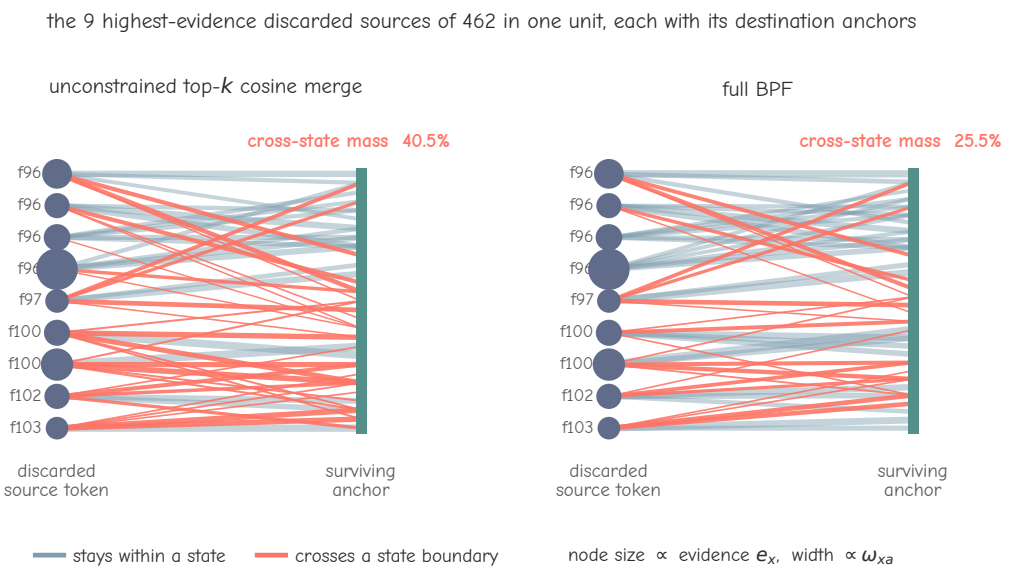

Figure 13: Soft transport plan (BPF). Discarded sources on the left, surviving anchors on the right, node size scales with query-gated boundary evidence e_x and edge width with the transport weight ω_{xa}. Red edges cross a trajectory-state boundary. The soft compatibility penalty reduces cross-state transported mass for these sources from 40.5% to 25.5%.

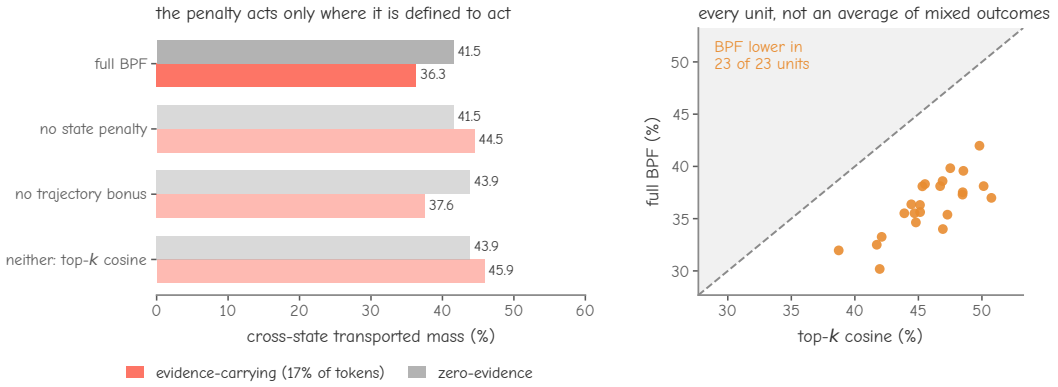

Figure 14: Ablation of BPF routing terms. Left: cross-state transported mass for each variant, averaged over the 23 units of this video and grouped by whether the source carries boundary evidence. The state penalty is proportional to e_x and acts on sources with positive evidence. Right: cross-state transported mass for these sources in each unit; full BPF reduces it in all 23 units.

B Extended Related Work

We review related VTG methods and compare how visual token compressors handle temporal evidence.

B.1 Video Temporal Grounding with LVLMs

Supervised grounding frameworks. Supervised VTG methods learn from temporal annotations on datasets such as Charades-STA (Sigurdsson et al., 2016) and ActivityNet-Captions (Caba Heilbron et al., 2015). UniVTG (Lin et al., 2023) unifies moment retrieval, highlight detection, and video summarization through pretraining on heterogeneous video-text data. TimeLens (Zhang et al.,

Table 7: Design comparison of visual token compressors. TIDE uses token correspondences to connect query-conditioned allocation with state-aware context fusion.

Method	Target Domain	Training-Free	Query-Conditioned Budget	Token Trajectory Tracking	Fusion Strategy	Temporal Evidence Treatment
VisionZip (Yang et al., 2025)	General VLM / QA	✓	✗	✗	Spatial compression	Attention-based selection
FastVID (Shen et al., 2025)	Video LLMs / QA	✓	✗	✗	None (Dynamic Pruning)	Density-based pruning
FlashVID (Fan et al., 2026)	Video LLMs / QA	✓	✗	✗	Tree-based Merging	Selection and merging
SemVID (Li et al., 2026a)	Video Grounding	✓	✓	✗	None (Taxonomy Pruning)	Semantic evidence pruning
TIDE	Video Grounding	✓	✓	✓	Boundary-Preserving Fusion	**State-aware allocation and fusion**

2026) uses reinforcement learning with verifiable rewards to improve temporal predictions, while UniTime (Li et al., 2026b) interleaves timestamp and visual tokens and adapts the frame scale to the video context. These approaches learn temporal localization through task-specific training.

Training-free and zero-shot approaches. Recent training-free VTG methods use frozen vision-language models with proposal scoring or hierarchical temporal decomposition. TFVTG (Zheng et al., 2024) scores dense candidate proposals, Luo et al. (Luo et al., 2024) use bottom-up proposal generation with conditional feature refinement, and HiTeA (Xu et al., 2026) aligns queries at event, scene, and action levels. These methods retain dense temporal sampling to preserve timestamp resolution. Since each frame contributes hundreds of visual tokens, the sequence can reach tens of thousands of tokens and make quadratic self-attention a substantial deployment cost. Visual token compression addresses this cost before LLM prefilling.

B.2 Design Comparison with Baseline Token Compressors

Table 7 compares spatial token compressors, general video compressors, and methods designed for VTG in terms of budgeting, temporal tracking, and fusion.

2D spatial token compression. Image-oriented approaches include ToMe (Bolya et al., 2023), FastV (Chen et al., 2024), and PyramidDrop (Xing et al., 2024). Their formulations emphasize spatial redundancy or visual token importance, whereas VTG also requires temporal positions and transitions for interval prediction.

VisionZip: attention-based visual compression (Yang et al., 2025). VisionZip uses visual attention to select a compact representation within each frame. Such scores describe which image regions receive attention, while VTG also requires deciding how to distribute capacity over time. TIDE retains visual saliency as a selection signal and adds query-conditioned frame allocation and a discount for temporally repeated content. This separates the amount of capacity assigned to a frame from the choice of tokens within it.

FastVID: dynamic temporal compression (Shen et al., 2025). FastVID uses dynamic temporal segmentation and density-based spatiotemporal pruning. The relevant distinction is how temporal change enters the compression decision. Frame-level variation can combine changes in object state, camera position, and scene composition. TIDE compares tokens with matched predecessors, then weights their transition scores by query relevance. This gives allocation a local correspondence reference and a connection to the event described by the query.

FlashVID: spatiotemporal selection and merging (Fan et al., 2026). FlashVID combines attention- and diversity-based selection with tree-based spatiotemporal merging. Selection and merging both contribute to its compressed representation. TIDE introduces an additional criterion for fusion: the compatibility of the temporal states associated with a source and an anchor. Query-relevant transition tokens receive a penalty for cross-state routing even when their appearance is similar. This design addresses the case in which similar visual content occurs on both sides of an event boundary.

SemVID: semantic evidence chains (Li et al., 2026a). SemVID partitions tokens into Object, Motion, and Context roles and uses query relevance and temporal variation to set per-frame token budgets. Both methods make compression specific to the grounding query. TIDE organizes its decisions around correspondence-based transitions and states: the transitions inform allocation and

selection, and the states constrain context recovery after anchors have been chosen. Soft fusion allows non-anchor tokens to contribute to the output through compatible destinations while preserving the prescribed token count.

B.3 Interpreting Comparisons with Full-Token Inference

Full-token inference provides a reference for the same frozen backbone, sampled video, prompt, and decoding configuration. TIDE changes the representation supplied to that backbone in two ways: selection changes which token positions are retained, and BPF updates the retained embeddings with weighted peripheral context. The comparison therefore evaluates a complete compressed representation, including its recovered context.

On ActivityNet-Captions, Qwen3-VL-8B reaches 42.75 mIoU at 12.5% retention, compared with 41.76 for full-token inference. At 25% retention, Qwen3-VL-4B reaches 43.74 versus 42.93. These settings show that retaining every visual token is not a necessary condition for the strongest grounding result from a frozen model. The ablations examine how TIDE constructs the compressed input: removing BPF tests context recovery after anchor selection, while the TSF and routing ablations examine the temporal signals used to organize that representation. Together, the comparisons connect the accuracy of the final sequence to the design of its allocation and fusion stages.

C Mathematical Formulations and Computational Complexity

This section formulates the compression objective and analyzes the computational scaling of TIDE.

C.1 Compression Objective

We consider VTG with a frozen LVLM whose parameters are Θ. Given dense visual embeddings $\mathbf{H} \in \mathbb{R}^{N_v \times d}$ and retained prompt and formatting embeddings $\mathbf{E} \in \mathbb{R}^{N_t \times d}$, uncompressed grounding decodes an interval using the frozen model. Query-relevance scoring uses the prompt subset $\{\mathbf{e}_v : v \in \mathcal{T}\}$ defined in Eq. equation 1:

$$(\hat{t}_s, \hat{t}_e) = \arg\max_{t_s, t_e} \log P_\Theta(t_s, t_e \mid \mathbf{H}, \mathbf{E}). \tag{12}$$

A training-free token compressor is defined as a mapping $\Phi : \mathbb{R}^{N_v \times d} \times \mathbb{R}^{N_t \times d} \to \mathbb{R}^{K \times d}$ with no learned parameters. TIDE realizes a target retention ratio $r \in (0, 1]$ through the integer budget $K = \sum_u B_u \approx r N_v$ defined in Section 4. The task objective is to find a compact representation $\tilde{\mathbf{H}} = \Phi(\mathbf{H}, \mathbf{E}, r)$ that supports accurate interval prediction. An idealized formulation expresses this goal as maximizing the likelihood of annotated boundaries under a token budget:

$$\max_{\Phi} \mathbb{E}_{(V,q,t_s,t_e)} \left[\log P_\Theta \left(t_s, t_e \mid \Phi(\mathbf{H}, \mathbf{E}, r), \mathbf{E}\right)\right] \quad \text{s.t.} \quad |\Phi(\mathbf{H}, \mathbf{E}, r)| = \sum_u B_u. \tag{13}$$

The expression above characterizes the desired compressed representation; TIDE does not optimize this likelihood objective. It constructs Φ through fixed allocation, selection, and fusion rules using transition strength, spectral diversity, and query relevance available at inference time. The LVLM remains frozen, and annotated intervals are used to evaluate predictions, not to construct the compressed representation.

C.2 Computational and Memory Complexity

We analyze LVLM prefilling for a standard Transformer with full multi-head attention. This dense-attention model specifies the assumptions used in the following compute and memory estimates. Let L_{LM} denote the number of Transformer layers, d the hidden dimension, and m the intermediate dimension of the gated feed-forward network (FFN). The full and compressed input lengths are $N = N_v + N_t$ and $N' = K + N_t$, respectively, with $K = \sum_u B_u$. Here N_t includes retained text and formatting positions; L continues to denote the temporal unit length.

Prefilling compute. For a sequence of length n, the query, key, value, and output projections require $8nd^2$ FLOPs per layer. The two attention matrix products require $4n^2d$ FLOPs, and the gated FFN requires $6ndm$ FLOPs. Counting each multiply-add as two FLOPs gives

$$F_{\text{prefill}}(n) = L_{\text{LM}}\left(4n^2d + 8nd^2 + 6ndm\right), \qquad n \in \{N, N'\}. \tag{14}$$

The quadratic attention term scales by $(N'/N)^2$, while projection and FFN costs scale by N'/N. When $N_v \gg N_t$ and $K \approx rN_v$, these factors approach r^2 and r. At $r = 12.5\%$, the visual-dominant approximation gives $1/64$ of the original attention arithmetic and $1/8$ of the linear arithmetic. Including $83.9\,\text{ms}$ of compression overhead, the measured speedup for compression plus LLM prefilling is $4.7\times$ on Qwen3-VL-4B (Table 6).

Key-value cache memory. Prefilling stores key and value states for subsequent autoregressive generation. For batch size b and key-value width d_{kv} per layer, the prefill cache size is

$$M_{\text{KV}}(n) = 2bL_{\text{LM}}nd_{kv}\,\text{sizeof}(\text{dtype}), \qquad n \in \{N, N'\}. \tag{15}$$

Thus $M_{\text{KV}}(N')/M_{\text{KV}}(N) = N'/N$. The visual-token contribution falls approximately eightfold at $r = 12.5\%$; total peak memory also includes model weights and temporary buffers.

Module-wise FLOPs. We analyze the selector given projected visual embeddings, saliency scores, and prompt embeddings. Let $N_v = TM$, with T frames and M tokens per frame. The reference configuration uses $L = 8$, $K_a = 8$, $W = 2$, and $k = 4$. The following estimates track feature arithmetic and spectral decomposition; measured compressor latency is reported separately in Table 6.

Temporal State Field (TSF). TSF computes adjacent-frame affinities once for message passing and again for matching contextualized features. These two passes require approximately $4(T-1)M^2d$ FLOPs. Forward and backward top-K_a aggregation adds approximately $4(T-1)MK_ad$ FLOPs. Feature normalization, contextual averaging, and matched transition evaluation contribute linear feature arithmetic:

$$F_{\text{TSF}} \approx 4(T-1)M^2d + 4(T-1)MK_ad + \mathcal{O}(N_vd). \tag{16}$$

Evidence Density Allocation (EDA). Query relevance compares N_v visual tokens with $|\mathcal{T}| \le N_t$ prompt tokens, requiring $2N_v|\mathcal{T}|d \le 2N_vN_td$ FLOPs. We use the latter bound below. Computing effective rank requires the singular values of the $M \times d$ frame matrix. Constructing its $M \times M$ Gram matrix costs approximately $2M^2d$ FLOPs, and spectral decomposition adds $\mathcal{O}(M^3)$ per frame. Across all T frames:

$$\text{FLOPs}_{\text{rank}} \;=\; 2TM^2d + \mathcal{O}(TM^3) \;=\; 2N_vMd + \mathcal{O}(TM^3). \tag{17}$$

Computing query-conditioned transition mass g_t, applying two-sided protection $\tilde{g}_t = \max(g_t, g_{t+1})$, and assigning integer frame quotas add $\mathcal{O}(N_v)$ scalar operations for fixed unit length. Including the Gram-matrix spectra, the EDA estimate is

$$F_{\text{EDA}} \approx 2N_vN_td + 2N_vMd + \mathcal{O}(TM^3 + N_v). \tag{18}$$

Evidence Skeleton Selection (ESS). The batched implementation precomputes each frame's $M \times M$ cosine matrix, then updates maximum redundancy as anchors are selected. With $b_{\max} = \max_t b_t$, its arithmetic is

$$F_{\text{ESS}} \approx 2N_vMd + \mathcal{O}(N_vb_{\max}). \tag{19}$$

Computing similarities on demand instead would cost $\mathcal{O}(KMd)$. Precomputation batches these products into larger matrix operations while implementing the same selection rule.

Boundary-Preserving Fusion (BPF). Unit u contains $S_u = |\mathcal{U}_u|M - B_u$ non-anchor sources and B_u anchors. The vectorized implementation computes their full $S_u \times B_u$ similarity matrix before applying the temporal mask. Each admitted source then contributes to at most k anchors. The feature arithmetic is therefore

$$F_{\text{BPF}} \approx 2d\sum_u S_uB_u + \mathcal{O}\big(k(N_v - K)d\big). \tag{20}$$

Table 8: Leading computational costs given visual features and saliency. $N = N_v + N_t$ and $N' = K + N_t$ are the full and compressed sequence lengths; L_{LM} is the layer count. Prefill costs assume full attention and FFN width proportional to d. TIDE uses batched similarities and fixed unit length L; $S_u = |\mathcal{U}_u|M - B_u$.

Method	Temporal Budgeting	Token Selection	Context Fusion	Compressor Overhead	Post-Compression Prefill
Uncompressed (Full)	–	–	–	0	$\mathcal{O}(L_{\mathrm{LM}}N^2d + L_{\mathrm{LM}}Nd^2)$
VisionZip (Yang et al., 2025)	$\mathcal{O}(1)$	$\mathcal{O}(N_v \log M)$	None	$\mathcal{O}(N_v \log M)$	$\mathcal{O}(L_{\mathrm{LM}}(N')^2d + L_{\mathrm{LM}}N'd^2)$
FastVID (Shen et al., 2025)	$\mathcal{O}(Td)$	$\mathcal{O}(N_vMd)$	None	$\mathcal{O}(N_vMd)$	$\mathcal{O}(L_{\mathrm{LM}}(N')^2d + L_{\mathrm{LM}}N'd^2)$
FlashVID (Fan et al., 2026)	$\mathcal{O}(1)$	$\mathcal{O}(N_vMd)$	$\mathcal{O}(N_vMd)$	$\mathcal{O}(N_vMd)$	$\mathcal{O}(L_{\mathrm{LM}}(N')^2d + L_{\mathrm{LM}}N'd^2)$
SemVID (Li et al., 2026a)	$\mathcal{O}(N_vN_td + Td)$	$\mathcal{O}(N_v \log M)$	None	$\mathcal{O}(N_vN_td + N_v \log M)$	$\mathcal{O}(L_{\mathrm{LM}}(N')^2d + L_{\mathrm{LM}}N'd^2)$
TIDE (Ours)	$\mathcal{O}(N_vMd + TM^3 + N_vN_td)$	$\mathcal{O}(N_vMd)$	$\mathcal{O}(d\sum_u S_uB_u + k(N_v - K)d)$	$\mathcal{O}(N_vMd + TM^3 + N_vN_td)$	$\mathcal{O}(L_{\mathrm{LM}}(N')^2d + L_{\mathrm{LM}}N'd^2)$

The window W bounds admissible routes, while the unit length L bounds the matrix computation. Ignoring integer rounding and a shorter final unit, the leading term is approximately $2Lr(1-r)N_vMd$.

Combining these terms, with $K_a \le M$ and $b_{\max} \le M$, gives

$$F_{\mathrm{TIDE}} = \mathcal{O}\left(N_vMd + TM^3 + N_vN_td + d\sum_u S_uB_u + k(N_v - K)d \right). \tag{21}$$

Scaling with video dimensions. Since $B_u \le LM$ and $\sum_u S_u = N_v - K$, the fusion term satisfies $\sum_u S_uB_u \le LM(N_v - K)$. For fixed spatial token count M, prompt length N_t, feature dimension d, unit length L, and remaining hyperparameters, the arithmetic cost of all four stages is asymptotically $\mathcal{O}(T)$. This bound describes computation; memory for full-video embeddings, trajectory metadata, and batched intermediate tensors also grows with T. Pairwise feature operations are quadratic in M, while the spectral decomposition contributes the cubic term shown above.

The spatial token count M depends on the backbone and input resolution. With $L = 8$, each unit-level operation processes at most eight frames, while the implementation retains full-video embeddings and trajectory information.

Linear compression cost and quadratic attention. Dense LLM self-attention requires $4L_{\mathrm{LM}}N^2d$ FLOPs under the model above. For fixed per-frame dimensions, prompt length, and hyperparameters, Eq. equation 21 implies

$$\frac{F_{\mathrm{TIDE}}}{4L_{\mathrm{LM}}N^2d} = \mathcal{O}\left(\frac{N_v}{L_{\mathrm{LM}}(N_v + N_t)^2} \right). \tag{22}$$

This explains why local compression can reduce the cost of processing dense video inputs: its arithmetic grows linearly with the number of frames, while full attention grows quadratically with the complete input length. Table 6 measures the resulting latency with compression included.

Asymptotic comparison with baselines. Table 8 summarizes leading-order costs under the stated token-grid assumptions. The comparison separates the overheads of selection, temporal processing, and context fusion, while Table 6 reports measured compression and prefilling latency.

D Method Details and Implementation

This section expands the methodology in Section 4, provides pseudocode, explains the design choices, and reports module-level profiling.

D.1 Module Interfaces

TIDE's stages exchange temporal signals, integer frame quotas, and anchor identities. TSF produces a continuous transition score for each token together with discrete trajectory and state identities. These quantities serve different purposes: transition magnitudes determine how much a representation has changed, trajectory identities link correspondences over time, and state identities identify the feature phases inherited along those correspondences. Computing them over the full video gives all temporal units the same reference.

EDA converts these token-level measurements into frame quotas $\{b_t\}$. Effective rank describes the diversity within a frame, average trajectory change summarizes its temporal variation, and query-weighted transition mass emphasizes changes associated with the prompt. Normalization by unit means puts these signals on comparable relative scales before they are combined. Largest-remainder rounding then converts the continuous weights into integer counts that sum to the unit budget. The implementation caps each count at the number of available tokens and distributes the remaining capacity among frames with spare capacity.

ESS uses the quotas to select anchor indices in each frame. Its score includes the predecessor discount $\gamma\rho_{t,i}$, with zero discount in the first frame, while MMR compares contextualized candidate embeddings with anchors already selected. EDA determines how many tokens a frame contributes; ESS determines which representations occupy those positions. The selected indices retain their frame and spatial identities for subsequent routing and sequence assembly.

BPF receives the selected anchors, non-anchor sources, and the shared trajectory-state reference. It forms candidate destinations within a temporal window ($W = 2$), applies the evidence gate e_x and cross-state penalty αe_x, and accumulates accepted source contributions into the anchors. Fusion changes anchor embeddings without adding output positions, so the frame and unit token counts established by EDA and ESS remain fixed. The updated anchors are returned in their original sequence order for LLM prefilling.

Reference hyperparameter configuration. The reference configuration uses temporal units of $L = 8$ frames, $K_a = 8$ message neighbors, and $k = 4$ fusion targets. Allocation combines spectral diversity and trajectory change with $\beta_1 = 0.4$, then incorporates query-conditioned transitions with $\beta_2 = 0.8$. Table 11 collects all defaults; the gray rows in Tables 13–15 identify the reference values for the parameters varied in the sensitivity experiments.

D.2 Algorithmic Pseudocode

Algorithms 1 and 2 describe TIDE's computation. TSF runs across the full video to establish continuous correspondences and states. EDA, ESS, and BPF then use this reference within each temporal unit.

D.3 Architectural Design Rationale

The following design choices connect temporal evidence to token allocation, selection, and fusion.

Correspondence-based transition scores. Comparing features at fixed spatial positions can respond to camera translation because the same position observes different content in adjacent frames. TSF first aggregates neighboring correspondence information, then compares each contextualized token with its matched predecessor. Stable content can therefore receive a low transition score after moving to another patch, while a change in the matched representation produces a stronger response. Figure 8 illustrates the resulting change in the ranking of the annotated boundary relative to frame-to-frame drift.

Greedy matching assigns each token its highest-affinity predecessor independently, allowing several tokens to inherit the same predecessor. The resulting trajectories describe local feature correspondences, including branching across patches. State identities mark relatively stable portions of those correspondences, with the video-level quantile adapting the transition threshold to the observed score distribution. EDA consumes frame-level aggregates of the continuous scores, ESS uses their token-level values, and BPF uses the inherited identities. This makes the role of a correspondence explicit at each stage, including cases in which newly visible content receives a large change score.

Two-sided allocation. An event boundary involves a contrast between observations before and after a transition. A large score at a frame describes change relative to its predecessor, so the preceding observation also helps represent that contrast. Within a unit, $\tilde{g}_t = \max(g_t, g_{t+1})$ gives each frame access to the next frame's transition signal. An internal transition can thus influence both adjacent allocation weights. Taking a maximum shares the signal without adding neighboring responses together, which would amplify clusters of large changes.

Algorithm 1 TIDE Complete End-to-End Compression Pipeline

Require: Video embeddings $\mathbf{H} \in \mathbb{R}^{T\times M\times d}$, saliency $\mathbf{S} \in \mathbb{R}^{T\times M}$, prompt embeddings $\mathbf{E}$, retention ratio r, and settings from Table 11.
Ensure: Compressed token sequence $\mathbf{H}^{\text{out}} \in \mathbb{R}^{K\times d}$ where $K = \sum_u B_u \approx rN_v$.
Compute raw query relevance q from $\mathbf{H}$ and $\mathbf{E}$ using Eq. equation 1.
Initialize $\mathbf{Z} \leftarrow \mathbf{H}$, $c_{1,i} \leftarrow 0$, $\rho_{1,i} \leftarrow 0$, and $\mathrm{tr}(1,i) = \mathrm{st}(1,i) \leftarrow i$.
if $T \geq 2$ **then**
    Compute adjacent-frame affinities and bidirectional top-$\min(K_a, M)$ messages.
    Form $\mathbf{z}_{t,i} \leftarrow (\mathbf{h}_{t,i} + \bar{\mathbf{m}}_{t,i})/2$ using the endpoint rules in Appendix E.3.
    Recompute affinities from $\mathbf{Z}$ and greedily link predecessors and trajectory identities.
    Compute $c_{t,i}$ using Eq. equation 4 and $\rho_{t,i} \leftarrow (1 - c_{t,i})_+$ for $t \geq 2$.
    Set $\theta \leftarrow \mathrm{Quantile}_\kappa(\{c_{t,i}\}_{t\geq 2})$ and assign states using Eq. equation 5.
end if
Partition video frames into disjoint temporal units $\{\mathcal{U}_u\}_{u=1}^{\lceil T/L\rceil}$ of size $\leq L$.
for each temporal unit $\mathcal{U}_u$ **do**
    Compute unit token budget: $B_u \leftarrow \mathrm{clip}\left(\lfloor r|\mathcal{U}_u|M\rceil, 1, |\mathcal{U}_u|M\right)$.
    Normalize query relevance over the unit: $\hat{q} \leftarrow \mathcal{N}_u(q)$.
    for each frame $t \in \mathcal{U}_u$ **do**
        Compute r_t from $\mathbf{Z}_t$, $c_t \leftarrow M^{-1}\sum_i c_{t,i}$, and $g_t \leftarrow \sum_i \hat{q}_{t,i}c_{t,i}/\max(\sum_i \hat{q}_{t,i}, \epsilon)$.
    end for
    for each frame $t \in \mathcal{U}_u$ **do**
        Apply two-sided protection $\tilde{c}_t \leftarrow \max(c_t, c_{t+1})$ and $\tilde{g}_t \leftarrow \max(g_t, g_{t+1})$ only when $t+1 \in \mathcal{U}_u$, otherwise use $\tilde{c}_t \leftarrow c_t$ and $\tilde{g}_t \leftarrow g_t$.
    end for
    Normalize each $x \in \{r, \tilde{c}, \tilde{g}\}$ by its unit mean: $\hat{x}_t \leftarrow x_t/\max(\bar{x}_u, \epsilon)$.
    Form $v_t \leftarrow (1-\beta_1)\hat{r}_t + \beta_1\hat{c}_t$ and $w_t \leftarrow (1-\beta_2)v_t + \beta_2\hat{g}_t$.
    Allocate integer quotas using Appendix E.3, ensuring $0 \leq b_t \leq M$ and $\sum_{t\in\mathcal{U}_u} b_t = B_u$.
    Apply Algorithm 2 to $\mathbf{Z}_{\mathcal{U}_u}$ and the shared scores, identities, and quotas.
end for
return Concatenate $\{\mathbf{H}_u^{\text{out}}\}_u$ across units in temporal sequence.

The final frame allocation combines this signal with spectral diversity and average trajectory change before integer rounding. The two adjacent frames can consequently receive different token counts while both retaining the transition's contribution to their allocation weights. Figure 10 shows this conversion from signals to quotas. At a unit seam, each unit distributes its own budget using its internal signals; the global TSF reference is retained, while token capacity stays local.

Temporal discounting in ESS. MMR discourages selecting similar embeddings within a frame. The temporal discount addresses a separate source of redundancy: a salient patch can remain nearly unchanged across successive frames and repeatedly occupy the budget. Discounting its score relative to a matched predecessor gives changing content more opportunity to enter the anchor set. The relevance score and the within-frame diversity term then determine the final selection jointly.

If every token in a frame has the same transition score, the discount is constant across candidates and leaves their ordering unchanged. ESS therefore adjusts selection according to differences in temporal change, without requiring a fixed fraction of new anchors at every frame. Appendix A.4 compares the combined discount-and-diversity rule with undiscounted selection, and Table 15 examines the weight assigned to the discount.

Soft compatibility in BPF. BPF uses the soft penalty $-\alpha e_x$ to favor state-compatible anchors, with stronger penalties for sources carrying query-relevant transition evidence. Consider a source representing the state before a transition when the anchors available in its temporal window all represent the state after it. A hard state constraint would leave the source with no admissible destination. A soft penalty instead lowers the affinity of those destinations, retaining a route if the best penalized similarity meets the source-admission threshold τ_m. This provides a way to recover context when a state has few surviving anchors.

Algorithm 2 Evidence Skeleton Selection (ESS) and Boundary-Preserving Fusion (BPF)

Require: Contextualized unit embeddings $\mathbf{Z}_{\mathcal{U}_u}$, scores s, q, c, ρ, trajectory and state identities, frame quotas $\{b_t\}$, and settings $(\gamma, \lambda, W, \alpha, \lambda_m, \tau_m, \sigma, \eta, k)$.
Ensure: Updated anchor embeddings $\mathbf{H}_u^{\text{out}} \in \mathbb{R}^{B_u \times d}$.

Set $\mathcal{I}_u \leftarrow \{(t,i) : t \in \mathcal{U}_u, 1 \le i \le M\}$ and $\hat{s} \leftarrow \mathcal{N}_u(s)$, $\hat{q} \leftarrow \mathcal{N}_u(q)$, $\hat{c} \leftarrow \mathcal{N}_u(c)$.
Initialize unit anchor set $\mathcal{A}_u \leftarrow \emptyset$.
for each frame $t \in \mathcal{U}_u$ **do**
    Set $a_{t,i} \leftarrow \hat{s}_{t,i} - \gamma\rho_{t,i}$ and $\hat{s}_{t,i}^{\text{tmp}} \leftarrow \mathcal{N}_t(a)_{t,i}$.
    Initialize frame anchor set $\mathcal{A}_t \leftarrow \emptyset$.
    while $|\mathcal{A}_t| < b_t$ **do**
        Select $i^\star$ by Eq. equation 8 using $\mathbf{Z}_t$; use zero redundancy for the first pick.
        $\mathcal{A}_t \leftarrow \mathcal{A}_t \cup \{i^\star\}$
    end while
    $\mathcal{A}_u \leftarrow \mathcal{A}_u \cup \{(t,i) : i \in \mathcal{A}_t\}$.
end for
Set $\mathcal{X}_u \leftarrow \mathcal{I}_u \setminus \mathcal{A}_u$.
For every $x \in \mathcal{I}_u$, compute $e_x^{\text{raw}} \leftarrow (\hat{q}_x - \bar{\hat{q}}_u)_+(\hat{c}_x - \bar{\hat{c}}_u)_+$; set $e \leftarrow \mathcal{N}_u(e^{\text{raw}})$.
Initialize accumulators $\mathbf{S}_a \leftarrow \mathbf{z}_a$ and $W_a \leftarrow 1$ for every anchor a.
for each source token $x \in \mathcal{X}_u$ **do**
    $\mathcal{K}(x) \leftarrow \emptyset$; set $\mathcal{V}(x) \leftarrow \{a \in \mathcal{A}_u : |t_a - t_x| \le W\}$.
    if $\mathcal{V}(x) \neq \emptyset$ **then**
        Compute $\tilde{z}_{xa}$ for $a \in \mathcal{V}(x)$ using Eq. equation 9 and $\mathbf{Z}$.
        if $\max_{a \in \mathcal{V}(x)} \tilde{z}_{xa} \ge \tau_m$ **then**
            $k_x \leftarrow \min(k, |\mathcal{V}(x)|)$; select $\mathcal{K}(x) \leftarrow \text{top-}k_{x\,a \in \mathcal{V}(x)}\, \tilde{z}_{xa}$.
            Compute ψ_{xa} and ω_{xa} by Eq. equation 10, with the stabilized Gaussian and softmax in Appendix E.3.
            Set source importance $\nu_x \leftarrow 1/2 + \hat{s}_x$.
            for each $a \in \mathcal{K}(x)$ **do**
                $\mathbf{S}_a \leftarrow \mathbf{S}_a + \omega_{xa}\nu_x\mathbf{z}_x$; $W_a \leftarrow W_a + \omega_{xa}\nu_x$.
            end for
        end if
    end if
end for
for each anchor $a \in \mathcal{A}_u$ **do**
    $\mathbf{h}_a^{\text{out}} \leftarrow \mathbf{S}_a / W_a$.
end for
return $\mathbf{H}_u^{\text{out}} = \{\mathbf{h}_a^{\text{out}}\}_{a \in \mathcal{A}_u}$ arranged in temporal order.

The source-admission threshold controls whether a source has a sufficiently compatible destination. A source that fails this criterion contributes nothing to the fused sequence; an accepted source distributes its contribution over the selected candidate anchors. Equation 11 retains the contextualized anchor with unit base weight and normalizes the incoming contributions. The output keeps the selected anchor positions while incorporating context from other tokens.

Roles of the routing terms. The evidence gate combines query relevance and transition strength: a source receives a positive state penalty when both scores exceed their unit means. This targets changes relevant to the prompt while allowing other sources to contribute through similarity-based routing. The temporal window bounds the candidate set, the Gaussian prior favors nearer frames within that set, and the trajectory reward favors matched content. Trajectory continuity and state compatibility capture different relationships because a trajectory can pass through several states. The routing score combines these relationships when choosing how to distribute context. Table 4 compares BPF with frame-boundary fusion and separately tests the temporal prior and trajectory reward.

Table 9: TIDE module latency and memory use for a 64-frame video ($M = 144$ tokens/frame, $N_v = 9,216$, $d = 1152$) on an NVIDIA A800-SXM4 GPU. Appendix E.7 gives the hardware and software configuration.

Stage	Core Operation	Latency (ms)	GPU Mem (MB)
TSF	Pairwise spatial-semantic affinity	4.1	18.4
	Top-8 neighbor message passing	6.8	24.6
	Backward trajectory tracking	7.3	12.2
	TSF Subtotal	**18.2**	24.6
EDA	Batched frame-level spectrum	3.1	16.8
	Effective rank & transition mass	2.8	8.4
	Two-sided quota allocation	2.5	4.2
	EDA Subtotal	**8.4**	16.8
ESS	Predecessor similarity discount	1.4	6.2
	Vectorized MMR anchor selection	21.2	32.0
	ESS Subtotal	**22.6**	32.0
BPF	Boundary gating & window filtering	6.5	14.2
	Top-k soft similarity routing	14.8	28.5
	Importance-weighted scatter fusion	13.4	36.4
	BPF Subtotal	**34.7**	36.4
Total TIDE	**Total preprocessing overhead**	**83.9**	**48.2**

D.4 GPU Latency and Spectral Computation

Table 9 reports GPU wall-clock latency for every TIDE stage, including the singular value decomposition used by EDA. At $T = 64$, $M = 144$, and $d = 1152$, the measured subtotals are TSF 18.2 ms, EDA 8.4 ms, ESS 22.6 ms, and BPF 34.7 ms, for 83.9 ms in total. These fixed-shape measurements describe the module overhead at the listed dimensions; the model-level comparison is reported in Table 6.

Spectral computation. For a frame matrix $\mathbf{H}_t \in \mathbb{R}^{M\times d}$, the squared singular values are the eigenvalues of the $M \times M$ Gram matrix $\mathbf{H}_t\mathbf{H}_t^\top$. The implementation constructs these matrices in a batch, obtains their spectra with `torch.linalg.svdvals`, and uses `eigvalsh` as a fallback. It then takes square roots to compute effective rank. This avoids a direct decomposition of every $M \times d$ feature matrix. The reported spectral-stage latency is 3.1 ms for the fixed dimensions in Table 9; runtime depends on tensor dimensions and the numerical backend.

Scope of the speedup. The $4.7\times$ result in Table 6 measures visual-token compression plus LLM prefilling. Video decoding, vision encoding, and autoregressive generation are excluded consistently for every method. The module profile above accounts for the 83.9 ms compression component, while the reduced visual sequence lowers prefilling latency and the visual-token KV-cache contribution.

The compute analysis and the latency profile describe different aspects of this saving. Compression performs affinity calculations, spectral decomposition, discrete selection, and scatter accumulation, whose GPU costs depend on batching and memory access as well as arithmetic count. MMR also makes successive anchor choices, whereas similarity calculations can process many token pairs together. The stage timings show how these operations contribute to the measured compressor overhead. Adding that overhead to the reduced prefill time gives the latency of the compressed inference path used in the comparison.

E BENCHMARKS AND EXPERIMENTAL SETUP

We describe the benchmark splits, shared hyperparameters, and hardware and software used in the experiments.

E.1 BENCHMARKS

The two VTG benchmarks cover indoor activities and open-domain events. The two VideoQA benchmarks test whether the compressed representation also supports question answering across different video durations.

Charades-STA (Sigurdsson et al., 2016; Gao et al., 2017). Charades-STA is an established benchmark for sentence-level temporal activity localization, constructed on top of the Charades daily home activity dataset (Sigurdsson et al., 2016) by Gao et al. (Gao et al., 2017). The dataset consists of $6,670$ distinct indoor videos recorded by hundreds of users performing daily household routines (e.g., cooking, cleaning, reading, holding objects, and sitting on furniture). It contains $16,124$ natural language query-moment annotations in total, split into $12,404$ training pairs and $3,720$ test pairs from $1,334$ test videos. Videos in Charades-STA have an average duration of 30.59 seconds, with target ground-truth action intervals averaging 8.09 seconds ($\approx 26\%$ of total video length). Charades-STA contains fine-grained object interactions against relatively stable indoor backgrounds, providing a setting in which subtle state changes matter for localization.

ActivityNet-Captions (Caba Heilbron et al., 2015; Krishna et al., 2017). ActivityNet-Captions is a large-scale, open-domain benchmark for dense video event captioning and temporal grounding, introduced by Krishna et al. (Krishna et al., 2017) based on the ActivityNet dataset (Caba Heilbron et al., 2015). The dataset comprises $20,000$ untrimmed YouTube videos across 200 activity classes, including sports, cooking, social gatherings, and music. It contains $100,000$ temporally localized natural language sentences, with an average sentence length of 13.48 words and an average of 3.65 temporally annotated moments per video. Following standard convention in training-free VTG literature (Zheng et al., 2024; Li et al., 2026a), we evaluate on the standard validation split (`val_1`), comprising $4,926$ videos and $17,031$ query-moment pairs. ActivityNet-Captions videos are long-form, with an average length of 180.0 seconds (≈ 3.0 minutes) and many videos extending beyond 10 minutes. The average target moment duration is 36.0 seconds ($\approx 20\%$ of video duration). The benchmark therefore evaluates compression for temporally localized queries while retaining surrounding video context.

LongVideoBench (Wu et al., 2024). LongVideoBench (Wu et al., 2024) evaluates multimodal LLMs on video question answering across a range of durations. The entire benchmark comprises $3,763$ web-collected videos paired with $6,678$ human-annotated, four-option multiple-choice questions across 17 fine-grained categories. Following the standard evaluation setting in `lmms-eval`, we evaluate on the validation split comprising 753 videos and $1,337$ multiple-choice questions (as test set annotations are held-out on the evaluation server). Video durations range from 8 seconds to 1 hour ($3,600$ seconds), grouped into four tiers: $(8\,\text{s}, 15\,\text{s}]$, $(15\,\text{s}, 60\,\text{s}]$, $(180\,\text{s}, 600\,\text{s}]$ ($3 \sim 10$ minutes), and $(900\,\text{s}, 3600\,\text{s}]$ ($15 \sim 60$ minutes). The questions test visual details, event order, and character intentions, requiring evidence from different parts of a video.

Video-MME (Fu et al., 2025). Video-MME (Fu et al., 2025) evaluates video understanding across genres and durations. It contains 900 videos totaling approximately 256 hours, covering 6 domains (Knowledge, Film & Television, Sports Competition, Anime, Daily Life Record, and Multilingual) and 30 subfields. The benchmark includes $2,700$ multiple-choice questions with four options each. Video-MME is partitioned into three balanced duration tiers. Short videos contain fewer than two minutes of content and average $80.8\,\text{s}$; medium videos last 4–15 minutes and average $520.2\,\text{s}$; long videos last 30–60 minutes and average $2,471.0\,\text{s}$. Each tier contains 300 videos and 900 questions, allowing us to compare compression across progressively longer contexts.

E.2 BENCHMARK STATISTICS

Table 10 summarizes video durations, domains, tasks, and metrics for the four evaluation splits.

Table 10: Statistics for the evaluated benchmarks. Video and query counts correspond to the evaluation splits used in this study.

Benchmark	Task	Domain	Videos	Queries / QAs	Duration	Target Span	Split	Metric
Charades-STA (Gao et al., 2017)	Video Grounding	Indoor activities	1,334	3,720	30.6s	8.1s (26%)	Test	R@k, mIoU
ActivityNet-Captions (Krishna et al., 2017)	Video Grounding	Open-domain activities	4,926	17,031	180.0s	36.0s (20%)	Val-1	R@k, mIoU
LongVideoBench (Wu et al., 2024)	Long-Context QA	Movies, vlogs, docs	753	1,337	8s–3600s	Full video	Val	Top-1 Accuracy (%)
Video-MME (Fu et al., 2025)	Multi-Scale VideoQA	6 domains, 30 fields	900	2,700	81s / 520s / 2471s	Full video	Test	Accuracy (%) by tier

E.3 Implementation Conventions

Visual saliency and query scores. Let $\mathsf{A}_{t,h,a,b}$ be the final vision block's self-attention probability from query patch a to key patch b in head h, before spatial merging or pooling. For H_a heads and P patches per frame, the received-attention score and output-token saliency are

$$v_{t,b} = \frac{1}{H_a P} \sum_{h=1}^{H_a} \sum_{a=1}^{P} \mathsf{A}_{t,h,a,b}, \qquad s_{t,i} = [\mathcal{P}(v_t)]_i. \tag{23}$$

The pooling operator $\mathcal{P}$ follows the backbone's visual-token layout. Qwen averages the scores in each spatial merger group. LLaVA applies the same spatial pooling operator and stride used for its visual features. These scores measure attention received by patches; no CLS token is required. In the Qwen wrappers, attention probabilities are reconstructed from the final block's queries and keys after vision rotary encoding. Softmax is evaluated in float32, and probabilities are cast to the model dtype before averaging. Query scores are computed from the projected input embeddings before temporal message passing. Both raw signals remain attached to their original token indices throughout compression.

Normalization domains and stable arithmetic. For a finite collection of scalar scores indexed by $\mathcal{I}$, define $v_{\min} = \min_{j\in\mathcal{I}} v_j$, $v_{\max} = \max_{j\in\mathcal{I}} v_j$, and

$$\mathcal{N}_{\mathcal{I}}(v)_i = \begin{cases} 0, & |v_{\max} - v_{\min}| \le 10^{-8} + 10^{-5}|v_{\min}|, \\ \dfrac{v_i - v_{\min}}{v_{\max} - v_{\min}}, & \text{otherwise.} \end{cases} \tag{24}$$

This matches the constant-input check in the implementation. The notation $\mathcal{N}_u$ uses all tokens $\mathcal{I}_u = \{(t,i) : t \in \mathcal{U}_u, 1 \le i \le M\}$ in a unit, while $\mathcal{N}_t$ uses one frame. EDA normalizes query scores with $\mathcal{N}_u$. ESS computes $a_{t,i} = \mathcal{N}_u(s)_{t,i} - \gamma\rho_{t,i}$, then uses $\mathcal{N}_t(a)$ in MMR. BPF independently uses $\mathcal{N}_u(q), \mathcal{N}_u(c)$, and $\mathcal{N}_u(s)$ for the evidence gate and source importance; it does not use the discounted ESS score.

Every mean or sum in EDA's weight normalization is lower-bounded by $\epsilon = 10^{-6}$. Thus zero transition mass contributes zero, and the remaining rank term supplies the allocation weights under the reference configuration. Effective rank uses nonnegative Gram-matrix spectra, a singular-value sum lower-bounded by ϵ, and $\log(\bar{\sigma}_j + 10^{-12})$ in the entropy. Vector normalization for cosine similarity bounds each vector norm below by 10^{-12}. The reference implementation sanitizes nonfinite input scores and features before allocation and selection.

Bidirectional messages and feature flow. Let $\mathcal{N}^+_{t,i}$ contain the top-$\min(K_a, M)$ entries of row i of $A^{(t)}$, and let $\mathcal{N}^-_{t,i}$ contain those of column i of $A^{(t-1)}$. Each direction has its own softmax normalization:

$$\begin{aligned} \mathbf{m}^+_{t,i} &= \sum_{j\in\mathcal{N}^+_{t,i}} \frac{\exp(A^{(t)}_{ij}/\tau_a)}{\sum_{j'\in\mathcal{N}^+_{t,i}} \exp(A^{(t)}_{ij'}/\tau_a)} \mathbf{h}_{t+1,j}, \\ \mathbf{m}^-_{t,i} &= \sum_{j\in\mathcal{N}^-_{t,i}} \frac{\exp(A^{(t-1)}_{ji}/\tau_a)}{\sum_{j'\in\mathcal{N}^-_{t,i}} \exp(A^{(t-1)}_{j'i}/\tau_a)} \mathbf{h}_{t-1,j}. \end{aligned} \tag{25}$$

For $T \geq 2$, the contextual message is

$$\bar{\mathbf{m}}_{t,i} = \begin{cases} \mathbf{m}^{+}_{1,i}, & t = 1, \\ (\mathbf{m}^{-}_{t,i} + \mathbf{m}^{+}_{t,i})/2, & 1 < t < T, \\ \mathbf{m}^{-}_{T,i}, & t = T, \end{cases} \qquad \mathbf{z}_{t,i} = \frac{\mathbf{h}_{t,i} + \bar{\mathbf{m}}_{t,i}}{2}. \tag{26}$$

The one-hop messages use the original features of both neighboring frames. Greedy trajectory matching then recomputes affinities from $\mathbf{z}$. EDA's spectra, ESS's feature similarities, and BPF's routing and accumulation also use $\mathbf{z}$, while saliency and query scores retain the values computed before message passing. With message passing disabled, $\mathbf{z} = \mathbf{h}$.

The selector constructs its spatial index grid from M: it sets $h_g = \lfloor \sqrt{M} \rfloor$ and $w_g = \lfloor M/h_g \rfloor$ when $h_g w_g = M$, and otherwise uses an $M \times 1$ grid. Row and column indices are normalized by $\max(h_g - 1, 1)$ and $\max(w_g - 1, 1)$. At $M = 66$, both message passing and trajectory matching therefore use $\mathbf{p}_i = ((i-1)/65, 0)$. These coordinates provide a token-index proximity prior; the 6×11 display layout and the backbone's native positional indices serve separate purposes. Message passing divides spatial distances by $D_p + \epsilon$; trajectory matching divides them by $\max(D_p, \epsilon)$. Ties in greedy predecessor matching select the first token index. Initial trajectory and state identities are $\mathrm{tr}(1, i) = \mathrm{st}(1, i) = i$, and newly created states receive distinct identities. For $T = 1$, message passing and trajectory construction are skipped; effective-rank allocation, saliency MMR, and within-frame fusion remain available.

Exact integer budgets and frame quotas. Unit token budgets use nearest-integer rounding with ties to even, followed by clipping to $[1, |\mathcal{U}_u| M]$. Consequently, the realized video budget is $K = \sum_u B_u$. For continuous frame quotas $a_t = B_u w_t / \max(\sum_{t' \in \mathcal{U}_u} w_{t'}, \epsilon)$, initialize

$$b_t^{(0)} = \min\{M, \max\{0, \lfloor a_t \rfloor\}\}, \qquad R = B_u - \sum_{t \in \mathcal{U}_u} b_t^{(0)}, \qquad f_t = a_t - \lfloor a_t \rfloor. \tag{27}$$

Order frames by decreasing f_t, breaking ties by increasing frame index. Traverse this order, adding one token to each frame with $b_t < M$ until $R = 0$; repeat the traversal when capping creates a larger residual. If numerical rounding produces an excess, remove tokens from positive quotas in reverse order until the sum is B_u. Feasibility follows from $1 \leq B_u \leq |\mathcal{U}_u| M$. If all weights are zero, all remainders tie and this procedure distributes the unit budget cyclically in frame order. A frame with $b_t = 0$ contributes no anchors; its tokens remain potential fusion sources. MMR uses zero redundancy before the first pick and resolves equal maximization scores by the first available token index.

Fusion admission and candidate counts. For each source, an empty temporal candidate set leads directly to discarding that source. Otherwise, the best penalized similarity is compared with τ_m. An admitted source routes to its top-k_x destinations, where $k_x = \min(k, |\mathcal{V}(x)|)$; individual destinations are not separately thresholded. The vectorized implementation masks unavailable destinations by -10^9, so padded top-k entries have zero softmax weight when fewer than k destinations are available. Similarity ties in top-k selection follow the underlying tensor operator.

For stable temporal weighting, define $G_\sigma(\Delta) = \max\{\exp[-\Delta^2/(2\sigma^2)], 10^{-6}\}$ for $\sigma > 0$. Its routing-logit contribution is $\log G_\sigma(t_a - t_x)$; setting the configuration scale to zero disables this contribution. The transport temperature is lower-bounded by 0.01, and softmax uses maximum-subtracted logits. Before routing, the raw evidence scores are normalized jointly over $\mathcal{I}_u$: $e = \mathcal{N}_u(e^{\mathrm{raw}})$. Accepted sources contribute $\omega_{xa}(0.5 + \mathcal{N}_u(s)_x)\mathbf{z}_x$. Each accumulator starts at $\mathbf{z}_a$ with weight one, so anchors receiving no context retain their input representation and fusion never changes the output token count.

E.4 Backbone Position Handling

After fusion, anchors are assembled in increasing order of their original flattened token indices, together with all retained text and formatting tokens. The compressed tensor is packed for computation; positional indices follow the backbone-specific rules below.

Qwen3-VL constructs its native temporal, height, and width indices before compression. The wrapper gathers all three components with the same global keep indices used for the embeddings and

slices the attention mask accordingly. The reference setting keeps the anchor's native coordinates. Qwen3.5 likewise computes native three-dimensional indices before compression and gathers them with the retained indices. Qwen3-VL's additional visual feature streams are selected at the same anchor indices. This preserves the original positional coordinates of retained visual and text tokens even though their storage sequence is shorter.

LLaVA-OneVision first constructs its native multimodal sequence, retaining its visual separator tokens. After replacing selected visual embeddings with fused anchors and removing unselected visual embeddings, the wrapper uses contiguous one-dimensional positions $0, \ldots, N_{\text{packed}} - 1$ for the packed sequence, including text and separators. When explicit position indices are absent, the language model generates this contiguous sequence internally. Thus sequence order is preserved in every backbone, while the positional coordinate system follows the integration used by that backbone.

E.5 Ablation Configurations

Tables 3 and 4 evaluate ActivityNet-Captions with Qwen3-VL-4B at $r = 12.5\%$. The reference system uses temporal message passing, trajectory-state construction, query-conditioned allocation, MMR with temporal discounting, and query-gated soft fusion. Each variant below specifies the changed computation and the operations that remain active. All variants preserve the prescribed unit token budgets.

Removing the temporal state field. The w/o TSF variant disables message passing and trajectory construction, so $\mathbf{z} = \mathbf{h}$ and no transition scores or trajectory/state identities are supplied downstream. EDA uses effective-rank-only allocation. ESS uses saliency MMR with no predecessor discount. BPF omits the evidence penalty and trajectory reward, while retaining cosine similarity, the temporal window $W = 2$, the source-admission threshold $\tau_m = 0.6$, Gaussian temporal weighting, and importance-weighted accumulation. Its routing logit is $\cos(\mathbf{h}_x, \mathbf{h}_a)/\lambda_m + \log G_\sigma(t_a - t_x)$, where G_σ is the stabilized Gaussian defined above. This defines the complete fallback computation without assigning artificial trajectory or state identities.

Allocation and selection controls. Rank-only allocation sets $\beta_1 = \beta_2 = 0$. Frame quotas still depend on effective rank and use the same capacity-constrained integer allocation; TSF, temporal discounting, and BPF remain active. The w/o Temporal Discount variant sets $\gamma = 0$ and retains saliency MMR, its spatial redundancy term, and all frame quotas from EDA. These controls evaluate the temporal contribution to allocation and selection on top of their spatial criteria.

Removing context fusion. The w/o Context Fusion variant uses the same TSF, EDA, and ESS computations as full TIDE, then returns the selected contextualized anchors $\mathbf{z}_a$ without adding peripheral source contributions. This is the implementation's `pure_select` output. The anchors retain TSF's message-passing context, so the comparison measures the contribution of peripheral context recovered after selection.

Message passing and trajectory controls. The w/o Message Passing variant uses $\mathbf{z} = \mathbf{h}$, then recomputes trajectories, states, and all dependent allocation and fusion signals from these features. The w/o Trajectory Signals variant retains message passing but supplies no trajectory identities, transition scores, or state identities to the downstream stages. It therefore uses the same rank-only allocation, undiscounted MMR, and local Gaussian-weighted fusion as w/o TSF, operating on the contextualized features $\mathbf{z}$. These two controls distinguish feature contextualization from the subsequent use of correspondence signals.

Alternative boundary fusion. Frame-boundary fusion replaces the query-gated state-routing branch with the frame-boundary-masked fusion branch while retaining the upstream trajectory-conditioned allocation and anchor selection. This branch constructs saliency-weighted frame descriptors within each unit, finds the largest adjacent cosine change, and inserts a cut when that change exceeds 0.10 and the next-largest change by at least 0.05. Units of at most two frames have no cut. The resulting mask permits fusion within each side of the accepted cut; units without an accepted cut retain the full local window. The variant uses soft top-k transport, temporal proximity

Table 11: Reference hyperparameters and reported sensitivity values. Defaults for swept parameters match the gray rows in Tables 13–15. Fixed settings have no reported sweep.

Module	Parameter	Default	Reported Sweep	Functional Role
Global	L	8	$\{2, 4, 6, 8, 10, 12, 16\}$	Temporal unit length
TSF	K_a	8	Fixed	Message neighbors per direction
	λ_s	0.3	Fixed	Spatial weight in affinity matrix
	τ_a	0.1	Fixed	One-hop aggregation temperature
	κ	0.8	$\{0.2, 0.4, 0.6, 0.8, 1.0\}$	State transition quantile cutoff
	ϵ	10^{-6}	Fixed	Numerical safety threshold
EDA	β_1	0.4	$\{0.2, 0.4, 0.6, 0.8, 1.0\}$	Trajectory change vs. effective-rank weight
	β_2	0.8	$\{0.2, 0.4, 0.6, 0.8, 1.0\}$	Query-conditioned transition vs. base weight
	LR	–	Fixed	Largest-remainder integer rounding
ESS	γ	0.3	$\{0.1, 0.3, 0.5, 0.7, 0.9\}$	Static similarity discount weight
	λ	0.7	Fixed	MMR relevance–diversity trade-off
BPF	k	4	$\{1, 2, 4, 6, 8, 12, 16\}$	Maximum fusion targets per source
	W	2	$\{0, 1, 2, 3, 4, 6, 7\}$	Candidate temporal window radius
	α	0.5	$\{0.1, 0.3, 0.5, 0.7, 0.9\}$	Cross-state boundary penalty
	λ_m	0.5	$\{0.1, 0.3, 0.5, 0.7, 0.9\}$	Soft transport temperature
	τ_m	0.6	$\{0.2, 0.4, 0.6, 0.8, 1.0\}$	Source-admission similarity threshold
	σ	1.0	$\{0.2, 0.4, 0.6, 0.8, 1.0\}$	Temporal Gaussian scale parameter
	η	0.5	$\{0.1, 0.3, 0.5, 0.7, 0.9\}$	Token trajectory routing bonus

weighting, and trajectory rewards, but does not compute the query-gated source evidence penalty. Its comparison with BPF measures the two complete boundary-handling schemes.

Temporal routing controls. The w/o Temporal Prior variant disables the Gaussian factor in the routing logits and retains the local candidate window, state penalty, trajectory reward, and source-admission threshold. The w/o Trajectory Reward variant sets $\eta = 0$ in the routing logits; it retains trajectory construction, query-conditioned allocation, temporal discounting, state compatibility, and the Gaussian prior. These variants isolate the corresponding routing terms within BPF.

E.6 Reference Hyperparameters and Sensitivity Protocol

The gray rows in Tables 13–15 define the reference configuration used throughout this manuscript. Table 11 collects these values together with the remaining fixed settings. Unless a sensitivity experiment or ablation explicitly changes a setting, it takes the reference value.

Sensitivity is evaluated on the Charades-STA test split with Qwen3-VL-4B at $r = 25\%$, using the same 3,720 queries, preprocessing, prompt, and decoding configuration at every point. Each run changes one parameter and holds the others at their reference values. The discrete values listed in Table 11 match the reported sweeps. Gray rows mark the reference values, and bold mIoU entries mark sweep maxima. These experiments characterize sensitivity around the reference configuration.

E.7 Hardware and Software

Our experiments and latency profiling use an NVIDIA A800-SXM4 GPU, dual Intel Xeon Platinum 8358 CPUs, and 512 GB of system memory. We use Ubuntu 22.04, PyTorch 2.1.2 with CUDA 12.1, Transformers 4.57.1, FlashAttention-2 2.5.6, and Decord 0.6.0. Model computation uses `bfloat16` precision. VTG and VideoQA use greedy decoding with temperature 0.0, top-p 1.0, and at most 256 new tokens. TIDE uses deterministic selection and rounding without learned parameters.

F Prompt Templates and Evaluation Settings

We report the user-message templates and input settings used for VTG and VideoQA. The backbone's native chat template combines this text with the video input. In the templates below, {sentence} is the grounding query, {question} is the VideoQA question, and {options} contains the lettered answer choices, one per line.

F.1 Video Temporal Grounding (VTG)

For video temporal grounding, we use Qwen3-VL-4B-Thinking and Qwen3-VL-8B-Thinking, abbreviated as Qwen3-VL-4B and Qwen3-VL-8B throughout the tables. After applying the native chat template, we append </think> followed by two newline characters to suppress thinking-style output, following SemVID (Li et al., 2026a). Generation uses do_sample=False and at most 256 new tokens. Qwen3.5 uses the same timestamp output format.

The shorter image side is resized to 480 pixels for Charades-STA and 256 pixels for ActivityNet-Captions, matching SemVID's image preprocessing. The sampling rates stated in the main text are supplied to the video loader before tokenization. The backbone's temporal patching determines the number of visual time steps passed to TIDE; for Qwen3-VL, this count is the temporal component of video_grid_thw. We use the model's native positional coordinates for these steps as described in Appendix E.3. The prompt requests a JSON object containing the predicted start and end times in seconds:

VTG user prompt

```
Given a textual query: {sentence}
When does the described content occur in the video? Return ONLY a
JSON object exactly in this format: {"Prediction": [start_time,
end_time]}. Use seconds as numbers and do not add any other text.
```

F.2 Video Question Answering (VideoQA)

For LLaVA-OneVision-7B, we use prompt formats adapted from lmms-eval, an open-source framework for evaluating large multimodal models.[1] In our evaluation code, lmms_eval_videomme and lmms_eval are the prompt_style configuration values for Video-MME and LongVideoBench, respectively. Their user-message templates are shown below.

Video-MME user prompt

```
Select the best answer to the following multiple-choice question based
on the video. Respond with only the letter (A, B, C, or D) of the
correct option.
{question}
{options}
Answer with the option's letter from the given choices directly.
```

LongVideoBench user prompt

```
{question}
{options}
Answer with the option's letter from the given choices directly.
```

For both benchmarks, we extract the predicted option letter from the generated answer and compare it with the annotated answer to compute accuracy.

[1] https://github.com/EvolvingLMMs-Lab/lmms-eval

Table 12: VTG results on ActivityNet-Captions with Qwen3.5-9B. The (%) column reports mIoU relative to the full-token baseline of the same backbone.

Method	$r = 25\%$					$r = 12.5\%$				
	R@0.3	**R@0.5**	**R@0.7**	**mIoU**	**(%)**	**R@0.3**	**R@0.5**	**R@0.7**	**mIoU**	**(%)**
Full-Token	68.79	52.45	35.68	51.10	100.00	68.79	52.45	35.68	51.10	100.00
FastVID	44.42	29.31	16.90	31.78	62.19	43.56	28.73	16.39	31.21	61.08
FlashVID	65.26	48.83	32.98	48.51	94.93	63.74	47.10	31.46	47.12	92.21
SemVID	57.01	41.66	27.96	42.51	83.19	53.04	38.16	24.99	38.73	75.79
TIDE	**68.49**	**52.05**	**34.70**	**50.63**	**99.08**	**66.96**	**49.93**	**32.57**	**49.05**	**95.99**

G Additional Experiments

G.1 Performance of Qwen3.5-9B

We evaluate Qwen3.5-9B, the largest backbone in our study, on ActivityNet-Captions to test TIDE at a larger model scale.

On Qwen3.5-9B, TIDE leads all compared compression methods at both retention ratios, reaching 50.63 mIoU at 25% and 49.05 mIoU at 12.5%. These scores retain 99.08% and 95.99% of full-token mIoU and extend the observed advantage to the largest backbone in our evaluation.

G.2 Full Hyperparameter Sensitivity Results

We report single-factor sensitivity sweeps on Charades-STA with Qwen3-VL-4B at r=25% retention (n=3720 queries per point). Each experiment varies one parameter and keeps the others at their defaults. Shaded rows mark the default values; bold mIoU values identify the best result in each sweep.

Core allocation and fusion parameters. Table 13 sweeps five parameters that govern budget allocation and fusion routing. Across the tested ranges, each changes mIoU by at most 0.25 pp, and the default remains within 0.1 pp of the best value for every parameter.

Table 13: Core parameter sensitivity on Charades-STA (r=25%, Qwen3-VL-4B).

(a) β_2 (query-transition allocation weight), default=0.8

β_2	R@0.3	R@0.5	R@0.7	mIoU
0.2	79.44	64.65	39.03	55.60
0.4	79.62	64.62	39.11	55.58
0.6	**79.81**	**64.97**	**38.92**	**55.72**
0.8	79.54	64.95	38.95	55.62
1.0	79.38	64.60	38.79	55.49

(b) α (boundary penalty), default=0.5

α	R@0.3	R@0.5	R@0.7	mIoU
0.1	79.27	64.70	38.68	55.50
0.3	79.35	64.70	38.82	55.53
0.5	79.54	64.95	38.95	55.62
0.7	**79.46**	**64.81**	**39.17**	**55.65**
0.9	79.46	64.76	38.76	55.60

(c) κ (state quantile), default=0.8

κ	R@0.3	R@0.5	R@0.7	mIoU
0.2	79.44	65.00	38.82	55.64
0.4	**79.38**	**65.11**	**39.03**	**55.69**
0.6	79.25	64.73	38.95	55.56
0.8	79.54	64.95	38.95	55.62
1.0	79.38	64.78	38.98	55.58

(d) λ_m (transport softmax temperature), default=0.5

λ_m	R@0.3	R@0.5	R@0.7	mIoU
0.1	79.41	64.84	38.74	55.53
0.3	79.33	64.78	38.90	55.57
0.5	79.54	64.95	38.95	55.62
0.7	**79.57**	**64.97**	**38.87**	**55.67**
0.9	79.27	64.76	38.63	55.52

(e) τ_m (source-admission threshold), default=0.6

τ_m	R@0.3	R@0.5	R@0.7	mIoU
0.2	79.49	64.87	39.06	55.62
0.4	**79.57**	**64.81**	**39.06**	**55.66**
0.6	79.54	64.95	38.95	55.62
0.8	79.38	64.84	38.63	55.60
1.0	79.57	64.97	38.68	55.61

Structural topology parameters. Table 14 varies unit length L, the number of fusion targets k, and temporal radius W on Charades-STA at 25% retention. For L from 2 to 12, mIoU stays between 55.55 and 55.74; the full sweep through 16 frames spans less than 0.3 points. Varying k from 1 to

16 changes mIoU by 0.10 points. The W sweep ranges from 55.42 to 55.69, placing the default configuration within 0.07 points of the best value.

The endpoint settings clarify which information the fusion stage can use. Setting $W = 0$ restricts fusion to the source frame, while $\tau_m = 1$ in Table 13 strongly restricts source admission. Their mIoU scores, 55.69 and 55.61, remain close to the default 55.62, showing that the selected anchors retain much of the grounding information in this Charades-STA setting. In the ActivityNet-Captions experiment at 12.5% retention, adding BPF improves mIoU by 1.94 points (Table 3). These results describe how the value of recovered context varies across the evaluated dataset and budget settings.

Table 14: Structure parameter sensitivity on Charades-STA (r=25%, Qwen3-VL-4B).

(a) L (unit length), default=8

L	R@0.3	R@0.5	R@0.7	mIoU
2	79.68	64.87	39.49	55.66
4	79.52	64.68	39.68	55.60
6	79.60	64.81	39.49	55.66
8	79.54	64.95	38.95	55.62
10	79.60	64.60	39.41	55.55
12	**79.70**	**64.76**	**39.65**	**55.74**
16	79.19	64.33	39.81	55.46

(b) k (top-k targets), default=4

k	R@0.3	R@0.5	R@0.7	mIoU
1	79.62	64.89	39.19	55.66
2	79.54	64.89	39.03	55.62
4	79.54	64.95	38.95	55.62
6	79.52	64.81	39.38	55.60
8	**79.38**	**64.95**	**39.46**	**55.68**
12	79.35	64.78	39.14	55.58
16	79.60	64.95	39.27	55.61

(c) W (temporal window), default=2

W	R@0.3	R@0.5	R@0.7	mIoU
0	**79.52**	**64.95**	**39.44**	**55.69**
1	79.35	64.35	38.76	55.42
2	79.54	64.95	38.95	55.62
3	79.30	64.44	38.95	55.54
4	79.35	64.89	39.03	55.64
6	79.44	64.65	39.06	55.58
7	79.35	64.68	39.17	55.58

Trajectory and transport parameters. Table 15 varies β_1, η, γ, and σ in the same Charades-STA setting. The β_1, η, and σ sweeps each span at most 0.12 mIoU points. For γ, performance rises to 55.82 at the upper end of the tested range, favoring stronger discounting of temporally repeated content.

Table 15: Trajectory and transport parameter sensitivity on Charades-STA (r=25%, Qwen3-VL-4B).

(a) β_1 (trajectory-change allocation weight), default=0.4

β_1	R@0.3	R@0.5	R@0.7	mIoU
0.2	**79.54**	**64.73**	**39.14**	**55.64**
0.4	79.54	64.95	38.95	55.62
0.6	79.41	64.84	38.76	55.54
0.8	79.33	64.57	38.90	55.55
1.0	79.38	64.73	39.27	55.57

(b) η (token trajectory bonus), default=0.5

η	R@0.3	R@0.5	R@0.7	mIoU
0.1	79.35	64.81	38.79	55.58
0.3	79.38	65.11	38.79	55.64
0.5	79.54	64.95	38.95	55.62
0.7	**79.33**	**64.76**	**39.22**	**55.65**
0.9	79.46	64.73	38.98	55.61

(c) γ (static discount weight), default=0.3

γ	R@0.3	R@0.5	R@0.7	mIoU
0.1	79.54	64.65	39.49	55.63
0.3	79.54	64.95	38.95	55.62
0.5	79.54	64.76	39.09	55.64
0.7	79.89	65.00	39.25	55.71
0.9	**80.05**	**64.84**	**39.44**	**55.82**

(d) σ (temporal Gaussian scale), default=1.0

σ	R@0.3	R@0.5	R@0.7	mIoU
0.2	79.60	64.57	38.92	55.61
0.4	79.57	64.60	38.79	**55.66**
0.6	79.54	64.46	38.90	55.59
0.8	79.57	64.76	38.49	55.57
1.0	79.54	64.95	38.95	55.62

H Limitations and Future Work

TIDE builds its temporal reference from greedy correspondences between adjacent frames. These local feature matches can become ambiguous under occlusion, rapid motion, or scene cuts, affecting transition scores and inherited state identities. Query relevance is also estimated from input-embedding similarity, whose agreement with the queried action depends on the backbone's representations. Confidence-aware matching and explicit alignment with action descriptions are directions for extending the current training-free design.

Disjoint temporal units of $L = 8$ frames bound local processing and preserve a fixed token budget within each unit. TSF correspondences and state identities remain continuous across the video, but EDA's two-sided allocation and BPF's routing operate within unit boundaries. A transition at a unit seam therefore receives a different allocation treatment from one inside a unit, and context cannot

be fused across the seam. The unit-length sweep in Table 14 reports aggregate accuracy; evaluating transitions near seams would help assess overlapping or adaptive units. Full-video embeddings and TSF metadata also require memory that grows with video length, motivating a streaming extension.

Our reported results cover 25% and 12.5% token retention. Below 12.5%, smaller integer budgets can leave frames with few or no anchors, reducing the available destinations for local fusion and potentially removing brief transition evidence. Performance in this regime remains to be evaluated. Segment-level summaries offer a possible extension for representing temporal evidence when retaining individual tokens becomes too restrictive.