LTC: RISK-GATED LAGRANGIAN TOKEN COMPRESSION FOR VIDEO LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

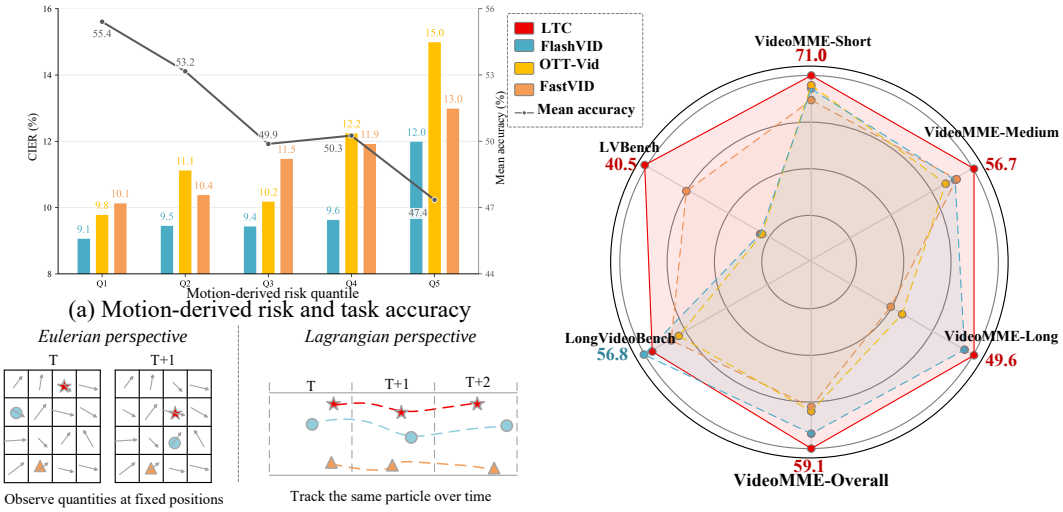
Video Large Language Models (video-LLMs) process large numbers of visual tokens, incurring substantial computational and memory overhead. Existing training-free compression methods reduce spatiotemporal redundancy through importance-based selection, cross-frame matching, and structured aggregation. However, these methods neglect the local motion structure of neighboring correspondences, which may lead to the loss of key dynamic information during compression. Based on this insight, we propose risk-gated Lagrangian Token Compression (LTC), a training-free inference framework for video-LLMs. Specifically, Token Flow Estimation (TFE) uses local feature matching to estimate token displacements and matching confidence. Eulerian Risk Diagnosis (ERD) then combines neighborhood motion structure with active low-confidence signals to construct an aggregation-risk proxy. Guided by this risk proxy, Risk-Gated Lagrangian Aggregation (RGLA) keeps protected tokens as independent candidates, aggregates the remaining tokens that satisfy feature-consistency constraints into compact representatives along temporal trajectories, and selects the final visual tokens by importance under a token budget. The framework directly reuses pretrained visual features without training an additional network. Experiments on three representative video-LLMs and three video understanding benchmarks validate LTC's effectiveness and applicability across models. Retaining only 25% of visual tokens, LTC achieves 101.3% of the vanilla model's average performance on LLaVA-OneVision. Code is available at `https://anonymous.4open.science/r/LTC-75E4`.

1 INTRODUCTION

In recent years, video Large Language Models (video-LLMs) have demonstrated strong multimodal understanding capabilities in tasks such as video question answering, temporal understanding, and long-video reasoning. These capabilities are supported by advances in unified image–video representations and video instruction tuning (Lin et al., 2024; Li et al., 2025; Zhang et al., 2025c). Videos are typically encoded into a large number of visual tokens, a count that grows with the number of input frames and spatial resolution, substantially increasing the computational and memory overhead of language models. Therefore, compressing video redundancy while preserving critical semantic and dynamic information under a limited token budget has become an important problem for efficient inference in video-LLMs.

Although existing methods such as FlashVID (Fan et al., 2026) and OTT-Vid (Kang et al., 2026) reduce redundancy through cross-frame matching, they do not explicitly account for displacement variation among neighboring correspondences when making compression decisions. To examine this issue, we compute motion-derived risk statistics from token flow and group videos into risk quantiles. To focus on errors introduced by compression, we report the Compression-Induced Error Rate (CIER), defined as the fraction of questions answered correctly by the vanilla (uncompressed) model that become incorrect after compression. As shown in Figure 1(a), higher-risk groups tend to exhibit higher CIER across the comparison methods, alongside lower mean task accuracy. These observations motivate us to investigate neighborhood motion structure as an additional cue for guiding compression. This leads to a natural question: "How can neighborhood motion structure be used to diagnose potential compression risks, and balance cross-frame redundancy aggregation with the preservation of critical dynamic information under a limited token budget?"

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.



(a) Motion-derived risk and task accuracy

(b) Eulerian and Lagrangian perspectives

(c) Performance comparison at 25% token retention

Figure 1: **Motivation and performance of LTC.** (a) Compression-Induced Error Rates (CIER; bars, left axis) of FlashVID, OTT-Vid, and FastVID, and their mean accuracy (gray line, right axis), across motion-derived risk quantiles. CIER measures the fraction of initially correct vanilla predictions that become incorrect after compression. (b) Eulerian diagnosis at fixed grid locations versus Lagrangian aggregation along cross-frame correspondences. (c) LLaVA-OneVision performance at 25% token retention across three benchmarks and VideoMME duration subsets.

To address this challenge, we propose risk-gated Lagrangian Token Compression (LTC), a training-free, plug-and-play framework for video-LLMs. In fluid mechanics, the Eulerian view examines a flow field at fixed spatial locations, whereas the Lagrangian view follows a material element as its position changes over time. LTC draws on this distinction: it diagnoses aggregation risk from local displacement patterns on the fixed token grid and merges eligible tokens along estimated cross-frame correspondence trajectories (Figure 1(b)). At the core of LTC, Risk-Gated Lagrangian Aggregation (RGLA) reduces cross-frame redundancy through risk-gated merging along motion trajectories, keeping potentially compression-sensitive tokens independent while aggregating eligible ones. Eulerian Risk Diagnosis (ERD) identifies tokens for protection using divergence, curl, and shear of the displacement field after dominant translation removal, together with active low-confidence signals. Token Flow Estimation (TFE) supports the other two stages by locally matching pretrained visual features to estimate displacement and matching confidence. This "diagnosis before aggregation" design balances redundancy reduction with dynamic information preservation without additional training.

We evaluate LTC at different token budgets on LLaVA-OneVision (Li et al., 2025) and LLaVA-Video (Zhang et al., 2025c), and with more input frames on Qwen3-VL (Qwen Team, 2025), across VideoMME (Fu et al., 2025), LongVideoBench (Wu et al., 2024), and LVBench (Wang et al., 2025). Figure 1(c) compares LLaVA-OneVision benchmark scores at 25% retention. LTC achieves the highest compressed average, reaching 101.3% of vanilla performance. On Qwen3-VL, LTC processes 128 frames at 25% retention and improves average accuracy by 4.4% relative to the 32-frame vanilla baseline. Compared with uncompressed 128-frame inference, LTC reduces Time-To-First-Token (TTFT) from 11.93 s to 4.99 s (58.17%). Our main contributions are summarized as follows:

- We identify that existing video token compression methods rely primarily on token importance and cross-frame feature similarity, without explicitly characterizing local motion variation among neighboring correspondences.
- We propose LTC, a training-free framework for video token compression. Through the TFE–ERD–RGLA pipeline, LTC uses risk information to guide token protection during aggregation and representative selection under a fixed token budget.
- We validate the overall design through cross-model evaluations, module ablations, and experiments with increased numbers of input frames. Across three representative video-

LLMs and three video understanding benchmarks, LTC maintains competitive performance while reducing visual tokens and demonstrates the potential for accommodating more video frames.

2 Related work

2.1 Video large language models

Visual instruction tuning connects visual encoders to language models for multimodal instruction following (Liu et al., 2023). Video-LLaVA (Lin et al., 2024) aligns image and video representations before projection, while LLaVA-OneVision (Li et al., 2025) and LLaVA-Video (Zhang et al., 2025c) develop visual task transfer and video instruction tuning. These video-LLMs, together with Qwen3-VL, support video understanding but must process increasingly long visual sequences as more frames are sampled. LTC targets this inference bottleneck by compressing visual tokens while keeping the pretrained model parameters unchanged.

2.2 Visual token compression

Visual token compression combines content selection with redundancy reduction. In vision transformers, EViT (Liang et al., 2022) retains attentive tokens and fuses the remainder, while ToMe (Bolya et al., 2023) merges similar tokens. For vision–language models, FastV (Chen et al., 2024) and SparseVLM (Zhang et al., 2025b) prune visual tokens within the language model using attention or text guidance; VisionZip (Yang et al., 2025) selects informative visual features before the language model. These methods motivate compact visual representations but do not by themselves address the reliability of aggregation along video trajectories.

Video-specific approaches additionally exploit temporal structure. TokenLearner (Ryoo et al., 2021) learns compact, input-dependent video tokens, and TESTA (Ren et al., 2023) aggregates similar frames and spatial patches in a video encoder. Among training-free compression methods for video-LLMs, PruneVid (Huang et al., 2025) combines temporal and spatial token reduction with query-relevant pruning. FastVID (Shen et al., 2025) uses dynamic temporal segmentation and density-based pruning. FlashVID (Fan et al., 2026) combines representative selection with spatiotemporal redundancy trees built through cross-position matching, while OTT-Vid (Kang et al., 2026) uses optimal transport to model token importance and frame-pair compressibility. Building on the importance of both redundancy reduction and dynamic-information preservation, LTC focuses on the local reliability of cross-frame aggregation: neighborhood displacement structure and matching uncertainty guide risk protection before eligible tokens are aggregated along trajectories.

2.3 Temporal correspondence and motion representation

Correspondence methods provide a complementary basis for organizing information over time. CoTracker (Karaev et al., 2024) jointly models point trajectories, and Coarse Correspondences (Liu et al., 2025) convey object correspondences to multimodal models through visual prompting. LART (Rajasegaran et al., 2023) adopts a Lagrangian view by combining pose and appearance along person tracklets for action recognition. LTC instead estimates token correspondences from pretrained visual features and uses them for compression. As illustrated in Figure 2, candidate correspondences can occur in neighborhoods with different displacement patterns. LTC analyzes these patterns on a fixed grid to diagnose aggregation risk, then constructs feature-consistent trajectories for eligible tokens. This connects Eulerian risk diagnosis with Lagrangian aggregation under a fixed token budget.

3 Methodology

3.1 Overview

LTC compresses visual tokens before language-model input without updating model parameters. For F sampled frames, let $X \in \mathbb{R}^{T\times H\times W\times C}$ be the encoded grid, where T is the encoded temporal length, which may differ from the sampled frame count F. H and W denote the spatial grid dimensions, C is the feature dimension, and $N = THW$ is the token count. At retention ratio r, LTC

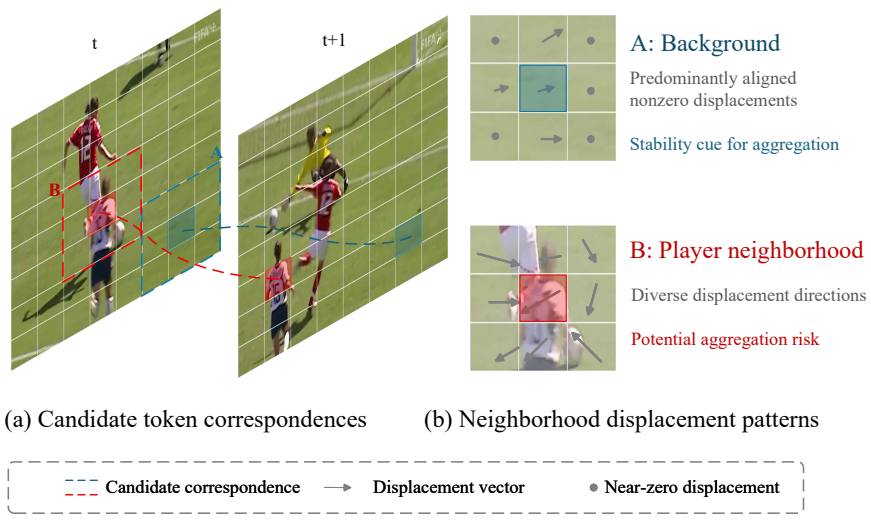


Figure 2: **Illustration of cross-frame candidate correspondences and neighborhood displacement structure.** (a) Candidate token correspondences between adjacent frames t and $t+1$. Blue and red dashed lines mark background region A and person region B, respectively. (b) Nonzero displacements in the background neighborhood have relatively consistent directions, while displacement directions in the person neighborhood are more dispersed. Gray arrows denote displacement vectors, and gray dots denote near-zero displacement.

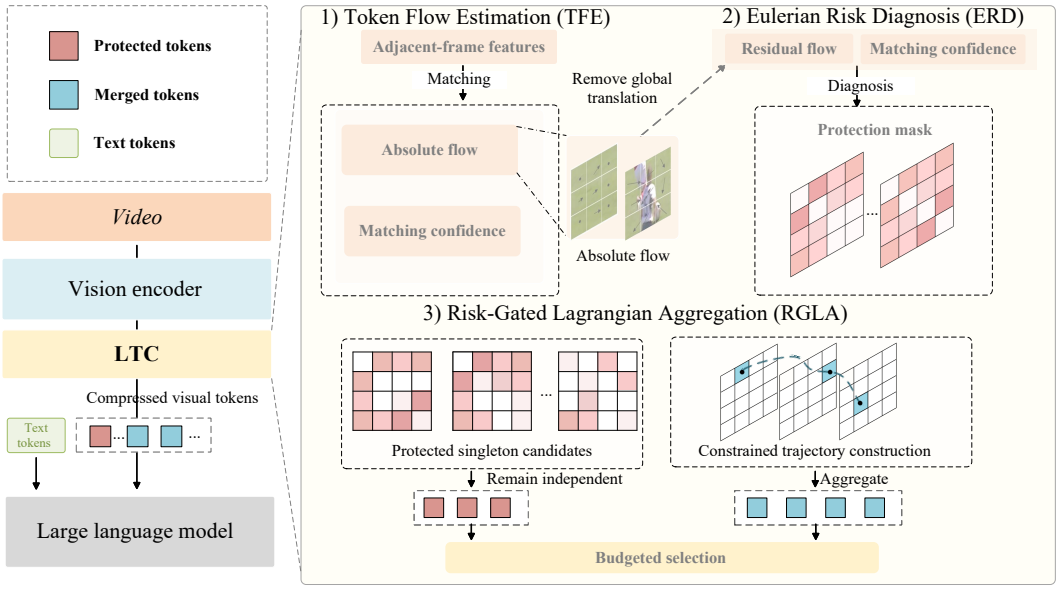


Figure 3: **Overview of LTC.** TFE estimates an absolute token flow from adjacent-frame features. ERD analyzes a residual flow after subtracting the dominant translation and combines motion structure with active low-confidence evidence. RGLA keeps protected tokens independent during trajectory construction and aggregates eligible tokens along routes guided by the absolute flow. Budgeted representative selection determines the final visual sequence.

outputs at most $B = \lceil rN \rceil$ representatives. LTC diagnoses aggregation risk on the fixed token grid (Eulerian) and aggregates eligible tokens along trajectories formed from cross-frame correspondences (Lagrangian; Figure 3).

3.2 TOKEN FLOW ESTIMATION (TFE)

Tokens with similar visual features can yield competing cross-frame correspondences. TFE therefore estimates both displacement and confidence. For a token at $(t, \mathbf{x})$, we search valid positions $\mathbf{y} \in \mathcal{N}(\mathbf{x})$ in a local window of the next frame, limiting search cost while allowing cross-position correspondence. We use ℓ_2-normalized features $\hat{\mathbf{x}}_{t,\mathbf{x}}$ so that matching compares feature directions through cosine similarity:

$$p_t(\mathbf{y} \mid \mathbf{x}) = \frac{\exp\big(\hat{\mathbf{x}}_{t,\mathbf{x}}^\top \hat{\mathbf{x}}_{t+1,\mathbf{y}}/\tau_m\big)}{\sum_{\mathbf{z}\in\mathcal{N}(\mathbf{x})} \exp\big(\hat{\mathbf{x}}_{t,\mathbf{x}}^\top \hat{\mathbf{x}}_{t+1,\mathbf{z}}/\tau_m\big)}, \tag{1}$$

where τ_m controls the sharpness of the matching distribution. Confidence combines two complementary cues: concentration across all candidates and the separation between the two leading matches:

$$c_t = (1-\eta)\left(1 - \frac{\mathcal{H}(p_t)}{\log M}\right) + \eta \frac{p_t^{(1)} - p_t^{(2)}}{\max(p_t^{(1)}, \epsilon)}, \tag{2}$$

where $M = |\mathcal{N}(\mathbf{x})|$ is the number of valid matching candidates, $\mathcal{H}(p_t)$ is the distribution entropy, and $p_t^{(1)}, p_t^{(2)}$ are the two largest probabilities. Normalizing entropy by $\log M$ accounts for differences in candidate count, η balances the two cues, and ϵ stabilizes the denominator. High-confidence matches use the maximum-probability displacement to follow the dominant match; otherwise, the expected displacement avoids committing to a single ambiguous candidate. Confidence-weighted spatial smoothing suppresses isolated displacement noise while reducing the influence of unreliable matches, producing the absolute routing flow $\mathbf{v}_t$.

RGLA uses $\mathbf{v}_t$ because trajectory routing requires the full image-plane displacement. ERD uses the residual field $\mathbf{u}_t = \mathbf{v}_t - \mathbf{g}_t$ to reduce spurious boundary risk responses introduced by zero-padded Sobel filtering, where $\mathbf{g}_t$ is the component-wise median translation among sufficiently confident matches. The median reduces sensitivity to outlying displacements. TFE reuses pretrained visual features; boundary handling and filtering details are provided in Appendix B.

3.3 EULERIAN RISK DIAGNOSIS (ERD)

Local motion patterns vary across video regions and over time, and once merged into a single representative, the original tokens are no longer separate candidates for budgeted selection. ERD therefore examines neighborhood motion variation to identify candidates that should remain independent before aggregation. For the residual flow $\mathbf{u} = (u_x, u_y)$, Sobel derivatives give divergence d, scalar curl ω, and shear magnitude s:

$$\begin{aligned} d &= \partial_x u_x + \partial_y u_y, \qquad \omega = \partial_x u_y - \partial_y u_x, \\ s &= \sqrt{(\partial_x u_x - \partial_y u_y)^2 + (\partial_y u_x + \partial_x u_y)^2}. \end{aligned} \tag{3}$$

These describe local expansion or contraction, rotation, and shape distortion. Stronger responses serve as a conservative protection cue for local dynamics that aggregation may dilute. Their confidence-weighted combination defines the kinematic risk proxy:

$$k_t = (\kappa_d |d_t| + \kappa_\omega |\omega_t| + \kappa_s s_t)(0.25 + 0.75 c_t). \tag{4}$$

Confidence weighting downweights potentially noisy derivative evidence from uncertain matches. We average the risk and confidence estimates from the preceding and following temporal pairs at each spatial grid position, obtaining token-level scores k_i and $\bar{c}_i$. This combines evidence from both temporal sides while preserving the score scale; boundary positions use the single available pair. We select kinematic-risk candidates $\mathcal{K}$ from the top 20% of the within-video risk distribution, adapting selection to each video's risk scale.

Uncertain matches may still contain informative changes. We therefore identify active low-confidence candidates $\mathcal{U}$ separately. Let a_i be the normalized rank of the token feature norm, used as a proxy for activation strength, and n_i the normalized rank of feature change over time at the same spatial position. A token enters $\mathcal{U}$ when its confidence is at or below a threshold and both activation and novelty meet or exceed their quantile thresholds, with $a_i, n_i > 0$. This joint condition filters weak responses while prioritizing active, changing content.

Attention-based salient candidates $\mathcal{S}$ additionally cover visually prominent content with limited motion. The protection set $\mathcal{P} = \mathcal{K} \cup \mathcal{U} \cup \mathcal{S}$ is passed to RGLA to constrain trajectory aggregation. Thresholds and degenerate-field handling are detailed in Appendix B.

3.4 Risk-Gated Lagrangian Aggregation (RGLA)

Lagrangian trajectory construction. RGLA uses ERD's protection set $\mathcal{P}$ to filter links proposed by TFE's absolute flow $\mathbf{v}_t$. Protected tokens remain independent candidates, while eligible tokens are aggregated along Lagrangian trajectories constructed from estimated cross-frame displacements. For an eligible source $(t, \mathbf{x})$, RGLA proposes the destination by following the estimated token flow:

$$\mathbf{y} = \mathrm{round}(\mathbf{x} + \mathbf{v}_t(\mathbf{x}))\,. \tag{5}$$

A link is accepted only when the destination is in bounds, both endpoints are unprotected, their feature similarity meets the routing threshold, and the resulting trajectory contains at most $L_{\max}$ tokens. A trajectory G links tokens across successive encoded temporal positions, with $|G|$ denoting its length, and the length cap limits the span over which correspondence errors can accumulate.

Trajectory representative construction. Each trajectory is merged into a single representative. Because averaging can dilute distinctive features, we blend the similarity-weighted mean $\boldsymbol{\mu}_G$ with the trajectory member closest to it in cosine similarity:

$$i_G^* = \arg\max_{i \in G} \cos(\mathbf{x}_i, \boldsymbol{\mu}_G), \qquad \mathbf{z}_G = \lambda \mathbf{x}_{i_G^*} + (1-\lambda)\boldsymbol{\mu}_G. \tag{6}$$

This combines pooled temporal evidence with an observed feature reference. A member near the trajectory's temporal midpoint provides the position used to place the merged representative in the output sequence, independently of the feature anchor i_G^*.

Budgeted selection. Protected singleton candidates (each consisting of one unmerged visual token) and merged representatives jointly undergo selection under budget B. Token importance combines saliency, kinematic risk, active low-confidence, and temporal novelty:

$$I_i = w_s s_i^{\mathrm{attn}} + w_k \mathcal{R}(k_i) + w_u \mathcal{R}((1-\bar{c}_i) a_i n_i) + w_n n_i, \tag{7}$$

where $\mathcal{R}$ maps distinct values to ranks in $[0, 1]$ to reduce dependence on raw score scales, and s_i^{attn} is normalized attention saliency. To score each trajectory representative for selection, we combine the maximum and mean member importance, capturing brief and sustained importance, with a small logarithmic length bonus for temporal extent. Selection reserves candidates for temporal coverage, then allocates the remaining budget across active low-confidence, kinematic-risk, and low-risk groups. Candidates are ranked by importance within each group, and unused slots are filled by global ranking. Quota values are given in Appendix B.2.

4 Experiments

4.1 Experimental setup

Benchmarks and baselines. We evaluate LTC on VideoMME (Fu et al., 2025), LongVideoBench (Wu et al., 2024), and LVBench (Wang et al., 2025). Comparisons with FastVID (Shen et al., 2025), FlashVID (Fan et al., 2026), and OTT-Vid (Kang et al., 2026) use matched retention ratios on LLaVA-OneVision and LLaVA-Video. Avg. is the unweighted mean of VideoMME-Overall, LongVideoBench, and LVBench; Rel. Acc. normalizes it by the uncompressed vanilla reference specified in each table. Details are given in Appendix A.

Implementation details. We use LLaVA-OneVision-Qwen2-7B-OV (Li et al., 2025), LLaVA-Video-7B-Qwen2 (Zhang et al., 2025c), and Qwen3-VL-8B-Instruct (Qwen Team, 2025). We use 32 frames for LLaVA-OneVision and 64 for the main LLaVA-Video results. Additional 32-frame LLaVA-Video results appear in Appendix C.2. Qwen3-VL uses a 32-frame vanilla reference and additional higher-frame inputs. Evaluations use `lmms-eval` (Zhang et al., 2025a), batch size one, and a single NVIDIA A6000 GPU, with model weights, shared preprocessing, and decoding settings fixed within each backbone comparison. Additional configurations are provided in Appendix B.

Table 1: **Compression results on LLaVA-OneVision-Qwen2-7B-OV with 32 frames.** Scores are accuracy (%); Avg. and Rel. Acc. follow Section 4.1. Bold marks the highest displayed compressed score per retention group, including ties.

Method	Retention ratio r	VideoMME				LVBench	LongVideo Bench	Avg.	
		Short	Medium	Long	Overall			Score	Rel. Acc. (%)
LLaVA-OneVision — 32 frames									
Vanilla	100%	70.0	56.9	48.9	58.6	38.9	56.6	51.4	100.0
FastVID	50%	70.7	56.6	48.9	58.7	39.6	56.3	51.5	100.3
FlashVID	50%	**71.9**	55.2	49.8	**59.0**	39.6	55.7	51.4	100.1
OTT-Vid	50%	70.7	55.9	**50.0**	58.9	39.1	56.8	51.6	100.4
LTC	50%	69.7	**57.0**	48.7	58.4	**39.8**	**57.1**	**51.8**	**100.8**
FastVID	30%	71.1	55.3	47.8	58.1	39.4	55.7	51.1	99.4
FlashVID	30%	**71.3**	**56.2**	**49.1**	**58.9**	39.6	**56.8**	**51.8**	**100.8**
OTT-Vid	30%	71.0	56.0	48.4	58.5	39.2	56.6	51.4	100.1
LTC	30%	70.4	56.0	49.0	58.5	**40.0**	**56.8**	**51.8**	**100.8**
FastVID	25%	70.4	56.2	47.6	58.1	39.7	56.2	51.3	99.9
FlashVID	25%	70.7	56.2	49.4	58.8	38.2	**56.8**	51.3	99.8
OTT-Vid	25%	70.8	56.1	47.9	58.3	38.2	55.9	50.8	98.8
LTC	25%	**71.0**	**56.7**	**49.6**	**59.1**	**40.5**	56.6	**52.0**	**101.3**
FastVID	20%	68.8	55.6	46.7	57.0	38.7	57.0	50.9	99.0
FlashVID	20%	69.9	**56.1**	**48.9**	**58.3**	38.9	**58.0**	**51.7**	**100.7**
OTT-Vid	20%	70.0	55.3	48.0	57.8	39.3	57.8	51.6	100.5
LTC	20%	**71.1**	54.9	48.3	58.1	**39.6**	56.8	51.5	100.2

Table 2: **Compression results on LLaVA-Video-7B-Qwen2 with 64 frames.** Metrics follow Section 4.1, using the 64-frame vanilla reference. Additional results appear in Appendix C.2.

Method	Retention ratio r	VideoMME				LVBench	LongVideo Bench	Avg.	
		Short	Medium	Long	Overall			Score	Rel. Acc. (%)
LLaVA-Video — 64 frames									
Vanilla	100%	76.3	61.3	53.4	63.7	40.6	59.3	54.5	100.0
FastVID	50%	75.4	**62.1**	53.3	**63.6**	**41.1**	58.5	54.4	99.8
FlashVID	50%	75.6	61.2	53.1	63.3	40.9	59.2	**54.5**	**100.0**
OTT-Vid	50%	75.7	60.7	53.4	63.3	40.2	58.9	54.1	99.3
LTC	50%	**76.0**	61.1	**53.8**	**63.6**	40.3	**59.6**	**54.5**	**100.0**
FastVID	30%	**75.2**	**60.8**	52.3	**62.8**	**40.8**	58.5	54.0	99.1
FlashVID	30%	74.2	59.8	51.9	62.0	40.5	**59.5**	54.0	99.1
OTT-Vid	30%	75.0	60.2	52.4	62.6	39.2	59.2	53.7	98.5
LTC	30%	75.0	59.7	**53.6**	62.7	40.4	59.1	**54.1**	**99.3**

4.2 Main results

LLaVA-OneVision. LTC achieves the highest or joint-highest compressed average at 50%, 30%, and 25% retention, scoring 51.8, 51.8, and 52.0, respectively (Table 1). Its strongest average occurs at 25%, reaching 101.3% of vanilla performance. At 20%, LTC scores 51.5, retaining 100.2% of vanilla performance but trailing FlashVID's average of 51.7. Additional budgets appear in Appendix C.

Table 3: **Higher-frame inputs for Qwen3-VL-8B-Instruct at a fixed frame-equivalent budget of $F \cdot r = 32$ frames.** Vanilla uses 32 frames without compression. Parentheses indicate frame-count multiples relative to vanilla.

Method	Frames F	Retention ratio r	VideoMME	LVBench	LongVideo Bench	Avg.	
						Score	Rel. Acc. (%)
Qwen3-VL-8B-Instruct							
Vanilla	32 (1.0×)	100%	64.7	39.6	59.1	54.5	100.0
LTC	64 (2.0×)	50%	65.7	40.9	61.2	55.9	102.7
LTC	80 (2.5×)	40%	66.0	40.9	**62.2**	56.3	103.4
LTC	128 (4.0×)	25%	**67.4**	**41.8**	61.5	**56.9**	**104.4**

Table 4: **Component ablations of ERD and RGLA.** All configurations use LLaVA-OneVision with 32 frames at 25% visual-token retention, sharing TFE and the same final token budget. Mean CIER equally averages CIER across fixed risk quintiles Q1–Q5.

Method	VideoMME	LVBench	LongVideoBench	Avg.	Mean CIER
Base filter	58.7	38.2	55.3	50.7	11.9
ERD	59.0	39.4	56.5	51.7	10.9
LA	58.1	39.0	**57.2**	51.4	12.6
ERD + LA	58.0	40.1	56.8	51.7	11.5
ERD + LA + random gate	58.3	38.9	57.1	51.5	10.9
Full LTC (ERD + RG + LA)	**59.1**	**40.5**	56.6	**52.0**	**10.1**

LLaVA-Video. With 64 input frames, LTC ties FlashVID for the highest displayed compressed average at 50% and achieves the highest average at 30%. LTC leads on VideoMME-Long at both retention ratios and LongVideoBench at 50%. Additional results appear in Appendix C.2.

Higher-frame inputs on Qwen3-VL. At 64, 80, and 128 frames (50%, 40%, and 25% retention), LTC exceeds the 32-frame vanilla average of 54.5 (Table 3). The 128-frame setting reaches 104.4% of vanilla performance.

4.3 Ablation studies

Our ablations evaluate how Eulerian Risk Diagnosis (ERD) and Risk-Gated Lagrangian Aggregation (RGLA) contribute to redundancy reduction and information preservation. With TFE and the final token budget fixed, we isolate the effects of ERD's risk statistic and the two components of RGLA: Risk-Gated (RG) protection and Lagrangian Aggregation (LA). Base filter in Table 4 selects singleton tokens using only attention saliency and temporal novelty.

ERD: risk statistic. DCS combines divergence, curl, and shear of the residual displacement field into a confidence-weighted kinematic-risk statistic. We compare it with alternative statistics within the full LTC pipeline while holding other components fixed and matching kinematic-risk and additional protected-token counts per video (Table 5 in Appendix B.5). DCS achieves the highest average among the tested risk statistics. Table 4 shows that adding ERD to base filter improves accuracy across all three benchmarks and reduces mean CIER. These results support the benefit of risk-based scoring and category quotas for singleton selection, even without trajectory aggregation or K/U merge protection. "K/U" refers to the kinematic-risk ($\mathcal{K}$) and active low-confidence ($\mathcal{U}$) candidate sets.

RGLA: Risk-Gated (RG) protection and Lagrangian Aggregation (LA). LA alone omits K/U scoring, quotas, and merge protection. Compared with base filter, it improves average accuracy but raises mean CIER, showing that aggregation benefits can coexist with greater loss of previously correct predictions. This motivates evaluating not only whether to aggregate, but also which tokens should remain independent.

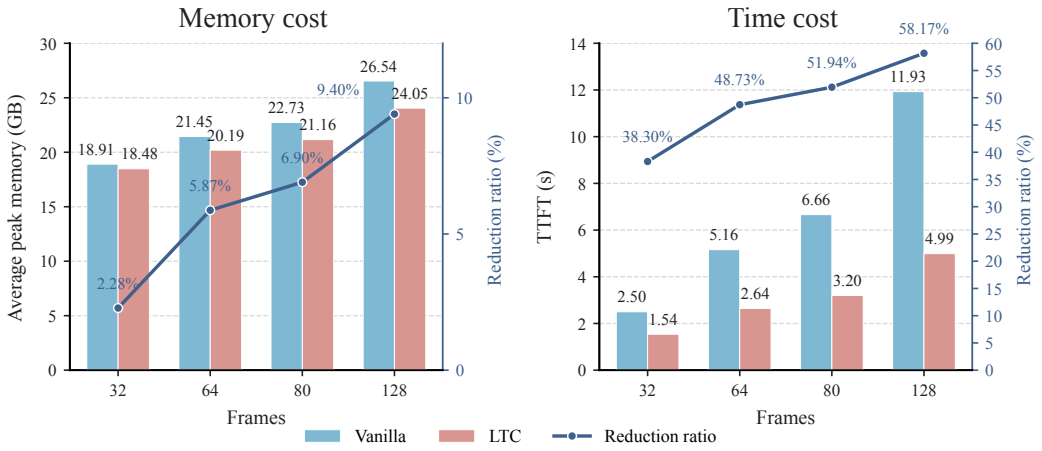


Figure 4: **Computational cost on Qwen3-VL-8B-Instruct.** Bars compare vanilla and LTC (25% token retention) in peak allocated GPU memory (left) and TTFT (right); curves show percentage reductions relative to vanilla.

To isolate the contribution of RG, we compare full LTC with ERD + LA, which retains risk scoring and quotas but omits K/U merge protection. Relative to this control, full LTC increases average accuracy from 51.7 to 52.0 and reduces mean CIER from 11.5% to 10.1%. An equal-count random gate replaces only the additional K/U protection beyond saliency protection with uniformly sampled eligible positions. Full LTC achieves higher average accuracy and lower mean CIER than this control. These gains support the contribution of RG in preventing potentially compression-sensitive tokens from being merged.

4.4 Computational cost analysis

We compare LTC at 25% token retention with the uncompressed vanilla baseline on Qwen3-VL-8B-Instruct using Time-To-First-Token (TTFT) and peak allocated GPU memory. TTFT includes compression overhead. Figure 4 summarizes the results; details are provided in Appendix C.4. As input length increases from 32 to 128 frames, LTC's TTFT reduction grows from 38.30% to 58.17%, while memory savings increase from 2.28% to 9.40%. At 128 frames, LTC reduces average TTFT from 11.93 s to 4.99 s and peak allocated GPU memory from 26.54 GB to 24.05 GB.

5 Conclusion

In this work, we propose LTC, a training-free, plug-and-play framework for visual token compression in video-LLMs. ERD diagnoses aggregation risk from neighborhood motion structure and active low-confidence signals; RGLA merges eligible tokens along motion trajectories while keeping protected tokens independent. Across three video-LLMs and three benchmarks, LTC maintains competitive accuracy with fewer visual tokens. On Qwen3-VL, it reduces latency and memory use at matched frame counts. This diagnosis-before-aggregation design reduces cross-frame redundancy while preserving critical dynamics.

AI use statement

In this work, we used GPT-5.6/6 to assist with summarizing and analyzing relevant literature, implementing the proposed method, and drafting portions of the manuscript and polishing the writing. We did not use generative AI tools to formulate mathematical claims, provide key ingredients for proving mathematical claims, assist with writing proofs, or conduct qualitative and thematic data analysis.

The authors reviewed the AI-assisted material incorporated into this work. We take responsibility for the final content of the paper, including its text, methodology, implementation, experimental results, and other materials developed with AI assistance.

Reproducibility statement

Section 3 describes the compression algorithm. Appendix A specifies the benchmark splits and reporting conventions, and Appendix B details the implementation and baseline configurations. Appendix C provides additional retention-ratio comparisons, ablations, and sensitivity analyses.

References

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

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 *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXXXI*, pp. 19–35, 2024.

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, ICLR 2026, Rio de Janeiro, Brazil, April 23-27, 2026*, 2026.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 24108–24118, 2025.

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, Vienna, Austria, July 27 - August 1, 2025*, pp. 19959–19973, 2025.

Minseok Kang, Minhyeok Lee, Jungho Lee, Minjung Kim, Donghyeong Kim, Dayeon Lee, Heeseung Choi, Ig-Jae Kim, and Sangyoun Lee. OTT-Vid: Optimal transport temporal token compression for video large language models. *CoRR*, abs/2605.11803, 2026.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker: It is better to track together. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXII*, pp. 18–35, 2024.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy visual task transfer. *Trans. Mach. Learn. Res.*, 2025, 2025.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. EViT: Expediting vision transformers via token reorganizations. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-LLaVA: Learning united visual representation by alignment before projection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 5971–5984, 2024.

Benlin Liu, Yuhao Dong, Yiqin Wang, Zixian Ma, Yansong Tang, Luming Tang, Yongming Rao, Wei-Chiu Ma, and Ranjay Krishna. Coarse correspondences boost spatial-temporal reasoning in multimodal language model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 3783–3792, 2025.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Qwen Team. Qwen3-VL technical report. *CoRR*, abs/2511.21631, 2025.

Jathushan Rajasegaran, Georgios Pavlakos, Angjoo Kanazawa, Christoph Feichtenhofer, and Jitendra Malik. On the benefits of 3d pose and tracking for human action recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 640–649, 2023.

Shuhuai Ren, Sishuo Chen, Shicheng Li, Xu Sun, and Lu Hou. TESTA: temporal-spatial token aggregation for long-form video-language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pp. 932–947. Association for Computational Linguistics, 2023.

Michael S. Ryoo, A. J. Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. TokenLearner: Adaptive space-time tokenization for videos. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 12786–12797, 2021.

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 *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, Yuxiao Dong, and Jie Tang. LVBench: An extreme long video understanding benchmark. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pp. 22958–22967, 2025.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. LongVideoBench: A benchmark for long-context interleaved video-language understanding. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, pp. 28828–28857, 2024.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 19792–19802, 2025.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. LMMs-Eval: Reality check on the evaluation of large multimodal models. In *Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pp. 881–916, 2025a.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A. Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. SparseVLM: Visual token sparsification for efficient vision-language model inference. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, pp. 74840–74857, 2025b.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. LLaVA-Video: Video instruction tuning with synthetic data. *Trans. Mach. Learn. Res.*, 2025, 2025c.

A Evaluation benchmarks and reporting conventions

A.1 Datasets and tasks

VideoMME. VideoMME contains 900 videos and 2,700 manually annotated multiple-choice questions, covering six domains, 30 content categories, and 12 task types (Fu et al., 2025). Videos range from 11 seconds to one hour. The short (less than two minutes), medium (4–15 minutes), and long (30–60 minutes) subsets each contain 300 videos. Each video has three questions with four answer options. Accuracy is computed by matching the predicted option to the reference answer. The tasks cover perception, reasoning, and summarization over multiple temporal scales.

LongVideoBench. LongVideoBench evaluates referring reasoning using 3,763 videos and 6,678 manually annotated multiple-choice questions (Wu et al., 2024). Its four duration groups cover 8–15 seconds, 15–60 seconds, 3–10 minutes, and 15–60 minutes. The benchmark contains 17 fine-grained tasks organized into perception and relation categories. Questions require locating a referred scene and recognizing objects, relating events, or reasoning across scenes. The validation set contains 752 videos and 1,337 questions, and the test set contains 3,011 videos and 5,341 questions. There is no training split. We evaluate the 1,337-question validation set. Questions have four or five answer options, and the metric is answer accuracy.

LVBench. LVBench contains 103 videos longer than 30 minutes and 1,549 four-option questions, with an average video duration of approximately 68.4 minutes (Wang et al., 2025). Tasks include entity recognition, event understanding, key-information retrieval, temporal grounding, reasoning, and summarization. These tasks require retrieving details and integrating evidence across long temporal intervals.

A.2 Aggregation and relative performance

All benchmark evaluations use lmms-eval (Zhang et al., 2025a). VideoMME Short, Medium, and Long denote duration-specific accuracy; Overall denotes accuracy over all questions. The cross-benchmark average equally weights VideoMME Overall, LongVideoBench, and LVBench:

$$\mathrm{Avg} = \frac{1}{3}\left(\mathrm{Acc}_{\mathrm{VideoMME}} + \mathrm{Acc}_{\mathrm{LongVideoBench}} + \mathrm{Acc}_{\mathrm{LVBench}}\right). \tag{8}$$

The VideoMME duration subsets are diagnostic breakdowns and are not counted as additional benchmarks in this average. Relative average performance is $100 \times \mathrm{Avg}_{\mathrm{compressed}}/\mathrm{Avg}_{\mathrm{vanilla}}$, using the vanilla reference specified in each table. For LLaVA-OneVision and LLaVA-Video, this reference uses the same backbone and input-frame count as the compressed methods. The higher-frame Qwen3-VL comparison instead uses its 32-frame vanilla reference at a matched nominal frame-equivalent visual-token budget. A per-benchmark relative score instead uses that benchmark's own vanilla score. Displayed scores retain their reported precision.

A.3 Video-level risk stratification and CIER analysis

Experimental setup. We conduct the analysis using LLaVA-OneVision-Qwen2-7B-OV with 32 uniformly sampled input frames. The target visual-token retention ratio is $r = 25\%$, corresponding to a budget of $B = \lceil 0.25N \rceil$, where N denotes the original visual-token count.

Video-level risk statistics. Video risk is computed from visual features and their correspondence estimates, without using ground-truth answers, predicted answers, or CIER values. For each token i, we compute a kinematic response and an activity-weighted low-confidence score:

$$\begin{aligned} \tilde{k}_i &= |d_i| + |\omega_i| + 0.5 s_i, \\ U_i &= (1 - c_i)\sqrt{a_i n_i}, \end{aligned} \tag{9}$$

where d_i, ω_i, and s_i denote the divergence, scalar curl, and shear magnitude derived from the residual flow, respectively. Here, c_i denotes matching confidence, while a_i and n_i are the within-video percentile ranks of activation strength and fixed-position temporal novelty. The diagnostic response $\tilde{k}_i$ is computed before the confidence weighting used in the ERD token-level risk proxy.

For each video v, $K_{\text{mean}}(v)$ is the mean kinematic response over all tokens, and $K_{\text{top10}}(v)$ is the mean response over the 10% of tokens with the largest $\tilde{k}_i$ values. $U_{\text{mean}}(v)$ is the mean activity-weighted uncertainty over all tokens.

Risk stratification. Within each model–dataset pair, we independently convert each of the three video-level statistics into percentile ranks. Let $P(\cdot)$ denote a percentile rank expressed on a $[0, 1]$ scale. The composite video-level risk score is

$$R_v = \frac{100}{3} \left[P(K_{\text{mean}}(v)) + P(K_{\text{top10}}(v)) + P(U_{\text{mean}}(v)) \right]. \tag{10}$$

Videos are assigned to five quantile groups, Q1–Q5, according to R_v, from lower to higher risk. All questions associated with the same video inherit that video's risk group, and the same grouping is used for all compression methods. The score R_v is a relative ranking statistic rather than a compression-failure probability or a calibrated predictor of downstream answer errors.

CIER computation and aggregation. The Compression-Induced Error Rate (CIER) measures the fraction of questions answered correctly by the uncompressed vanilla model that become incorrect after compression. Within each risk group, let N_{C} denote the number of matched questions answered correctly by the vanilla model, and let N_{CW} denote the number of these questions answered incorrectly after compression. CIER is computed as

$$\text{CIER} = 100 \times \frac{N_{\text{CW}}}{N_{\text{C}}}, \tag{11}$$

and is reported as a percentage. By conditioning on correct vanilla predictions, CIER measures the loss of previously correct answers after compression.

We report sample-weighted pooled CIER within each risk group. Specifically, we pool the numerator and denominator counts across datasets before computing CIER. This is equivalent to weighting each dataset's CIER by its number of matched questions answered correctly by the vanilla model in that group.

B Implementation details

B.1 Evaluation and backbone integration

Experiments use `lmms-eval` with batch size one on an NVIDIA A6000 GPU. LLaVA-OneVision uses 32 uniformly sampled frames; the main LLaVA-Video comparison uses 64, with additional 32-frame results in Appendix C.2. Qwen3-VL uses a 32-frame vanilla reference and the higher-frame settings in Section 4. Checkpoint weights, preprocessing, prompts, and generation settings are fixed within each comparison. Evaluation uses FlashAttention-2 and `decord` decoding.

LTC is applied after visual projection and spatial pooling in LLaVA, and after the visual encoder and merger in Qwen3-VL. Output anchors determine sequence repacking and positional indices; the Qwen3-VL adapter also gathers aligned DeepStack features. Text and separator tokens are outside the visual-payload budget. Attention saliency averages heads and query positions and is aligned to the pooled or merged grid. LTC operates before language-model prefill, without training or question-conditioned scoring.

B.2 LTC settings

The settings below define the default LTC configuration. They control local correspondence estimation, trajectory aggregation, protection criteria, and token-budget allocation. Sensitivity analyses for the matching-window size, matching temperature, maximum trajectory length, and risk-budget allocation are reported in Appendix C.3. The remaining parameters are fixed heuristic defaults.

Unless otherwise specified, all LTC experiments use a 5×5 matching window, a temperature of 0.07, and a 3×3 confidence-weighted smoothing kernel. The confidence mixing coefficient is $\eta = 0.35$, and the numerical stabilizer is $\epsilon = 10^{-8}$. The routing similarity threshold is 0.25 and the maximum

trajectory length $L_{\max}$ is 32. Each representative blends the member closest to its weighted mean with that mean using $\lambda = 0.65$.

The kinematic-risk weights are $(\kappa_d, \kappa_\omega, \kappa_s) = (1, 1, 0.5)$. Aggregation protection covers the upper 20% of kinematic scores, active low-confidence candidates with confidence at most 0.20, and the top 10% of informative per-frame saliency. Active low-confidence additionally requires activation at or above its median and novelty at or above its 70th percentile.

Token importance uses $(w_s, w_k, w_u, w_n) = (3, 1.5, 1.5, 1)$ for attention saliency, kinematic risk, active low-confidence, and temporal novelty, respectively. Up to four candidates per frame are reserved for temporal coverage when capacity permits. The remaining budget assigns 15% each to active low-confidence and kinematic-risk candidates, and 70% to low-risk candidates.

B.3 Baseline configurations

The following baseline configurations apply to the LLaVA-OneVision and LLaVA-Video comparisons. The reported Qwen3-VL experiment compares LTC with its vanilla reference; FlashVID and OTT-Vid were not evaluated on Qwen3-VL.

FastVID. The official wrapper uses target retention r, `DySeg_c`=8, `DySeg_tau`=0.90, `STPrune_d`=0.4, `DTM_p`=4, and `DTM_beta`=0.6.

FlashVID. Segmentation and complementary segmentation are enabled, with $\alpha = 0.70$, temporal threshold 0.8, and expansion 1.25. Both LLaVA backbones use `attn_div_v2`, at least eight segments, and layer-20 pruning with inner-language-model retention 0.3. The OneVision/LVBench setting additionally uses segmentation threshold 0.90. We match the average number of visual tokens processed per language-model layer between FlashVID and LTC, accounting for FlashVID's token expansion and inner-LLM pruning.

OTT-Vid. Temporal compression is enabled with share 0.3, mass temperature 0.3, matching-cost threshold 0.3, Sinkhorn regularization 0.01, and 200 iterations. Transport-budget temperature is 0.2 for OneVision and 0.3 for LLaVA-Video. Both LLaVA backbones use locality scale 1.0, `many_to_one` matching, `loo` mass estimation, and `position_aligned` allocation.

B.4 Risk and budget handling

Matching masks out-of-grid candidates and normalizes entropy by $\log M$ for $M > 1$, where M is the number of valid matching candidates. When $M = 1$, the entropy-based concentration component $1 - \mathcal{H}(p_t)/\log M$ is explicitly set to zero. The global translation is the component-wise median of sufficiently confident routing vectors; all vectors are used if none passes the threshold. Spatial derivatives use 3×3 Sobel filters. Constant risk, saliency, or rank fields do not trigger arbitrary protection solely through ties. Connections follow absolute flow, while risk diagnosis uses the residual flow after translation removal.

When multiple sources propose the same destination, the highest-similarity link is retained, giving each token at most one predecessor and one successor. This keeps competing correspondence paths separate.

Group importance combines maximum and mean member scores with weights 0.65 and 0.35, plus a $0.05 \log(1 + |G|)$ length bonus. Temporal coverage is reserved before allocation to the three budgeted groups. Among non-salient anchors, active low-confidence takes precedence over kinematic risk, and the remaining anchors form the low-risk group. Global importance ranking fills unused capacity from all unselected candidates, including salient anchors. Protection keeps candidates independent during aggregation, while final selection enforces the common budget. If aggregation produces fewer candidates than the budget, all are retained without duplication. Trajectory metadata supports diagnostics and alignment and is not passed through a learned trajectory encoder. Temporal novelty is computed from feature changes at the same spatial position between adjacent encoded temporal positions:

$$\nu_{t \to t+1, h, w} = \frac{1 - \cos(\mathbf{x}_{t,h,w}, \mathbf{x}_{t+1,h,w})}{2}. \tag{12}$$

Table 5: **DCS risk-statistic ablation within ERD.** Results use complete LTC on LLaVA-OneVision with 32 frames and 25% visual-token retention. All alternatives match DCS in kinematic-risk and additional protected-token counts.

Risk statistic	VideoMME	LVBench	LongVideoBench	Avg.
Matching uncertainty	58.3	39.4	56.3	51.3
Local displacement variance	58.3	40.0	56.7	51.6
Flow gradient norm	58.2	40.0	56.8	51.7
Motion magnitude	58.5	39.9	**57.0**	51.8
Div. + Curl + Shear (DCS)	**59.1**	**40.5**	56.6	**52.0**

Larger values indicate greater feature change at spatial position (h, w). For interior temporal positions, we average the scores of the preceding and following temporal pairs; boundary positions use the single available adjacent-pair score. The resulting token-level scores are converted into within-video percentile ranks n_i. These scores are used both to identify active low-confidence candidates $\mathcal{U}$ and to compute token importance for budgeted selection.

B.5 Details of Risk-Statistic Ablations

All variants retain the complete ERD and RGLA pipelines. Each alternative replaces the kinematic-risk statistic used in importance scoring, $\mathcal{K}$-category assignment, and merge protection.

Motion magnitude. This statistic measures the strength of residual motion at each token, without explicitly describing neighborhood motion structure:

$$q^{\mathrm{mag}}(p) = \|\mathbf{u}(p)\|_2. \tag{13}$$

Flow gradient norm. The Frobenius norm of the spatial Jacobian $\mathbf{J}$ measures local variation in the displacement field. Derivatives use the same 3×3 Sobel operators as DCS:

$$q^{\mathrm{grad}}(p) = \|\mathbf{J}(\mathbf{u})(p)\|_F. \tag{14}$$

Local displacement variance. This statistic measures displacement dispersion within a 3×3 neighborhood $\mathcal{N}(p)$, whose mean displacement is $\bar{\mathbf{u}}(p)$. We use replication padding and sum the component variances without taking a square root:

$$q^{\mathrm{var}}(p) = \frac{1}{|\mathcal{N}(p)|} \sum_{j \in \mathcal{N}(p)} \|\mathbf{u}(j) - \bar{\mathbf{u}}(p)\|_2^2. \tag{15}$$

Matching uncertainty. This baseline assigns higher risk to less confident token correspondences. The original active low-confidence branch remains enabled:

$$q^{\mathrm{unc}}(p) = 1 - c(p). \tag{16}$$

All experiments use LLaVA-OneVision with 32 frames and 25% visual-token retention. For each video, the numbers of kinematic-risk candidates and additional protected tokens are matched to those under DCS. TFE outputs, importance weights, category-quota rules, trajectory aggregation, temporal coverage, and budget filling remain fixed. Table 5 reports the resulting accuracies.

C Additional quantitative results

C.1 LLaVA-OneVision across retention ratios

Figure 5 extends the LLaVA-OneVision comparison to six retention ratios, from 10% to 50%. Each benchmark is normalized by its own vanilla score, while Average divides the three-benchmark mean by the vanilla mean. At 20%, 25%, 30%, and 50% retention, LTC retains 100.25%, 101.27%,

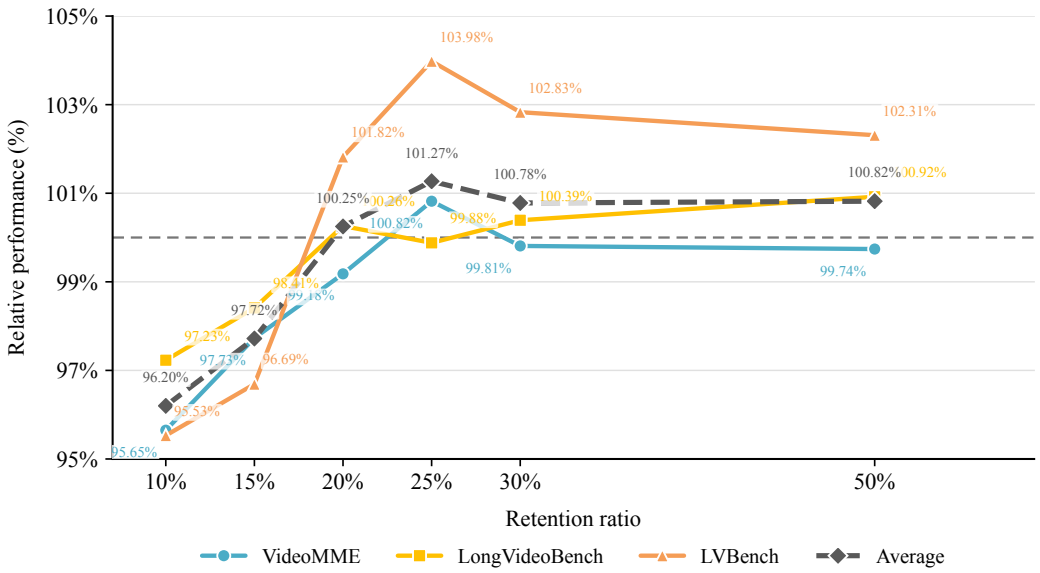


Figure 5: **LLaVA-OneVision relative performance across six visual-token retention ratios.** The dashed horizontal line denotes vanilla performance. Individual benchmarks use their own vanilla scores as denominators; Average uses the cross-benchmark vanilla mean.

100.78%, and 100.82% of vanilla average performance, respectively. The best relative average occurs at 25%, where VideoMME, LongVideoBench, and LVBench reach 100.82%, 99.88%, and 103.98% of their corresponding vanilla scores. Performance is therefore not monotonic in the number of retained tokens. At 15% and 10% retention, the relative average decreases to 97.72% and 96.20%. At 10%, the three benchmark-relative scores are 95.65%, 97.23%, and 95.53%, respectively.

C.2 Additional LLaVA-Video results

Table 6 groups the additional LLaVA-Video-7B-Qwen2 results by input-frame count. The 32-frame comparison covers retention ratios $r \in \{0.20, 0.25, 0.30, 0.50\}$; the 64-frame comparison adds 20% and 25% retention to the main results in Table 2.

With 32 frames, LTC achieves the highest three-benchmark average at 20% and 30% retention, scoring 52.0 and 52.2, respectively, while FastVID and FlashVID lead at 25% and 50%. LTC leads on LongVideoBench at 20%, 25%, and 30%, and ties the highest score at 50%. With 64 frames, LTC ties for the highest displayed average at both 20% and 25% retention, scoring 53.5 and 53.7, respectively, and achieves the highest VideoMME Short and Long scores.

C.3 Additional ablations and sensitivity analyses

Risk-stratified compression errors. Figure 6 compares the six configurations in Table 4 across five video-level risk quintiles. The quintiles are determined using pre-compression risk scores and remain fixed across configurations, from the lowest-risk Q1 to the highest-risk Q5. CIER measures the fraction of questions answered correctly by the uncompressed model that become incorrect after compression.

Across configurations, CIER generally increases toward higher-risk quintiles, indicating greater susceptibility to compression-induced errors. ERD lowers CIER relative to base filter in most quintiles. Risk-gated merge protection further reduces CIER relative to ERD + LA and its equal-count random-gate variant in Q1–Q4 under the evaluated random seed. Full LTC achieves the lowest CIER in Q1–Q3, where its improvements over base filter are most pronounced. Full LTC achieves the lowest mean CIER in Table 4, with its strongest improvements over base filter occurring in the low- and intermediate-risk groups.

Table 6: **Additional LLaVA-Video-7B-Qwen2 results with 32 and 64 input frames.** Avg. averages VideoMME Overall, LongVideoBench, and LVBench; Rel. Acc. normalizes the reported Avg. by vanilla at the same frame count.

Method	Retention ratio r	VideoMME				LVBench	LongVideo Bench	Avg.	
		Short	Medium	Long	Overall			Score	Rel. Acc. (%)
LLaVA-Video — 32 frames									
Vanilla	100%	74.2	58.4	51.2	61.3	38.9	58.0	52.8	100.0
FastVID	50%	73.9	57.6	50.2	60.6	38.7	57.4	52.2	98.9
FlashVID	50%	74.0	**57.8**	50.3	**60.7**	**38.8**	57.6	**52.4**	**99.2**
OTT-Vid	50%	72.3	56.7	**51.3**	60.1	38.0	**57.8**	52.0	98.5
LTC	50%	**74.1**	56.4	50.7	60.4	37.8	**57.8**	52.0	98.5
FastVID	30%	**73.4**	57.1	49.8	60.1	38.7	56.9	51.9	98.3
FlashVID	30%	73.0	57.2	50.4	60.2	**39.5**	56.6	52.1	98.7
OTT-Vid	30%	72.2	56.9	**51.4**	60.2	37.4	56.7	51.4	97.3
LTC	30%	73.1	**57.8**	50.7	**60.5**	37.3	**58.8**	**52.2**	**98.9**
FastVID	25%	73.1	57.6	50.1	60.3	**39.2**	57.3	**52.3**	**99.1**
FlashVID	25%	72.7	**57.9**	50.7	**60.4**	39.1	57.1	52.2	98.9
OTT-Vid	25%	71.8	56.6	50.4	59.6	37.4	56.6	51.2	97.0
LTC	25%	**73.2**	57.3	**50.8**	**60.4**	37.2	**58.1**	51.9	98.3
FastVID	20%	72.1	56.7	49.1	59.3	**38.3**	56.5	51.4	97.3
FlashVID	20%	72.2	**59.2**	49.8	**60.4**	**38.3**	56.5	51.8	98.1
OTT-Vid	20%	70.9	56.6	51.3	59.6	37.6	56.7	51.3	97.2
LTC	20%	**72.7**	56.4	**51.7**	60.3	38.0	**57.7**	**52.0**	**98.5**
LLaVA-Video — 64 frames									
Vanilla	100%	76.3	61.3	53.4	63.7	40.6	59.3	54.5	100.0
FastVID	25%	74.0	60.7	51.7	62.1	40.5	58.4	**53.7**	**98.5**
FlashVID	25%	74.4	**61.4**	51.3	**62.4**	39.6	59.1	**53.7**	**98.5**
OTT-Vid	25%	74.1	60.0	51.7	61.9	38.8	**59.4**	53.4	98.0
LTC	25%	**75.0**	59.2	**52.4**	62.2	**40.6**	58.3	**53.7**	**98.5**
FastVID	20%	73.6	**60.3**	51.7	61.9	**41.0**	56.8	53.2	97.6
FlashVID	20%	73.6	59.4	51.8	61.6	40.7	58.3	**53.5**	**98.2**
OTT-Vid	20%	73.8	59.4	50.3	61.2	39.2	**58.6**	53.0	97.2
LTC	20%	**74.6**	**60.3**	**52.4**	**62.4**	40.3	57.7	**53.5**	**98.2**

In the highest-risk quintile, CIER remains elevated across configurations, and the gains from risk-gated merge protection diminish and become less consistent. ERD alone achieves lower CIER than full LTC in Q4–Q5, indicating that selecting individual tokens without trajectory aggregation better preserves previously correct predictions in these groups.

Allocation to risk categories. Table 7 varies the fraction α of the category budget assigned to risk categories after temporal-coverage reservation. The main setting $\alpha = 0.30$ assigns 15% each to kinematic-risk and active low-confidence candidates, and 70% to low-risk candidates, achieving the highest average accuracy of 52.0. Both $\alpha = 0$ and $\alpha = 0.15$ yield 51.6, while increasing α to 0.45, 0.60, and 1.00 yields 51.9, 51.5, and 51.2, respectively. These results favor balancing risk-sensitive retention with preservation of low-risk content under a fixed token budget.

Local matching window. The matching window controls the spatial search range for token correspondences between adjacent frames. Table 8 compares square windows with side lengths 1, 3, 5, and 7. A side length of one restricts correspondence search to the same grid position, whereas larger windows permit cross-position matching. The 5×5 window achieves the highest average

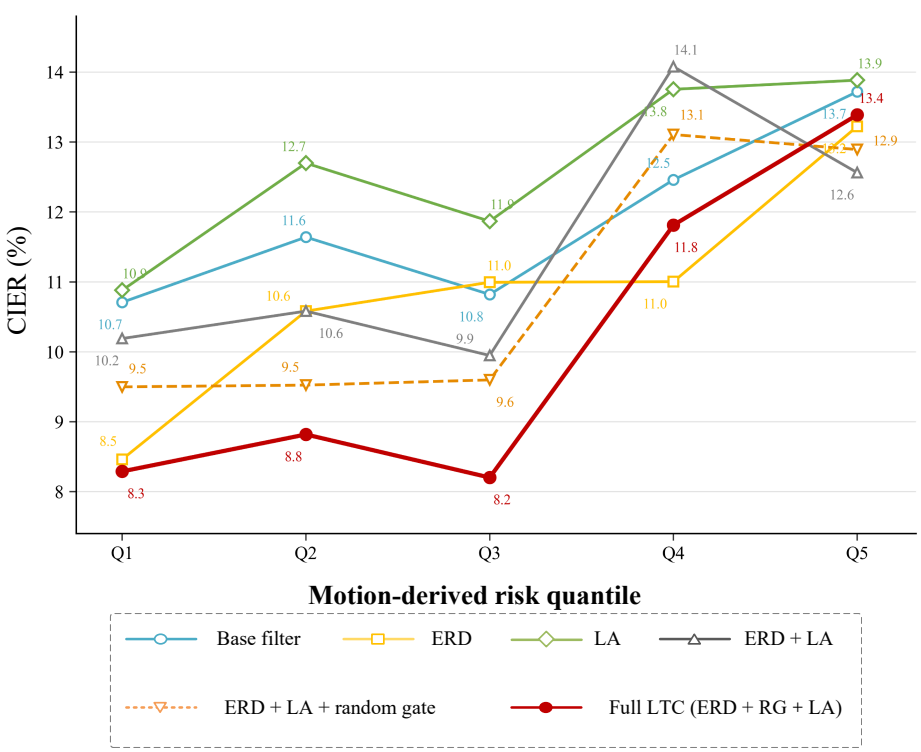


Figure 6: **Compression-Induced Error Rate (CIER) across video-level risk quintiles on LLaVA-OneVision at 25% visual-token retention.** All configurations use the same risk-group assignments and vanilla-correct reference questions. Q1–Q5 represent increasing risk, and lower CIER is better. All settings share TFE. ERD + LA omits K/U merge protection, while its random-gate variant replaces the additional K/U protection positions with an equal number of randomly selected positions (seed 1234). Full LTC uses K/U-guided merge protection.

Table 7: **Risk-budget allocation at a fixed total token budget on LLaVA-OneVision at 25% retention.** After temporal-coverage reservation, the total risk share α is split equally between kinematic-risk and active low-confidence candidates; low-risk candidates receive $1 - \alpha$.

α	Kinematic risk	Active low confidence	Low risk	VideoMME	LVBench	LongVideo Bench	Avg.
0.00	0.000	0.000	1.000	57.7	39.7	57.5	51.6
0.15	0.075	0.075	0.850	58.1	39.3	**57.6**	51.6
0.30 (main)	0.150	0.150	0.700	**59.1**	**40.5**	56.6	**52.0**
0.45	0.225	0.225	0.550	59.0	40.4	56.4	51.9
0.60	0.300	0.300	0.400	58.4	39.5	56.6	51.5
1.00	0.500	0.500	0.000	58.5	39.1	56.1	51.2

accuracy of 52.0. It obtains the best VideoMME and LVBench scores and ties the 7×7 setting on LongVideoBench. The results support local cross-position matching while showing that a larger search range does not necessarily improve performance.

Matching temperature. The matching temperature τ_m controls the sharpness of the local correspondence distribution in TFE. Table 9 compares five temperature values while keeping all other settings fixed. The default setting, $\tau_m = 0.07$, achieves the highest average accuracy of 52.0, together with the best scores on VideoMME and LVBench. LongVideoBench instead achieves its highest score at $\tau_m = 0.14$. The results show a non-monotonic relationship between temperature and accuracy, with different benchmarks favoring different temperatures. Among the tested values, $\tau_m = 0.07$

Table 8: **Sensitivity to the TFE matching-window side length on LLaVA-OneVision at 25% retention.** The main setting is 5×5.

Window side	VideoMME	LVBench	LongVideoBench	Avg.
1	58.4	39.0	56.2	51.2
3	58.6	38.5	56.2	51.1
5 (main)	**59.1**	**40.5**	**56.6**	**52.0**
7	57.9	39.8	**56.6**	51.4

Table 9: **Sensitivity to the TFE matching temperature on LLaVA-OneVision at 25% retention.** The matching window is fixed at 5×5, and the main temperature setting is $\tau_m = 0.07$.

τ_m	VideoMME	LVBench	LongVideoBench	Avg.
0.03	58.4	39.6	57.2	51.8
0.05	58.1	38.5	57.0	51.2
0.07 (main)	**59.1**	**40.5**	56.6	**52.0**
0.10	57.7	39.8	57.1	51.5
0.14	58.1	38.8	**57.7**	51.6

provides the best average performance across the three benchmarks, supporting its use as the default setting.

Maximum trajectory length. Table 10 examines the effect of $L_{\max}$ on compression performance. Average accuracy varies little as $L_{\max}$ increases from 8 to 32. We use $L_{\max} = 32$ as the default throughout our experiments. Actual trajectory lengths remain constrained by risk gating, feature consistency, and valid links, so this setting permits longer aggregation without requiring trajectories to reach the maximum length.

Composition of a fixed risk quota. While Table 7 varies the total risk allocation, Table 11 holds it fixed at 30%. K-only assigns the entire risk quota to kinematic-risk candidates; U-only assigns it to active low-confidence candidates. The joint setting assigns 15% to each. All three assign the remaining 70% to low-risk candidates. The joint setting achieves an average accuracy of 52.0, exceeding K-only and U-only by 0.7 and 0.2 percentage points, respectively. On LVBench, the corresponding gains are 1.6 and 0.4 points. At a fixed total risk budget, the combined allocation yields the strongest cross-benchmark average, supporting the use of both neighborhood kinematics and active low-confidence information in representative selection.

C.4 Additional efficiency results

Table 12 compares latency, LLM FLOPs, and peak allocated GPU memory on Qwen3-VL-8B-Instruct with 32, 64, 80, and 128 input frames. Vanilla uses all visual tokens, while LTC retains 25%.

D Qualitative results

The frame strips in Figures 7 and 8, and the thumbnails in Figure 9, are contiguous subsequences of the uniformly sampled model-input frames. The LLaVA-Video examples use $F = 64$ input frames, while the LLaVA-OneVision examples and the risk visualization use $F = 32$.

D.1 Answer comparisons

Green annotations indicate correct answers, and red annotations indicate incorrect answers in the following answer-comparison figures.

Table 10: **Sensitivity to the maximum trajectory length on LLaVA-OneVision at 25% retention.** The main experimental setting uses $L_{\max} = 32$.

$L_{\max}$ (tokens)	VideoMME	LVBench	LongVideoBench	Avg.
4	58.6	39.5	55.5	51.2
8	**59.1**	39.8	**57.2**	**52.0**
16	59.0	40.4	56.6	**52.0**
32 (main)	**59.1**	**40.5**	56.6	**52.0**

Table 11: **Risk-category composition at a fixed 30% risk allocation on LLaVA-OneVision at 25% retention.** K and U denote the kinematic-risk and active low-confidence candidate sets, respectively.

Configuration	K quota	U quota	VideoMME	LVBench	LongVideo Bench	Avg.
K-only	30%	0%	57.8	38.9	**57.3**	51.3
U-only	0%	30%	**59.1**	40.1	56.3	51.8
Joint (main)	15%	15%	**59.1**	**40.5**	56.6	**52.0**

Table 12: **Detailed efficiency comparison on Qwen3-VL-8B-Instruct.** Mean latencies, LLM FLOPs, and peak allocated GPU memory across VideoMME, LongVideoBench, and LVBench. Parentheses give speedups relative to vanilla at the same frame count, computed as vanilla latency divided by the corresponding method latency.

F	Method	Retention ratio r	Latency (ms)			LLM cost (TFLOPs)	Peak allocated memory (GB)
			LLM forward	TTFT	Total		
32	Vanilla	100%	1758.3	2497.4 (1.00×)	3100.9 (1.00×)	188.125	18.909
	LTC	25%	534.9	1540.9 (1.62×)	2105.5 (1.47×)	41.669	18.478
64	Vanilla	100%	3803.8	5158.6 (1.00×)	5774.4 (1.00×)	469.101	21.453
	LTC	25%	915.1	2644.8 (1.95×)	3191.1 (1.81×)	88.177	20.193
80	Vanilla	100%	5008.5	6659.3 (1.00×)	7245.1 (1.00×)	645.047	22.725
	LTC	25%	1113.5	3200.3 (2.08×)	3704.8 (1.96×)	113.886	21.156
128	Vanilla	100%	9366.1	11929.8 (1.00×)	12507.8 (1.00×)	1314.728	26.541
	LTC	25%	1770.2	4990.0 (2.39×)	5539.1 (2.26×)	200.838	24.046

LLaVA-Video. Figure 7 shows three answer-comparison examples. In the kitchen scene, LTC selects "washing the potatoes", while all three baselines select "cutting the potatoes". Distinguishing these actions requires interpreting hand–object interactions over time. For the question about why a woman is crying, LTC selects "she felt foolish", while the baselines select "insufficient information to determine", illustrating disagreement in contextual interpretation. In the action-order example, LTC selects unfurling a banner, catching balls, and then tumbling from the platform; the baselines place the fall before the other events. This question depends on temporal ordering among the same set of events.

LLaVA-OneVision. Figure 8 contains examples of cross-frame entity counting, tool retrieval, and time-specific score reading. LTC selects three polar bears, while the baselines select four, illustrating the need to associate entities across frames. In the rubber-band car example, LTC selects "pencil" as the unnecessary tool, while the baselines select "hammer"; the question combines retrieval over the construction process with a negative condition. For the halftime basketball score, LTC selects 38–34, OTT-Vid selects 67–61, and FlashVID and FastVID select 67–60. Answering requires aligning



Question: In a room with white tiles, there is a gray countertop. Next to the countertop stands a man wearing a blue checkered shirt, with tattoos on his arms. Beside him, there is a white bowl containing potatoes. What is he doing?

Options:
A. Washing the tomatoes
B. Peeling the potatoes
C. Cutting the potatoes **FlashVID OTT-Vid FastVID** ✗
D. Washing the green peppers
E. Washing the potatoes **LTC** ✓



Question: What was the reason for the woman in the video crying?

Options:
A. She was unhappy with the man.
B. She found the man in the video to be foolish.
C. She felt foolish. **LTC** ✓
D. Insufficient information to determine. **FlashVID OTT-Vid FastVID** ✗



Question: What sequence of actions followed after they stepped on the stilts?
(a) They unfurled a banner from their mouths. (b) They tumbled down from the high platform.
(c) They caught the balls thrown by the ground staff.
Options:
A. (a)(b)(c).
B. (a)(c)(b). **LTC** ✓
C. (b)(c)(a).
D. (b)(a)(c). **FlashVID OTT-Vid FastVID** ✗

Figure 7: **Answer comparisons on LLaVA-Video.** The three examples concern action recognition, contextual understanding, and event ordering. Green annotations indicate correct answers, and red annotations indicate incorrect answers. Each frame strip above a question shows consecutive frames selected from the uniformly sampled sequence of $F = 64$ model inputs.

fine-grained numeric evidence with the requested time interval. These selected examples illustrate method outputs; aggregate performance is assessed by the benchmark results.

D.2 Frame-level risk and retained tokens

Figure 9 visualizes frame-level risk and retained-token counts on LLaVA-OneVision with $F = 32$ uniformly sampled input frames for each of three example videos. The displayed thumbnails are consecutive frames selected from this sampled input sequence. Risk varies across the sampled timeline, and the retained representation is distributed across frames. Allocation also depends on visual importance, trajectory formation, category quotas, and the global budget, so the risk curve alone does not determine the number of retained tokens.

E Limitations and future work

Correspondence range and trajectory scale. LTC estimates adjacent-frame correspondences on a fixed visual grid and forms trajectories using discrete routing. The reported main configuration uses a 5×5 matching window and a maximum trajectory length of 32. The search window, frame-sampling interval, and spatial resolution jointly constrain correspondence coverage. Multi-scale matching, adaptive search windows, and hierarchical trajectories could extend coverage for larger inter-frame displacements and longer temporal spans while controlling overhead.



Question: How many polar bears are visible/seen in the video?

Options:
A. 4. **FlashVID OTT-Vid FastVID** ✗
B. 3. **LTC** ✓
C. 1.
D. 2.



Question: As depicted in the video, which tool is not necessary to make a rubber band car?

Options:
A. Straw.
B. Pencil. **LTC** ✓
C. Scissors.
D. Hammer. **FlashVID OTT-Vid FastVID** ✗



Question: What is the score at the end of the half?

Options:
A. 38 - 31.
B. 38 - 34. **LTC** ✓
C. 67 - 61. **OTT-Vid** ✗
D. 67 - 60. **FlashVID FastVID** ✗

Figure 8: **Answer comparisons on LLaVA-OneVision.** The examples concern cross-frame entity counting, tool retrieval, and reading a score at a specified time. Green annotations indicate correct answers, and red annotations indicate incorrect answers. Each displayed frame strip is a contiguous subsequence of the $F = 32$ uniformly sampled model inputs.

Structural risk and task relevance. ERD provides a proxy based on local correspondence structure and matching reliability. Its value is not a calibrated probability of downstream answer error. Current decisions depend on visual features, correspondence estimates, and category budgets, and are not conditioned on the question text. Question-conditioned risk weighting and budget allocation may help align structural stability and visual saliency with the information required by a particular query.

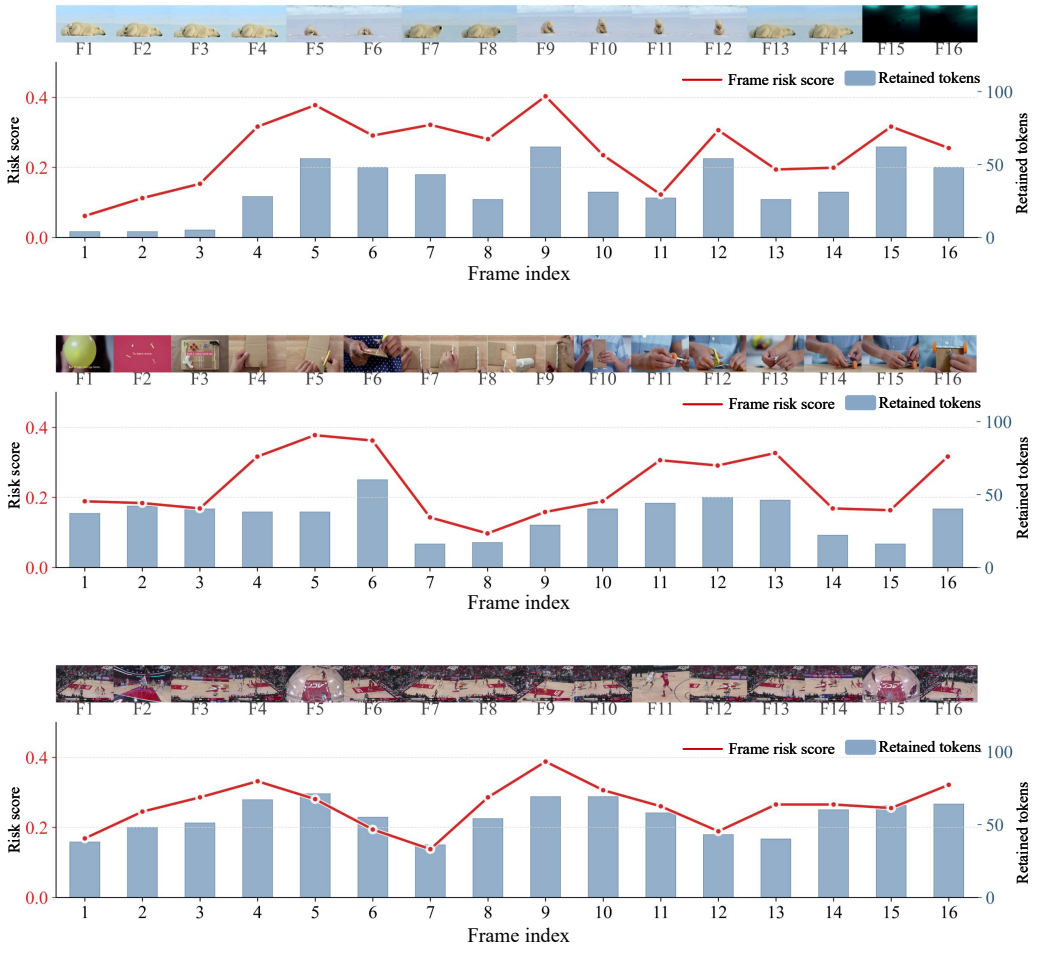


Figure 9: **Frame-level risk scores and retained-token counts on LLaVA-OneVision with $F = 32$ uniformly sampled input frames for each of three example videos.** Red curves use the left axes and blue bars use the right axes. Thumbnails show consecutive frames selected from the uniformly sampled model-input sequence. These traces visualize temporal allocation and do not establish a causal relation between risk and task errors.