CoGCo: Codec Guided Training-free Video-LLM Token Compressor

Anonymous authors
Paper under double-blind review

Abstract

Video Large Language Models (Video LLMs) process thousands of visual tokens from sampled frames, making redundant video representations a major source of inference cost. Training-free video token compression must preserve persistent scene context and subtle temporal changes, yet token saliency alone does not reveal which information requires an additional representative. We present CoGCo, which exploits the structure of standard video codecs to separate these complementary responsibilities. Reference-frame tokens establish semantic anchors through attention- and diversity-aware selection, supplemented by reference locations supported by subsequent updates. Prediction-frame selection combines camera-compensated motion vectors, residuals, and within-frame diversity to preserve complementary temporal evidence. It excludes a small fraction of the highest-attention tokens from independent selection, as our analysis suggests that such features primarily capture overall semantic context rather than fine-grained updates. Moreover, we leverage codec frame packets to efficiently allocate the token budget. Across four benchmarks and four backbones, CoGCo achieves higher aggregate accuracy than the SOTA training-free compression baselines at evaluated retention settings of 15%, 20%, and 25%.

1 Introduction

Video Large Language Models (Video LLMs) have emerged as a powerful paradigm for understanding dynamic visual content, enabling video question answering, fine-grained motion reasoning, and long-video analysis (Li et al., 2024; Zhang et al., 2024; Bai et al., 2025; Wang et al., 2025). Most existing Video LLMs process videos through a frame-based paradigm: they sample RGB frames from a video, encode each frame into patch-level visual features, project them into the language-model space, and concatenate the resulting visual tokens with text tokens for autoregressive reasoning. While effective, this paradigm causes the visual context to grow rapidly with the number of sampled frames. Longer or more densely sampled videos can therefore introduce thousands of visual tokens, substantially increasing prefilling and memory costs. Efficiently reducing redundant visual tokens is therefore critical to scaling Video LLMs toward longer and finer-grained video understanding.

This bottleneck has motivated growing interest in visual token compression, particularly training-free approaches that reduce inference cost while keeping increasingly large pretrained Video LLMs unchanged. Existing methods broadly follow two strategies. One line identifies redundant tokens from model-internal semantic signals, such as attention, visual features, or token similarity, and prunes or merges less informative tokens Chen et al. (2024); Yang et al. (2025); Huang et al. (2024); Shen et al. (2025). While effective at preserving semantically important content, these signals do not explicitly indicate whether the content has changed or has already been sufficiently represented over time, as shown in Figure 1. A second line more explicitly exploits cross-frame redundancy through temporal grouping, similarity-based merging, adaptive token allocation, or feature-derived temporal structures Fu et al. (2025); Hyun et al. (2025); Shao et al. (2025); Yuan et al. (2026); Ju et al. (2026); Fan et al. (2026); Tao et al. (2024). However, the temporal relationships used for compression are still largely inferred from learned visual representations, making it difficult to explicitly distinguish information that should persist across frames from genuinely new temporal updates. Consequently, existing methods remain centered on model-derived notions of importance and redundancy, providing limited explicit guidance on what should persist as reference context and what should receive additional representation as a temporal update.

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: Token selection visualization at 10% retention. FlashVID fails to select tokens in the hand region.

We note that video codecs used in standards such as H.264/AVC provide a natural and efficient cue of visual information over time. Modern codecs preserve self-contained reference content while exploiting inter-frame predictability to represent subsequent frames through prediction, motion, and residual corrections Wiegand et al. (2003); International Telecommunication Union (2020). This structure is particularly well suited to video token compression. Reference I frames provide stable visual context for preserving important semantic appearance. Predictive P frames, which are encoded by referencing preceding frames through motion-compensated prediction and residual corrections, further organize visual information according to its temporal role: what persists as semantic context and what constitutes a new update over time. It therefore provides a richer, video-native cue of semantic information that captures not only *what is important*, but also *what is already represented* and *what has changed*. This distinction is crucial for video compression: a salient object may remain important throughout a sequence, yet repeatedly preserving its unchanged appearance contributes little new information; conversely, a small object motion, local state change, or action transition may have modest frame-level saliency while providing critical evidence for temporal reasoning. Codec prediction naturally distinguishes these cases by reusing predictable appearance while exposing temporal updates through motion and residuals. Moreover, because codec is derived from the video itself rather than from any particular model representation, it provides a model-independent temporal organization that can be readily applied across different Video LLM backbones.

However, codec evidence and visual-token features describe video content from different perspectives: codec cues are spatially and temporally grounded, whereas visual features become contextualized through the visual encoder. Although each visual token remains associated with a spatial position, its feature is therefore not confined to the local content of the corresponding image patch. Our analysis suggests that some high attention features primarily capture broad semantic context rather than localized information. Such features are useful for establishing I-frame semantic anchors, but repeatedly allocating independent P-frame tokens to them can consume capacity better used for complementary updates. This motivates treating semantic reference context and predictive update evidence according to their distinct roles rather than applying a uniform selection criterion.

Based on this insight, we propose CoGCo, a training-free visual token compression framework that organizes existing semantic tokens around codec reference–update roles. CoGCo comprises four main components: **(1) Codec-guided token selection.** I-frames provide attention-guided, diverse appearance anchors, while residual cues and camera-compensated motion expose P-frame update candidates for within-frame selection. To reserve capacity for fine-grained updates, we exclude a small fraction of the high-attention P-frame positions from selection. **(2) Codec Packet-aware budget allocation.** Packet-aware allocation coordinates appearance and update budgets. **(3) Global Completion** then fills the remaining budget globally. **(4) Codec-guided token aggregation.** Once tokens are finalized, non-selected tokens are incorporated into retained representatives using spatial, codec-derived temporal, and feature similarity. Together, these components preserve appearance context, prioritize temporal updates, and absorb compatible redundancy under a fixed visual-token budget, without changing the visual encoder or LLM weights.

We evaluate CoGCo on LLaVA-OneVision, LLaVA-Video, InternVL3.5, and Qwen3-VL across VideoMME, MVBench, MotionBench, and LongVideoBench. Across the evaluated retention settings, CoGCo improves aggregate accuracy over the compared training-free compression baselines. Notably,

compared to the vanilla LLaVA-OneVision baseline, it achieves higher average accuracy across the four benchmarks while retaining only 15% of the tokens.

To conclude, our main contributions are as follows:

- We introduce CoGCo, a training-free framework that uses the video codec reference-update structure to compress vision tokens in Video LLMs.
- We develop a role-specific selection strategy that couples I-frame semantic anchoring with subsequent update evidence while prioritizing complementary P-frame updates over repeated semantic context. This coordination preserves persistent visual context and reserves limited token capacity for fine-grained temporal evidence.
- Extensive experiments across four Video LLMs, four benchmarks, and multiple compression budgets demonstrate consistent improvements over SOTA training-free baselines.

2 Related Work

2.1 Video Coding and Codec Representations

Video coding exploits spatial and temporal redundancy to reduce storage and transmission costs. Early standards such as H.261 established hybrid block-based coding, combining motion-compensated prediction with transform coding (Sullivan et al., 2012). I-frames encode appearance, whereas P-frames predict content from previously decoded reference pictures. Motion vectors specify reference locations, while the remaining prediction errors are encoded as residuals through transform, quantization, and entropy coding. This prediction-and-correction principle is retained in modern standards including H.264/AVC, H.265/HEVC, and H.266/VVC, despite increasingly sophisticated partitioning and prediction tools (Wiegand et al., 2003; Sullivan et al., 2012; Bross et al., 2021). Their bitstreams therefore contain coded appearance, motion, residual, and reference information that can support Video LLM vision token compression beyond pixel reconstruction.

2.2 Visual Token Compression for Video LLMs

Training-free visual token compression reduces computation by selecting or aggregating tokens according to importance and redundancy. Attention-based pruning and attention–feature selection preserve salient or representative tokens before or inside the LLM (Chen et al., 2024; Yang et al., 2025; Huang et al., 2024). Feature similarity identifies redundant tokens that can share a representative (Yang et al., 2025), while diversity-based selection preserves coverage of distinct visual evidence rather than repeatedly retaining similar high-scoring tokens (Alvar et al., 2025). Cross-frame fusion and spatiotemporal merging exploit repeated content across neighboring frames, with compression extending from the vision encoder to decoding-time caches (Wang et al., 2026; Tao et al., 2024; Fu et al., 2025; Hyun et al., 2025). Temporal segmentation, density-based selection, and reconstruction-oriented uniqueness further coordinate coverage and token budgets across a video (Shen et al., 2025; Shao et al., 2025; Yuan et al., 2026). Tree- and forest-based approaches explicitly organize token correspondences using feature similarity and spatial–temporal constraints (Ju et al., 2026; Fan et al., 2026). Despite differences in compression stage and grouping granularity, these approaches largely infer importance and redundancy from learned model representations.

However, attention-based importance may overlook whether content has already been sufficiently represented. Under limited budgets, repeatedly salient appearance can crowd out subtle updates, while similarity-based pruning or merging risks suppressing discriminative motion and state changes. CoGCo complements these signals with codec reference-update structure, preserving appearance anchors while reserving capacity for informative temporal updates.

2.3 Codec-Based Video Understanding

Codec representations have long been exploited for traditional video understanding. Early compressed-domain methods treat I-frame appearance, motion vectors, and residuals as complementary inputs, using motion-vector networks, multi-stream processing, and learned motion refinement (Zhang et al., 2016; Wu et al., 2018; Shou et al., 2019). Teacher-student learning and multimodal attention further improve compressed-domain representations (Battash et al., 2020; Chen & Ho, 2022). Codec

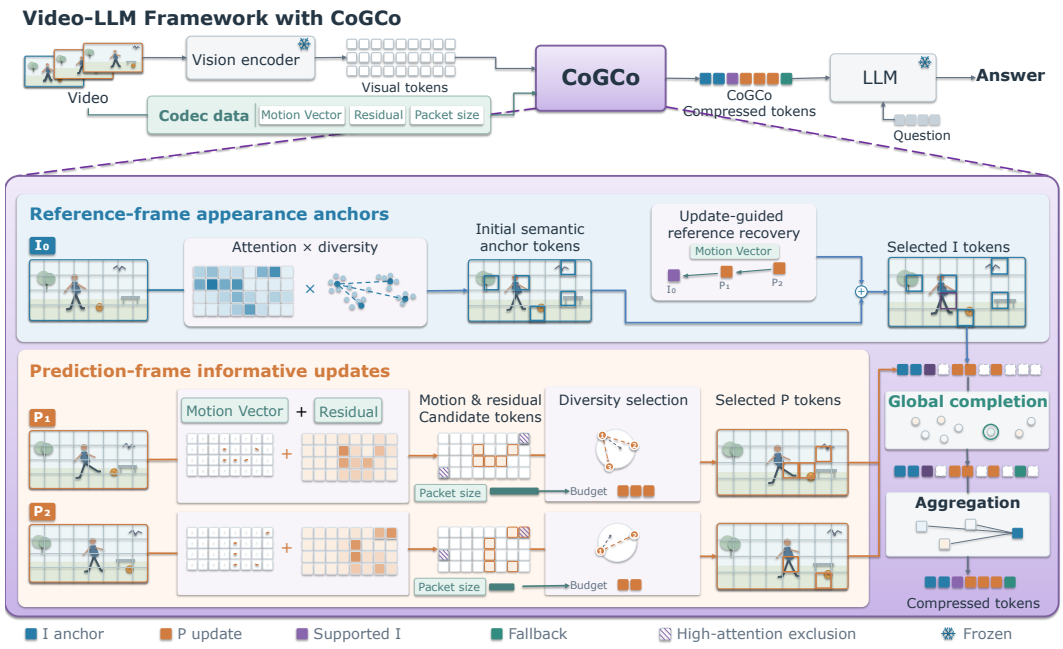


Figure 2: Overview of CoGCo. Codec guides visual-token compression before the LLM. I-frames form attention-guided semantic anchors, while motion and residual cues identify P-frame update candidates; high-attention positions are excluded in P-frames and packet costs guide P-frame budgets. Later update evidence recovers overlooked reference locations, followed by global completion and compatible feature aggregation.

primitives also support pose estimation, event-boundary detection, and video object segmentation (Fan et al., 2021; Li et al., 2022; Xu & Yao, 2022). These task-specific approaches establish the usefulness of codec information for visual understanding.

Recent codec-aware Video LLMs combine sparse RGB appearance with learned GOP-level motion features or lightweight motion/residual representations (Zhao et al., 2025; Sarkar et al., 2026; Yashima et al., 2026). Others redesign the visual interface through codec-aligned sparse encoders, compact predictive tokens, or codec-native tokenizers (Tang et al., 2026; Hou et al., 2026; Yang et al., 2026). Their common emphasis is constructing or learning visual representations from codec information through dedicated encoders, pretraining, or multimodal fine-tuning. However, CoGCo uses codec information to organize and compress existing visual tokens which brings codec-derived temporal cues to pretrained semantic representations without requiring additional model training

3 METHOD

As illustrated in Figure 2, CoGCo uses codec data to compress the visual tokens produced by the vision encoder before they enter the LLM. The framework comprises four main components. **(1) Codec-guided token selection** establishes semantic anchors on reference frames using attention and feature diversity, while subsequent update evidence helps recover overlooked reference locations. For prediction frames, it selects complementary update tokens based on motion, residuals, and feature diversity, while excluding a small fraction of high-attention positions from selection to focus limited P-frame capacity on complementary updates. **(2) Packet-aware budget allocation** efficiently coordinates reference and update capacity by reserving reference tokens and distributing P-frame budgets according to codec packet costs. **(3) Global completion** fills unused capacity with complementary tokens from across the video, finalizing the representative set. **(4) codec-guided token aggregation** incorporates compatible non-selected features into the retained representatives using spatial, codec, and feature-similarity relationships.

3.1 CODEC-GUIDED TOKEN SELECTION

To separately capture static visual context and dynamic scene updates, we first preserve reference appearance and predictive changes through role-specific selection.

3.1.1 REFERENCE-FRAME APPEARANCE ANCHORS

Initial semantic anchors. Reference appearance supplies context for interpreting later changes, but repetitive responses can consume its anchor budget. We combine attention with feature distinctiveness. Given within-frame rank-and-magnitude calibrated attention $\widetilde{a}_{t,i}$, and denoting cosine distance between original visual features by $d(\cdot,\cdot)$, the initial I-frame score is

$$u_{t,i} = \min_{j \neq i} d\big((t,i),(t,j)\big), \qquad J^I_{t,i} = \widetilde{a}_{t,i} \max(u_{t,i}, \varepsilon). \tag{1}$$

Here $u_{t,i}$ measures nearest-neighbor feature distinctiveness and $\varepsilon > 0$ is a small positive floor. The resulting score initializes attention-calibrated max-min diversity selection: after the first anchor, the diversity term is updated against the currently selected I-frame anchors, favoring tokens that remain both semantically important and feature-distinct from the retained set. Selection continues within the offered I-frame budget to form the initial anchor set $S^{I,0}_t$.

Update-guided reference recovery. Initial anchors capture salient and diverse I-frame appearance, but may miss reference locations that become important for interpreting later changes. We therefore trace P-frame update evidence back to the corresponding I-frame locations through codec prediction chains. Supported but unselected I-frame positions form a video-wide candidate pool, from which we select additional reference tokens by feature diversity against all retained representatives, subject to the remaining token budget. Support computation and selection details are provided in Appendix B.2.

3.1.2 PREDICTION-FRAME INFORMATIVE UPDATES

Appearance anchors preserve reference context, but do not indicate which information must be newly represented in subsequent frames. We therefore use codec residual and motion evidence to identify locations that contain potential updates in P-frames. Since high-attention visual features tend to capture broad semantic context rather than localized updates, we preserve their role as reference context but restrict a small fraction from becoming independent P-frame representatives. Within the resulting candidate pool, diversity is then used to select complementary update representatives.

Update candidates. Let $r_{t,i}$ be the nonnegative, patch-pooled cue from the decoder's residual buffer. To reduce coherent camera motion, we estimate an affine field $\widehat{\mathbf{m}}_t(\mathbf{p})$ using low-residual positions. The compensated motion magnitude and raw update evidence are

$$o_{t,i} = \left|\mathbf{m}_{t,i} - \widehat{\mathbf{m}}_t(\mathbf{p}_i)\right|_2, \qquad e_{t,i} = r_{t,i} + o_{t,i}. \tag{2}$$

We form the P-frame update candidate pool $\mathcal{A}_t$ by thresholding the raw evidence $e_{t,i}$, retaining positions with strong codec-indicated update evidence. The threshold and candidate-pool safeguards are detailed in Appendix B.1.

Role-specific high attention exclusion. Because high attention features primarily carry overall semantic context rather than local information that requires additional P-frame capacity, we exclude the top-δ attention fraction of each P-frame from independent update selection, denoting these positions by $\mathcal{E}_t$. The resulting eligible update pool is $\mathcal{C}_t = \{i \in \mathcal{A}_t \mid i \notin \mathcal{E}_t\}$. This restriction prevents these positions from becoming independent P-frame representatives, including during global completion, while their features remain available for compatible aggregation. The diagnostic analysis, sensitivity experiments on δ, and implementation details are provided in Appendix A.3.

Within-frame diversity selection. Within each P-frame, we order the eligible update candidates by greedy max-min feature diversity. The first candidate is initialized as the one with the greatest mean cosine distance to the others, and each subsequent candidate maximizes its minimum distance to the already selected candidates:

$$i_1 = \arg\max_{i \in \mathcal{C}_t} \frac{\sum_{j \in \mathcal{C}_t \setminus \{i\}} d_t(i,j)}{|\mathcal{C}_t| - 1}, \qquad i_{\ell+1} = \arg \max_{i \in \mathcal{C}_t \setminus \mathcal{S}^{P,\ell}_t} \min_{j \in \mathcal{S}^{P,\ell}_t} d_t(i,j), \tag{3}$$

where $d_t(i,j) = d((t,i),(t,j))$ and $\mathcal{S}_t^{P,\ell}$ contains the first ℓ candidates in this ordering. The number of retained candidates is determined by the GOP-aware allocation described in Section 3.2.

3.2 Packet-aware budget allocation

Role-specific selection determines which visual evidence should be retained within each frame, but not how the fixed token budget should be distributed across the video. We therefore allocate capacity hierarchically: first across prediction groups while preserving reference capacity, and then among P-frames using codec packet costs to reflect their relative update demand.

We first distribute the total budget across prediction groups in proportion to their original visual-token counts, giving each group a budget B_g with $\sum_g B_g = B$. Within each group, the I-frame is first assigned an anchor budget, while the remaining capacity is shared by its P-frames. Let A_g denote the number of accepted I-frame anchors and n_g^P the number of P-frames. We cap the total P-frame budget as

$$C_g^P = \min\left\{ B_g - A_g,\ \left\lfloor \frac{n_g^P B_g}{n_g^P + \alpha_{\min}} \right\rfloor \right\}, \tag{4}$$

where $\alpha_{\min}$ reserves sufficient capacity for reference content.

We further use the codec packet cost as a lightweight cue for relative update demand, as higher coding cost generally indicates a larger amount of frame-specific update information. However, allocating the budget solely according to packet cost could under-represent low-cost frames that still contain useful visual evidence. We therefore combine uniform coverage with packet-adaptive allocation. For $t \in \mathcal{P}_g$, we define

$$w_t = (1-\lambda) + \lambda \frac{\hat{q}_t}{\bar{\hat{q}}_{\mathcal{P}_g}}, \tag{5}$$

where $\hat{q}_t = \log(1 + q_t/m_g)$ is the median-normalized packet cost, $m_g = \max\{\mathrm{median}_{u\in\mathcal{P}_g} q_u, 1\}$, and $\bar{\hat{q}}_{\mathcal{P}_g}$ is its group mean. Following prior allocation schemes that balance uniform coverage with adaptive demand (Hu et al., 2012; Feng et al., 2025; Wu et al., 2023), we use $\lambda = 0.5$, yielding an equal mixture of uniform and codec-adaptive frame allocation. Allocation details are provided in Appendix B.2.

3.3 Global Completion and Codec-Guided Token Aggregation

Global completion. Any capacity left after local I/P selection and Update-guided reference recovery is released to a video-global pool, where remaining eligible I/P positions compete against each other by unweighted max-min diversity. This allows unused local capacity to be reassigned to complementary evidence elsewhere in the video.

Codec-Guided Token Aggregation. Selecting tokens still leaves compatible evidence in non-selected tokens. We incorporate that evidence through feature-checked assignments, rather than discarding every non-selected token or treating every codec link as a valid semantic merge.

We first preserve all pre-recorded assignments whose destinations are retained tokens. Remaining unassigned I-frame positions and active P-frame positions are then matched to compatible retained tokens within the same frame. For P-frame tokens outside the candidate pool, assignments may instead be routed through codec parent links to a retained ancestor when the destination feature similarity satisfies the corresponding criterion. To ensure coverage, a final pass first attempts assignment to a retained ancestor and searches all retained tokens throughout the video. Each discarded token is assigned to at most one retained token, and all evaluations are performed using the original token features. Route-specific assignment criteria are detailed in Appendix B.3.

Following previous token-merging practice, a token w with accepted members $\mathcal{M}_w$ becomes

$$\overline{\mathbf{x}}_w = \frac{\mathbf{x}_w + \sum_{v\in\mathcal{M}_w} \omega_{vw}\mathbf{x}_v}{1 + \sum_{v\in\mathcal{M}_w} \omega_{vw}}. \tag{6}$$

Unmatched features are pruned and the resulting B aggregated tokens retain their representative positions and original sequence order and are passed to the frozen LLM.

3.4 SPATIOTEMPORAL ENCODER ADAPTATION

Spatiotemporal encoders, including Qwen3-VL and GLM-4.1V-Thinking, jointly encode multiple frames (Bai et al., 2025; GLM-V Team, 2025), whereas codec roles remain frame-specific. For temporal group size k, source position (t, i) maps to native position $\phi(t, i) = (\lfloor t/k \rfloor, i)$, using zero-based indices. Our evaluated Qwen3-VL adapter uses $k = 2$. Codec evidence, candidate provenance, and prediction chains are constructed on the source timeline before projection.

Each source frame retains its I/P role and Group of Pictures (GOP) membership, including within mixed IP and PI units. It contributes $1/k$ native-grid capacity to budget bookkeeping, while retention counts unique native positions. Source GOPs are processed chronologically: their I-frame roles establish reference anchors, followed by selection for their source P-frames. Predictive allocation uses individual source-frame packet costs, and local selection uses the corresponding source-specific candidates and codec scores.

All selection calls share the set of previously retained physical representatives, including those selected by earlier P-frame roles. These representatives seed subsequent selection and cannot consume another slot. Remaining capacity is handled by update-guided reference recovery and global completion in the native feature domain. Role-dependent eligibility and model integration are detailed in Appendix C.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Implementation. We re-encode the videos using an H.264 I/P prediction structure with no B-frames. The resulting codec sidecar records the frame type and reference index for each frame, together with motion vectors and residuals for predictive regions. We then map these block-level codec signals to the corresponding visual-token grid using the original image resolution and backbone-specific token layout. We evaluate LLaVA-OneVision-7B, LLaVA-Video-7B, InternVL3.5-8B, and Qwen3-VL-8B (Li et al., 2024; Zhang et al., 2024; Wang et al., 2025; Bai et al., 2025). LLaVA-OneVision, InternVL, and Qwen use 32 source frames on the three general benchmarks; LLaVA-Video uses 64 frames following the previous standard. All use eight frames on MotionBench. All models are frozen, and experiments run on a cluster equipped with 16 NVIDIA V100 GPUs. Complete implementation details and parameter configurations are provided in Appendix A.

Comparisons and benchmarks. We evaluate on VideoMME Fu et al. (2024), MVBench Li et al. (2023), MotionBench Hong et al. (2025), and LongVideoBench Wu et al. (2024), covering general video understanding, temporal and motion reasoning, and long-video understanding at 15%, 20%, and 25% visual-token retention ratios. We compare all methods under same protocol across VisionZip (Yang et al., 2025), FastVID (Shen et al., 2025), UniComp (Yuan et al., 2026), and FlashVID (Fan et al., 2026) methods.

4.2 RESULTS

Accuracy across backbones and budgets. Tables 1 and 2 show that CoGCo attains the highest four-benchmark average among the compared methods in all backbone-budget combinations. LLaVA-OneVision preserves over 100% of its uncompressed average performance at the 15% retention ratio. And LLaVA-Video preserves 98% of its average performance at a retention ratio as low as 15%. Even with InternVL, we preserve 96% of the uncompressed performance on average across the three retention settings, improving relative accuracy over FlashVID by 2.0 percentage points on average.

Motion and long-video performance. CoGCo matches or exceeds it on MotionBench and LongVideoBench, two benchmarks that are particularly challenging for token compression due to their reliance on fine-grained motion cues and long-range temporal context. These gains persist at the tightest budget: at 15%, LLaVA-OneVision improves by 0.8 percentage points on both MotionBench and LongVideoBench.

Compression-accuracy trade-off. Figure 3 plots the equal-weight four-benchmark average against target visual-token retention on LLaVA-OneVision. CoGCo achieves the highest average at every

Table 1: Comparison of SOTA training-free token compression methods across framewise Video LLMs.

Method	Ret.	VideoMME				MVBench	MotionBench	LongVideoBench	Avg.	Rel. Acc.
		Short	Medium	Long	Overall					
LLaVA-OneVision										
Vanilla	100%	69.9	56.7	48.9	58.5	58.3	53.2	56.6	56.7	100.0%
VisionZip	25%	68.8	57.3	48.2	58.1	57.8	51.5	56.4	56.0	98.8%
FastVID	25%	69.9	56.3	47.4	57.9	**58.1**	51.6	55.9	55.9	98.6%
UniComp	25%	70.9	57.1	48.4	58.8	56.8	52.4	57.6	56.4	99.6%
FlashVID	25%	71.2	57.0	49.3	59.2	58.0	53.1	56.3	56.7	100.0%
CoGCo (ours)	25%	**71.3**	**57.3**	**49.8**	**59.5**	58.0	**53.5**	**57.9**	**57.2**	**101.0%**
VisionZip	20%	68.6	**57.0**	48.3	58.0	57.6	51.5	55.4	55.6	98.2%
FastVID	20%	69.9	56.3	47.4	57.9	58.1	51.4	55.9	55.8	98.5%
UniComp	20%	**70.6**	56.9	48.2	58.6	56.6	51.6	57.7	56.1	99.1%
FlashVID	20%	70.1	55.4	48.9	58.2	58.2	52.7	57.4	56.6	100.0%
CoGCo (ours)	20%	70.0	56.8	**49.8**	**58.8**	**58.2**	**53.7**	**58.0**	**57.2**	**100.9%**
VisionZip	15%	63.8	54.6	48.3	55.6	53.5	50.0	54.1	53.3	94.1%
FastVID	15%	69.7	55.8	47.7	57.7	**58.2**	51.0	56.7	55.9	98.7%
UniComp	15%	69.6	55.5	48.5	57.9	55.9	51.3	57.4	55.6	98.2%
FlashVID	15%	69.6	56.0	48.9	58.2	57.9	52.4	57.3	56.5	99.6%
CoGCo (ours)	15%	**69.8**	**56.9**	**49.9**	**58.9**	57.7	**53.2**	**58.1**	**57.0**	**100.6%**
LLaVA-Video										
Vanilla	100%	77.0	62.1	53.3	64.2	61.9	53.6	59.5	59.8	100.0%
VisionZip	25%	74.0	60.3	52.9	62.4	60.0	50.1	58.3	57.7	96.5%
FastVID	25%	74.7	60.1	53.6	62.8	**60.5**	52.1	58.2	58.4	97.7%
UniComp	25%	70.3	58.4	52.3	60.4	59.0	52.2	**59.4**	57.8	96.6%
FlashVID	25%	74.2	**61.4**	51.6	62.4	60.2	52.8	58.4	58.5	97.7%
CoGCo (ours)	25%	**74.7**	60.3	**54.1**	**63.1**	60.2	**52.9**	59.3	**58.9**	**98.5%**
VisionZip	20%	72.3	59.6	**53.3**	61.7	59.8	49.4	58.0	57.2	95.7%
FastVID	20%	74.6	**60.8**	52.3	**62.6**	**60.2**	51.8	57.1	57.9	96.9%
UniComp	20%	69.9	58.3	52.4	60.2	58.2	51.6	59.3	57.3	95.9%
FlashVID	20%	74.1	60.0	52.3	62.2	59.8	52.8	58.5	58.3	97.5%
CoGCo (ours)	20%	**74.6**	59.4	53.1	62.3	60.0	**53.0**	**59.3**	**58.7**	**98.1%**
VisionZip	15%	72.9	58.1	51.9	61.0	59.6	48.6	57.2	56.6	94.6%
FastVID	15%	73.4	58.1	51.8	61.1	60.3	51.4	57.7	57.6	96.4%
UniComp	15%	70.2	57.9	51.4	59.9	56.7	51.8	58.7	56.8	94.9%
FlashVID	15%	73.8	59.6	52.1	61.8	**60.4**	52.5	58.3	58.3	97.4%
CoGCo (ours)	15%	**74.1**	**60.0**	**52.7**	**62.3**	59.5	**53.0**	**58.9**	**58.4**	**97.7%**
InternVL-3.5										
Vanilla	100%	76.6	63.1	53.7	64.4	72.9	56.9	60.0	63.6	100.0%
UniComp	25%	73.9	61.4	52.2	62.5	66.8	55.4	57.7	60.6	95.4%
FlashVID	25%	72.0	58.8	50.8	60.5	66.7	55.5	59.2	60.5	95.2%
CoGCo (ours)	25%	**74.9**	**61.6**	**53.2**	**63.2**	**68.6**	**55.7**	**59.8**	**61.8**	**97.3%**
UniComp	20%	73.1	59.7	**52.7**	61.8	65.3	55.1	58.2	60.1	94.6%
FlashVID	20%	70.8	58.3	51.6	60.2	65.1	55.0	57.6	59.5	93.6%
CoGCo (ours)	20%	**74.6**	**59.7**	51.7	**62.0**	**67.8**	**55.5**	**59.2**	**61.1**	**96.2%**
UniComp	15%	70.4	58.2	52.2	60.3	63.4	54.9	**57.8**	59.1	93.0%
FlashVID	15%	70.4	58.8	51.2	60.1	64.7	55.3	57.6	59.4	93.5%
CoGCo (ours)	15%	**72.7**	**61.1**	**52.4**	**62.1**	**65.8**	**55.6**	57.7	**60.3**	**94.9%**

evaluated setting. Reducing target retention from 25% to 15% lowers average accuracy by only 0.2 percentage points, from 57.2% to 57.0%. At 15%, it still exceeds the full-token average of 56.7% and FlashVID at 25%, achieving higher accuracy with a 40% smaller target token budget than the latter, with comparable latency shown in Figure 4.

4.3 Ablation Studies

We ablate two aspects of CoGCo: codec-guided update coordination and role-specific handling of high attention features. Table 3 reports the average accuracy over VideoMME, MVBench, MotionBench, and LongVideoBench on LLaVA-Video at three retention ratios.

Codec-guided update coordination. We first examine how codec information contributes to P-frame update selection and cross-frame coordination. Replacing within-frame diversity within the

Table 2: Comparison of state-of-the-art training-free visual token compression methods on Qwen3-VL backbone.

Method	Ret.	VideoMME	MVBench	MotionBench	LongVideoBench	Avg.
Vanilla	100%	61.3	66.9	55.7	58.8	60.7
UniComp	25%	57.3	63.5	52.2	53.0	56.5
FlashVID	25%	58.4	64.5	54.1	**57.4**	58.6
CoGCo	25%	**59.4**	**65.3**	**54.2**	56.9	**59.0**
UniComp	20%	56.4	62.6	51.6	52.4	55.8
FlashVID	20%	58.0	64.6	53.7	55.8	58.0
CoGCo	20%	**58.4**	**64.6**	**53.7**	**56.7**	**58.4**
UniComp	15%	54.9	59.2	50.1	50.6	53.7
FlashVID	15%	56.4	**63.8**	52.6	54.0	56.7
CoGCo	15%	**58.1**	63.4	**53.3**	**56.6**	**57.9**

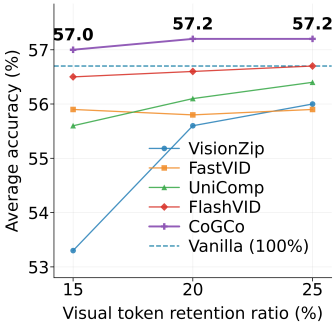


Figure 3: Accuracy-retention trade-off on LLaVA-OneVision.

Table 3: Ablations on LLaVA-Video. Each entry averages VideoMME, MVBench, MotionBench, and LongVideoBench.

Variant	25% Ret.	20% Ret.	15% Ret.	Average
Codec-guided coordination				
Attn-based P candidates selection	58.7	58.2	58.1	58.3
Without update-guided P candidates	58.5	57.8	57.5	57.9
Uniform P-frame allocation	58.5	58.5	58.0	58.3
No update-guided reference recovery	58.4	58.3	58.2	58.3
Role-aware attention handling				
No attention-based exclusion	58.6	58.6	57.9	58.4
Exclude in both I and P frames	58.7	58.7	57.9	58.4
Random P exclusion, equal count	58.8	58.6	58.2	58.5
Full CoGCo	58.9	58.7	58.4	58.7

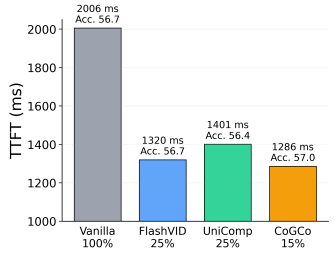


Figure 4: Time-To-First-Token (TTFT) comparison at comparable accuracy on LLaVA-OneVision.

codec-defined update pool with attention-based selection reduces the average from 58.7 to 58.3. Removing the update candidate pool altogether and applying diversity directly to P-frame selection causes the largest drop, reaching 57.5 at 15% retention compared with 58.4 for the full method. Packet-aware allocation and update-guided reference recovery are also beneficial: replacing the former with uniform P-frame allocation or removing the latter reduces the overall average to 58.3. Together, these results show that codec cues are useful not only for identifying candidate updates, but also for distributing update capacity and refining reference coverage.

Role-specific high attention handling. We next examine whether high attention tokens should be treated differently according to their reference or update role. Removing P-frame attention exclusion reduces the overall average from 58.7 to 58.4, while applying the same exclusion to both I- and P-frames also yields 58.4. Randomly excluding the same number of P-frame positions still performs worse than the full method. These results indicate that the gain does not arise simply from removing P-frame tokens: preserving high attention features as potential reference context while restricting their independent P-frame selection provides a better use of the limited update budget.

5 Conclusion

We presented CoGCo, a training-free video token compression framework that uses video codecs as a natural prior for preserving semantic context and temporal evidence. Attention- and diversity-aware selection establishes I-frame semantic anchors, while subsequent predictive evidence helps recover initially overlooked reference content. P-frame selection complements these anchors with diverse updates identified through motion and residual cues, restricting independent selection from high-attention fraction to reserve capacity for fine-grained temporal evidence. Codec-guided allocation and aggregation coordinate these complementary representations under a limited token budget without modifying model weights. Across four Video LLM backbones and four benchmarks, CoGCo improves aggregate accuracy over the compared training-free compression baselines at the evaluated retention settings. On LLaVA-OneVision, its four-benchmark average exceeds the uncompressed vanilla version at only 15% retention. Limitation and discussion are provided in Appendix E.

AI USE DISCLOSURE

Generative AI tools were used to assist with language editing and polishing, including improving grammar, clarity, readability, and presentation. Generative AI tools were also used to create and edit software code. The tools were not used to develop the research methodology, design experiments, generate or interpret experimental results, or formulate the scientific conclusions of this work. All AI-assisted edits were reviewed and verified by the authors. The authors take full responsibility for the final content of the paper.

REFERENCES

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://arxiv.org/abs/2503.02175.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

Barak Battash, Haim Barad, Hanlin Tang, and Amit Bleiweiss. Mimic the raw domain: Accelerating action recognition in the compressed domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 684–685, 2020. URL https://openaccess.thecvf.com/content_CVPRW_2020/html/w40/Battash_Mimic_the_Raw_Domain_Accelerating_Action_Recognition_in_the_Compressed_CVPRW_2020_paper.html.

Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J. Sullivan, and Jens-Rainer Ohm. Overview of the Versatile Video Coding (VVC) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3736–3764, 2021. doi: 10.1109/TCSVT.2021.3101953. URL https://doi.org/10.1109/TCSVT.2021.3101953.

Jiawei Chen and Chiu Man Ho. MM-ViT: Multi-modal video transformer for compressed video action recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1910–1921, 2022. URL https://openaccess.thecvf.com/content/WACV2022/html/Chen_MM-ViT_Multi-Modal_Video_Transformer_for_Compressed_Video_Action_Recognition_WACV_2022_paper.html.

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. *arXiv preprint arXiv:2403.06764*, 2024. URL https://arxiv.org/abs/2403.06764.

Pengxu Chen, Yao Zhu, Guangming Zhu, Jun Sheng, Jincai Huang, Xiangyang Ji, and Liang Zhang. Look clearly before answering: Mitigating hallucinations in lvlms via saliency-driven perceptual realignment. In *Proceedings of the ACM International Conference on Multimedia*, 2026a. doi: 10.1145/3767308.3836111.

Xueyi Chen, Keda Tao, Kele Shao, and Huan Wang. StreamingTOM: Streaming token compression for efficient video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026b. URL https://arxiv.org/abs/2510.18269.

Zhipeng Fan, Jun Liu, and Yao Wang. Motion adaptive pose estimation from compressed videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11719–11728, 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Fan_Motion_Adaptive_Pose_Estimation_From_Compressed_Videos_ICCV_2021_paper.html.

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. *arXiv preprint arXiv:2602.08024*, 2026. URL https://arxiv.org/abs/2602.08024.

Yuan Feng, Junlin Lv, Yukun Cao, Xike Xie, and S. Kevin Zhou. Ada-kv: Optimizing kv cache eviction by adaptive budget allocation for efficient llm inference. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-3773.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, et al. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. *arXiv preprint arXiv:2405.21075*, 2024. URL https://arxiv.org/abs/2405.21075.

Tianyu Fu, Tengxuan Liu, Qinghao Han, Guohao Dai, Shengen Yan, Huazhong Yang, Xuefei Ning, and Yu Wang. FrameFusion: Combining similarity and importance for video token reduction on large vision language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025. URL https://arxiv.org/abs/2501.01986.

GLM-V Team. Glm-4.5v and glm-4.1v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025. URL https://arxiv.org/abs/2507.01006.

Wenyi Hong, Yean Cheng, Zhuoyi Yang, et al. MotionBench: Benchmarking and improving fine-grained video motion understanding for vision language models. *arXiv preprint arXiv:2501.02955*, 2025. URL https://arxiv.org/abs/2501.02955.

Haowen Hou, Zhen Huang, Zheming Liang, Qingyi Si, Chenglin Li, Shuai Dong, Kele Shao, Ruilin Li, Dianyi Wang, Nan Duan, and Jiaqi Wang. AdaCodec: A predictive visual code for video MLLMs. *arXiv preprint arXiv:2606.02569*, 2026. URL https://arxiv.org/abs/2606.02569.

Hai-Miao Hu, Bo Li, Weiyao Lin, Wei Li, and Ming-Ting Sun. Region-based rate control for h.264/avc for low bit-rate applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(11):1564–1576, 2012. doi: 10.1109/TCSVT.2012.2199398.

Xiaohu Huang, Hao Zhou, and Kai Han. PruneVid: Visual token pruning for efficient video large language models. *arXiv preprint arXiv:2412.16117*, 2024. URL https://arxiv.org/abs/2412.16117.

Jeongseok Hyun, Sukjun Hwang, Su Ho Han, Taeoh Kim, Inwoong Lee, Dongyoon Wee, Joon-Young Lee, Seon Joo Kim, and Minho Shim. Multi-granular spatio-temporal token merging for training-free acceleration of video LLMs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23990–24000, 2025. URL https://arxiv.org/abs/2507.07990.

International Telecommunication Union. ITU, ISO and IEC receive another primetime emmy for video compression, 2020. ITU News.

Shaobo Ju, Baiyang Song, Tao Chen, Jiapeng Zhang, Qiong Wu, Chao Chang, Huaixi Wang, Yiyi Zhou, and Rongrong Ji. ForestPrune: High-ratio visual token compression for video multimodal large language models via spatial-temporal forest modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Findings*, pp. 8326–8336, 2026. URL https://arxiv.org/abs/2603.22911.

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. *arXiv preprint arXiv:2408.03326*, 2024. URL https://arxiv.org/abs/2408.03326.

Congcong Li, Xinyao Wang, Longyin Wen, Dexiang Hong, Tiejian Luo, and Libo Zhang. End-to-end compressed video representation learning for generic event boundary detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13967–13976, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Li_End-to-End_Compressed_Video_Representation_Learning_for_Generic_Event_Boundary_Detection_CVPR_2022_paper.html.

Kunchang Li, Yali Wang, Yinan He, et al. MVBench: A comprehensive multi-modal video understanding benchmark. *arXiv preprint arXiv:2311.17005*, 2023. URL https://arxiv.org/abs/2311.17005.

Andrew Lu, Wentinn Liao, Liuhui Wang, Huzheng Yang, and Jianbo Shi. Artifacts and attention sinks: Structured approximations for efficient vision transformers. *arXiv preprint arXiv:2507.16018*, 2025. URL https://arxiv.org/abs/2507.16018.

Jiayun Luo, Wan-Cyuan Fan, Lyuyang Wang, Xiangteng He, Tanzila Rahman, Purang Abolmaesumi, and Leonid Sigal. To sink or not to sink: Visual information pathways in large vision-language models. *arXiv preprint arXiv:2510.08510*, 2025. URL https://arxiv.org/abs/2510.08510.

Sayan Deb Sarkar, Rémi Pautrat, Ondrej Miksik, Marc Pollefeys, Iro Armeni, Mahdi Rad, and Mihai Dusmanu. Cope-videolm: Leveraging codec primitives for efficient video language modeling. *arXiv preprint arXiv:2602.13191*, 2026. URL https://arxiv.org/abs/2602.13191.

Kele Shao, Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. HoliTom: Holistic token merging for fast video large language models. *arXiv preprint arXiv:2505.21334*, 2025. URL https://arxiv.org/abs/2505.21334.

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. *arXiv preprint arXiv:2503.11187*, 2025. URL https://arxiv.org/abs/2503.11187.

Zheng Shou, Xudong Lin, Yannis Kalantidis, Laura Sevilla-Lara, Marcus Rohrbach, Shih-Fu Chang, and Zhicheng Yan. DMC-Net: Generating discriminative motion cues for fast compressed video action recognition. *arXiv preprint arXiv:1901.03460*, 2019. URL https://arxiv.org/abs/1901.03460.

Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the High Efficiency Video Coding (HEVC) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668, 2012. doi: 10.1109/TCSVT.2012.2221191. URL https://doi.org/10.1109/TCSVT.2012.2221191.

Feilong Tang, Xiang An, Yunyao Yan, Yin Xie, Bin Qin, Kaicheng Yang, Yifei Shen, Yuanhan Zhang, Chunyuan Li, Shikun Feng, Changrui Chen, Huajie Tan, Ming Hu, Manyuan Zhang, Bo Li, Ziyong Feng, Ziwei Liu, Zongyuan Ge, and Jiankang Deng. OneVision-Encoder: Codec-aligned sparsity as a foundational principle for multimodal intelligence. *arXiv preprint arXiv:2602.08683*, 2026. URL https://arxiv.org/abs/2602.08683.

Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. DyCoke: Dynamic compression of tokens for fast video large language models. *arXiv preprint arXiv:2411.15024*, 2024. URL https://arxiv.org/abs/2411.15024.

Hesong Wang, Xin Jin, Lu Lu, Chenhaowen Li, Jian Chen, Qiang Liu, and Huan Wang. EarlyTom: Early token compression completes fast video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 40559–40568, 2026.

Weiyun Wang, Zhangwei Gao, Lixin Gu, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025. URL https://arxiv.org/abs/2508.18265.

Thomas Wiegand, Gary J. Sullivan, Gisle Bjøntegaard, and Ajay Luthra. Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7): 560–576, 2003. doi: 10.1109/TCSVT.2003.815165. URL https://doi.org/10.1109/TCSVT.2003.815165.

Chao-Yuan Wu, Manzil Zaheer, Hexiang Hu, R. Manmatha, Alexander J. Smola, and Philipp Krähenbühl. Compressed video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. URL https://arxiv.org/abs/1712.00636.

Chenxi Wu, Min Zhu, Qinyang Tan, Yadhu Kartha, and Lu Lu. A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 403:115671, 2023. doi: 10.1016/j.cma.2022.115671.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. LongVideoBench: A benchmark for long-context interleaved video-language understanding. *arXiv preprint arXiv:2407.15754*, 2024. URL https://arxiv.org/abs/2407.15754.

Kai Xu and Angela Yao. Accelerating video object segmentation with compressed video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1342–1351, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Xu_Accelerating_Video_Object_Segmentation_With_Compressed_Video_CVPR_2022_paper.html.

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. doi: 10.1109/CVPR52734.2025.01843. URL https://arxiv.org/abs/2412.04467.

Senqiao Yang, Kaichen Zhang, Zhaoyang Jia, et al. Mage-VL: An efficient codec-native streaming multimodal foundation model. *arXiv preprint arXiv:2607.24904*, 2026. URL https://arxiv.org/abs/2607.24904.

Daichi Yashima, Shuhei Kurita, Yusuke Oda, and Komei Sugiura. ReMoRa: Multimodal large language model based on refined motion representation for long-video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 31845–31855, 2026. URL https://arxiv.org/abs/2602.16412.

Chao Yuan, Shimin Chen, Minliang Lin, Limeng Qiao, Guanglu Wan, and Lin Ma. UniComp: Rethinking video compression through informational uniqueness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2512.03575. arXiv:2512.03575.

Bowen Zhang, Limin Wang, Zhe Wang, Yu Qiao, and Hanli Wang. Real-time action recognition with enhanced motion vector CNNs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2718–2726, 2016. URL https://openaccess.thecvf.com/content_cvpr_2016/html/Zhang_Real-Time_Action_Recognition_CVPR_2016_paper.html.

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

Zijia Zhao, Yuqi Huo, Tongtian Yue, Longteng Guo, Haoyu Lu, Bingning Wang, Weipeng Chen, and Jing Liu. Efficient motion-aware video MLLM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24159–24168, 2025. URL https://arxiv.org/abs/2503.13016.

A Implementation and Evaluation Protocol

A.1 Data Preparation and Evaluation Setup

Frame sampling and codec preparation. The preparation pipeline fixes the sampled frame sequence and spatial transformation before extracting codec metadata. Visual inputs and codec signals therefore refer to the same prepared sequence. LLaVA-OneVision, InternVL3.5, and Qwen3-VL use 32 source frames on VideoMME, MVBench, and LongVideoBench, while LLaVA-Video uses 64 frames. All four backbones use eight frames on MotionBench.

The prepared videos use an H.264 I/P prediction structure with a single reference and no B-frames. Scene-cut I-frames are allowed, so the sampled sequence need not consist of one I-frame followed by a fixed number of P-frames. The codec sidecar records frame types, reference indices, motion vectors, patch-level residual cues, and frame packet sizes.

Spatial and temporal alignment. Patch centers and motion displacements are interpreted in the recorded image coordinate system. The displaced position is quantized by floor and clipped at the image boundary when constructing codec parent links. Invalid reference indices use the guarded previous-frame fallback. I-frame motion and residual maps are zeroed, and negative residual values are clipped.

The evaluated framewise implementation uses the cached, nonnegative, patch-pooled cue from the decoder's residual buffer. Qwen3-VL instead requires an explicit source-frame-to-temporal-unit mapping, detailed in Appendix C.

Packet metadata and preprocessing cost. Frame packet sizes are read from companion metadata associated with the prepared video. The loader checks frame counts, the I/P timeline, and the association between the packet metadata and codec sidecar. Packet sizes must be finite and positive. They represent coding costs, rather than semantic annotations or visual-token counts.

Codec metadata is prepared in advance and reused across questions about the same video. The outer selector does not re-encode the video or invoke packet probing for each question. Sidecar preparation is a one-time per-video preprocessing cost and is excluded from the model-inference TTFT reported in Figure 4; the timing boundary is specified in Appendix A.5.

A.2 Backbone Interfaces and Token Accounting

Visual interfaces. Table 4 summarizes the visual grids and inner-pruning schedules. LLaVA-OneVision uses bilinear spatial pooling, whereas LLaVA-Video uses average pooling. InternVL3.5 uses one tile per frame, pixel-shuffle/projector features, and its model-specific conversational wrapper. These three framewise backbones share the same outer selection and aggregation algorithm.

Qwen3-VL uses native temporal patches spanning two source frames. Its source-frame bookkeeping, physical-token mapping, and adapter-specific selection rules are described separately in Appendix C. Newline, separator, timestamp, and other formatting positions are handled by the corresponding model interface rather than treated as independent codec patch tokens.

Compression configuration. We adopt FlashVID's hybrid compression convention, combining expanded pre-LLM token retention with a fixed inner-LLM pruning step (Fan et al., 2026). The reported retention setting r specifies the nominal compression budget rather than a uniform token fraction at every LLM layer. The outer expansion factor and backbone-specific inner-pruning schedules are fixed across datasets and retention settings; they are not tuned as part of CoGCo. For a nominal retention setting $r \in \{0.15, 0.20, 0.25\}$, the outer token budget is

$$\rho_v = \min(1, er), \qquad B = \max\{1, \min(V, \lfloor V\rho_v \rfloor)\}, \tag{7}$$

where V is the number of physical visual patch tokens and $e = 1.25$ is the outer-budget scaling adopted from FlashVID. For framewise backbones, $V = TN$; for Qwen3-VL's two-frame temporal grouping, $V = (T/2)N$.

The first K language-model layers process the B retained patch tokens. Before zero-based layer K, text-to-visual attention is used to retain $\lceil \eta B \rceil$ patch tokens for the remaining layers. We use

Table 4: Visual interfaces and inner-pruning schedules. The Frames column applies to VideoMME, MVBench, and LongVideoBench; Motion denotes MotionBench. L is the number of language-model layers. Pruning occurs before zero-based layer K, after the first K layers have processed the outer representation. η is the inner visual-token retention fraction.

Backbone	Frames	Motion	Grid/unit	Units/frame	L	K	η
LLaVA-OneVision-7B	32	8	14×14	1	28	20	0.30
LLaVA-Video-7B	64	8	13×13	1	28	20	0.30
InternVL3.5-8B	32	8	16×16	1	36	20	0.30
Qwen3-VL-8B	32	8	12×12	$1/2$	36	28	0.10

$(K, \eta) = (20, 0.30)$ for the framewise backbones and $(28, 0.10)$ for Qwen3-VL, as summarized in Table 4. The scaling factor and inner-pruning settings are fixed across datasets and nominal retention levels. Formatting positions follow the corresponding model interface. The outer codec-guided selector is question-agnostic.

A.3 Hyperparameters and Configuration Selection

Shared settings and sensitivity analysis. The reported configuration uses three shared settings: the packet-allocation mixing weight $\lambda = 0.50$, the P-role attention-exclusion fraction $\delta = 0.04$, and the update-candidate quantile $q = 0.70$. These values are fixed across datasets and nominal retention ratios. Their sensitivities on LLaVA-OneVision and LLaVA-Video are examined in Tables 5, 6, and 7, respectively.

For each sensitivity sweep, only the parameter under study is varied. The four backbone–dataset-specific parameters $\alpha_{\min}$, $\alpha_{\max}$, τ_{sel}, and $\tau_{\text{I-rem}}$ remain fixed at the corresponding full-method values in Table 8. The λ sweep keeps $\delta = 0.04$, and the δ sweep keeps $\lambda = 0.50$. All other hyperparameters, nominal token budgets, inner-pruning schedules, and evaluation settings remain unchanged; no parameter configuration is reselected for an individual sweep value.

Packet-allocation mixing weight. In Eq. (5), λ interpolates between uniform frame coverage and allocation according to codec packet cost. Setting $\lambda = 0$ gives uniform P-frame allocation, whereas larger values place progressively more weight on relative coding demand. We use $\lambda = 0.50$ as an equal mixture of these two signals. Table 5 evaluates neighboring values on both LLaVA-OneVision and LLaVA-Video. The balanced setting gives the highest mean accuracy across the three retention ratios for both backbones, although individual ratios can tie with or slightly favor a neighboring value. We therefore fix $\lambda = 0.50$ for all framewise experiments rather than tuning it separately for each dataset or retention ratio.

P-frame attention-exclusion fraction. Prior work has identified visual attention sinks that receive disproportionately large attention. Reported examples include approximately sevenfold attention to propagated ViT sinks during LLM decoding (Luo et al., 2025) and a roughly tenfold contrast between sink and discriminative tokens in a sink-dominated LLaVA attention head (Chen et al., 2026a). Related analyses identify sinks directly within visual encoders through unusually high incoming or CLS-to-patch attention (Lu et al., 2025). Motivated by these observations, we examine the high-attention tail in the visual-encoder scores used by our method. We estimate its prevalence and evaluate neighboring exclusion fractions to select a shared P-role exclusion setting.

To estimate the prevalence of high attention positions, we perform a separate diagnostic that is not used during inference. For each frame or native temporal grid, we min–max normalize the visual-encoder attention scores to obtain $a_{u,i}^{\text{diag}} \in [0, 1]$ and assign integer levels

$$\ell_{u,i} = \left\lfloor 10 a_{u,i}^{\text{diag}} \right\rfloor \in \{0, 1, \ldots, 10\}. \tag{8}$$

We count positions with $\ell_{u,i} \geq 3$, equivalently $a_{u,i}^{\text{diag}} \geq 0.3$, as high attention positions for this diagnostic. These integer levels describe normalized attention magnitude rather than a universal classification of semantic sink tokens.

Table 5: Sensitivity to the packet mixing weight λ. Each entry averages VideoMME, MVBench, MotionBench, and LongVideoBench at the corresponding nominal retention ratio.

Backbone	λ	25% Ret.	20% Ret.	15% Ret.
LLaVA-OneVision	0.3	56.9	56.9	56.7
	0.4	57.0	57.0	56.8
	0.5	**57.2**	**57.2**	**57.0**
	0.6	56.9	56.8	56.9
	0.7	56.8	56.7	56.5
LLaVA-Video	0.3	58.6	58.7	58.2
	0.4	58.5	58.6	58.1
	0.5	**58.9**	**58.7**	58.4
	0.6	58.7	58.6	**58.5**
	0.7	58.7	58.7	58.3

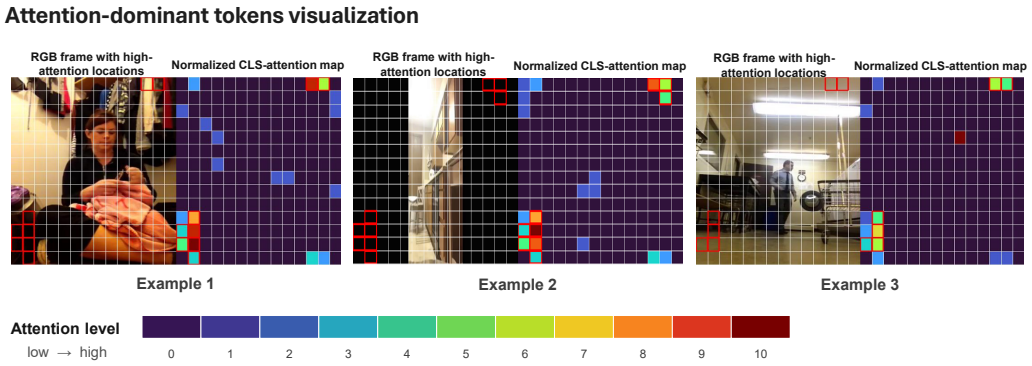


Figure 5: Examples of high attention visual positions. For each example, the left panel shows the RGB frame with high attention positions outlined in red, and the right panel shows the corresponding min–max normalized CLS-attention map. Colors denote quantized attention levels from 0 (lowest) to 10 (highest), and red outlines indicate positions with $\ell_{u,i} \geq 3$.

Figure 5 illustrates that the attention distribution contains a sparse but pronounced high-attention tail, motivating a quantitative estimate of its prevalence.

We perform this diagnostic on 100 videos for each backbone. For LLaVA-OneVision, we analyze 3,200 frames and observe on average 9.8 such positions among 196 visual tokens per frame, corresponding to approximately 5.0%. For LLaVA-Video, over 6,400 frames, the corresponding average is 5.5 among 169 tokens, or approximately 3.3%. For Qwen3-VL, 3,200 source frames map to 1,600 native temporal feature grids under its temporal patch size $k = 2$; across these 12×12 grids, we observe on average 5.8 high attention positions among 144 physical positions, corresponding to 4.1%.

Pooling all three diagnostics according to the number of visual positions gives

$$\frac{3200 \times 9.8 + 6400 \times 5.5 + 1600 \times 5.8}{3200 \times 196 + 6400 \times 169 + 1600 \times 144} \simeq 3.91\%. \tag{9}$$

The consistency across different visual encoder architectures, together with the pooled prevalence near 4%, motivates a shared exclusion fraction of $\delta = 0.04$.

The operational rule does not depend on the diagnostic normalization or an absolute attention threshold. Instead, for each P-frame we exclude the $\lfloor \delta N \rfloor$ highest-attention positions from independent root selection. This rank-based rule avoids dependence on the absolute attention scale of a particular backbone. I-frame positions remain unrestricted, and excluded P-frame positions remain eligible to contribute through compatible feature aggregation.

Table 6 further compares neighboring values of δ. The 4% setting is best or tied for best in all six backbone–retention combinations, supporting the prevalence-based choice while showing degradation when either fewer or more high attention P-frame positions are excluded. We therefore use $\delta = 0.04$ throughout the framewise experiments.

Table 6: Sensitivity to the P-frame high-attention exclusion fraction δ. Each entry averages VideoMME, MVBench, MotionBench, and LongVideoBench.

Backbone	Ret.	$\delta = 3\%$	$\delta = 4\%$	$\delta = 5\%$
LLaVA-OneVision	25%	56.9	**57.2**	56.8
	20%	56.9	**57.2**	56.9
	15%	56.4	**57.0**	56.5
LLaVA-Video	25%	58.8	**58.9**	58.7
	20%	**58.7**	**58.7**	**58.7**
	15%	58.2	**58.4**	58.3

Table 7: Sensitivity to the update-candidate quantile q. Each entry reports the equal-weight average accuracy (%) over VideoMME, MVBench, MotionBench, and LongVideoBench at the indicated nominal retention ratio. The best results in each row are bolded, including ties at the reported precision.

Backbone	Ret.	$q = 0.60$	$q = 0.70$	$q = 0.80$
LLaVA-OneVision	25%	56.8	**57.2**	56.7
	20%	56.8	**57.2**	57.1
	15%	56.3	**57.0**	56.2
LLaVA-Video	25%	**58.9**	**58.9**	58.7
	20%	**58.7**	**58.7**	58.5
	15%	58.2	**58.4**	58.0

Update-candidate quantile. The quantile level q controls the initial exposure of codec-guided P-frame candidates:

$$e_{t,i} \geq \max\{Q_q(\mathbf{e}_t),\, 0.5\}\,, \tag{10}$$

where Q_q denotes the q-quantile of the frame's raw update evidence. Lower values relax this initial gate, whereas higher values restrict it to positions with stronger codec evidence. The top-eight union, zero-evidence handling, and GOP-level candidate supplementation are described in Appendix B.1.

Table 7 compares $q = 0.60$, 0.70, and 0.80 on LLaVA-OneVision and LLaVA-Video. The setting $q = 0.70$ achieves the highest or tied-highest four-benchmark average at every evaluated backbone–retention combination. Although $q = 0.60$ matches its performance on LLaVA-Video at 25% and 20% retention, $q = 0.70$ performs best on both backbones at 15% retention. These results support retaining $q = 0.70$ as a shared setting rather than selecting a separate quantile for each dataset or retention ratio.

Other fixed settings. The evaluated framewise configuration enables the final ancestor/global rescue pass with cosine threshold 0.85. This pass may recover previously unassigned members but does not guarantee that every discarded token is merged. Table 9 summarizes the aggregation routes, their precedence, and acceptance tests; the corresponding member weights are specified in Appendix B.3.

Selected numerical configurations. Following prior training-free token-compression studies, we empirically configure inference-time hyperparameters while keeping all pretrained model weights frozen (Fan et al., 2026; Shao et al., 2025). Table 8 reports the four searched parameters, selected through full-benchmark evaluations. Each backbone–dataset pair uses one configuration shared across all three nominal retention settings, without budget-specific reselection. The shared settings λ, δ, and q remain fixed as specified above. For the compared baselines, we use the author-recommended method-specific hyperparameters. For the three framewise backbones, the four searched parameters operate directly on per-frame token grids; for Qwen3-VL, the same parameter definitions are applied inside the source-frame adaptation described in Appendix C.

The allocation parameters $\alpha_{\min}$ and $\alpha_{\max}$ control the provisional P budget and candidate-exposure floor, respectively. They do not impose a final hard I:P ratio after global completion. The threshold τ_{sel} controls selection-time merging during I-anchor selection and global completion. The threshold $\tau_{\text{I-rem}}$ controls same-frame merging associated with I-addition capping and cleanup. Neither

Table 8: Selected configurations for the four backbones. Each backbone–dataset configuration is shared across the 15%, 20%, and 25% retention settings. The packet-allocation coefficient is fixed at $\lambda = 0.50$.

Backbone	Dataset	$\alpha_{\min}$	$\alpha_{\max}$	τ_{sel}	$\tau_{\text{I-rem}}$
LLaVA-OneVision	VideoMME	1.25	2.25	0.875	0.825
	MVBench	1.75	2.75	0.925	0.825
	MotionBench	1.75	2.75	0.925	0.825
	LongVideoBench	1.25	2.00	0.850	0.875
LLaVA-Video	VideoMME	1.00	2.50	0.925	0.925
	MVBench	1.25	2.00	0.875	0.875
	MotionBench	1.75	3.25	0.925	0.875
	LongVideoBench	1.50	2.25	0.900	0.850
InternVL3.5	VideoMME	1.75	2.50	0.925	0.875
	MVBench	1.50	2.50	0.875	0.850
	MotionBench	2.25	3.00	0.925	0.825
	LongVideoBench	1.50	2.50	0.900	0.850
Qwen3-VL	VideoMME	2.25	3.25	0.900	0.850
	MVBench	2.25	3.25	0.900	0.850
	MotionBench	2.25	3.25	0.900	0.850
	LongVideoBench	2.00	3.00	0.900	0.850

threshold is used by the local pure-diversity P-root selector, which performs no selection-time merging.

A.4 Benchmark Metrics and Numerical Reporting

Benchmark-specific metrics. Evaluation uses the benchmark-specific prompts and answer parsers. MVBench follows task-level aggregation. VideoMME reports Short, Medium, Long, and Overall accuracy. LongVideoBench uses its benchmark evaluator. MotionBench uses the established valid non-NA population; invalid or NA annotations are not treated as ordinary incorrect answers. Inference or parsing failures on otherwise eligible questions are not removed merely because the model fails to answer them.

Average and relative accuracy. Let A_{VM}, A_{MV}, A_{MB}, and A_{LVB} denote VideoMME Overall, MVBench, MotionBench, and LongVideoBench accuracy, respectively. For each backbone and retention setting, the four-benchmark macro average is

$$A_{\text{avg}} = \frac{A_{\text{VM}} + A_{\text{MV}} + A_{\text{MB}} + A_{\text{LVB}}}{4}. \tag{11}$$

Relative accuracy is defined with respect to the Vanilla average of the same backbone:

$$A_{\text{rel}} = 100\,\frac{A_{\text{avg}}}{A_{\text{avg}}^{\text{Vanilla}}}. \tag{12}$$

A.5 Inference Latency Measurement

We measure time-to-first-token (TTFT) on LLaVA-OneVision-7B using a single NVIDIA V100 GPU with batch size one. The timing set consists of 96 fixed VideoMME videos, with 32 videos from each of the Short, Medium, and Long categories. Each video uses the 32-frame input setting described in Appendix A.1. We average three repeated measurements for each video and report the mean over the 96 videos. All compared methods use the same timing inputs and measurement boundary.

TTFT is measured as wall-clock time from submitting the prepared inputs to the model, before vision encoding, until the first generated token is returned to the CPU. The timed interval includes visual encoding and projection, online token compression where applicable, LLM prefilling, and first-token generation and return. Video decoding, frame preparation, and offline codec-sidecar generation

are outside this interval. The reported metric therefore characterizes model-inference latency with prepared inputs, rather than cold-start latency from a raw video file.

B Detailed Selection and Aggregation

B.1 Score Calibration and Candidate Construction

Visual attention extraction. We use the same backbone-specific visual-attention extraction procedure as FlashVID (Fan et al., 2026). For encoders with an explicit CLS token, the score is derived from CLS-to-patch attention. For encoders without a CLS token, a CLS-equivalent score is obtained by averaging the attention received by each visual token within its frame or native temporal grid. We refer to both cases as visual-encoder attention scores.

Ordinal calibration. After replacing nonfinite entries by zero, let $\mathrm{rank}^{\downarrow}(v_i) \in \{1, \dots, N\}$ be the stable, descending, one-based ordinal rank. The implementation uses

$$\mathcal{I}(v_i) = -\log \frac{\mathrm{rank}^{\downarrow}(v_i) + 1}{N + 1}. \tag{13}$$

A vector with range below $\varepsilon = 10^{-6}$ receives zero information scores. In a nonconstant vector, equal values are ordered by their original indices; this ordinal calibration does not assign identical scores to all tied values.

I-frame appearance anchors. Let $\mathcal{N}_\varepsilon(\mathbf{a})$ denote within-frame min–max normalization with ε added to the denominator, returning zero when the range is below ε. Attention is sanitized and clipped to nonnegative values before calibration. Define

$$c_{t,i} = \mathcal{I}(a_{t,i}) \left(10^6 \max\{\mathcal{N}_\varepsilon(\mathbf{a}_t)_i, \varepsilon\}\right)^{\beta}, \tag{14}$$

$$\widetilde{a}_{t,i} = \begin{cases} 1, & \max_j c_{t,j} \le 0, \\ \max\left\{ \dfrac{c_{t,i} - \min_j c_{t,j}}{\max_j c_{t,j} - \min_j c_{t,j} + \varepsilon}, \varepsilon \right\}, & \text{otherwise.} \end{cases} \tag{15}$$

We use $\beta = 1$. Equation (1) initializes the first anchor using calibrated attention and nearest-neighbor distinctiveness within the frame. Let $\mathcal{S}_t^I$ denote the currently accepted I-frame anchors. After the first anchor, each remaining candidate is scored by

$$J_{t,i}^I(\mathcal{S}_t^I) = \widetilde{a}_{t,i} \max\left\{ \min_{w \in \mathcal{S}_t^I} d\big((t,i), w\big), \varepsilon \right\}. \tag{16}$$

The diversity term is updated after each accepted anchor. A proposed candidate whose maximum cosine similarity to an accepted same-frame anchor reaches τ_{sel} is recorded for merging rather than retained independently. Recorded members do not consume anchor slots or become diversity seeds. Selection continues until the offered budget is reached or no eligible candidates remain. All similarity checks use the original, unaggregated features.

Camera compensation. The affine camera-motion fit initially uses finite-motion positions at or below the 60th residual percentile. If fewer than eight such positions are available, all finite-motion positions are used. With fewer than eight finite-motion positions, the implementation uses median-motion subtraction. Otherwise, 96 batched three-point RANSAC hypotheses are evaluated with a 2.5-pixel inlier threshold and random seed 12345. The best valid inlier set is refitted; degenerate cases use a guarded least-squares fit. The original motion vectors, not the compensated vectors, define codec parents.

Calibrated predictive-update score. For each source P-frame, we separately calibrate the residual cue and camera-compensated motion using Eq. (13), and define

$$s_{t,i}^P = \mathcal{I}(r_{t,i}) + \mathcal{I}(o_{t,i}). \tag{17}$$

Each calibration is computed over the spatial positions of the corresponding source frame. For a token indexed by $v = (t, i)$, we write $s_v^P = s_{t,i}^P$. This score supplies future-reference support and the predictive scores used in background aggregation. Candidate exposure instead uses the raw evidence $e_{t,i} = r_{t,i} + o_{t,i}$, and frame-local P-frame diversity selection remains unweighted.

P-frame candidate exposure. Let $e_{t,i} = r_{t,i} + o_{t,i}$, using the residual notation of Eq. (2). For a frame with positive raw evidence, the initial pool contains positions satisfying

$$e_{t,i} \geq \max\{Q_{0.70}(\mathbf{e}_t), 0.5\}, \tag{18}$$

together with its top-$\min(8, N)$ raw-evidence positions. A frame with entirely zero evidence initially contributes no candidates. If the pooled GOP candidates fall below the exposure floor F_g^P, additional positions are taken in descending raw-evidence order from the remaining P positions. This supplementation can include zero-evidence positions. Let $\mathcal{A}_t$ denote the resulting pre-exclusion pool and $\mathcal{C}_t = \mathcal{A}_t \setminus \mathcal{E}_t$ its root-eligible subset. Excluded positions remain available as aggregation members. Sensitivity to the quantile level is reported in Table 7.

P-frame diversity selection. The local P selector retains exactly $k_t = \min\{b_t^P, |\mathcal{C}_t|\}$ representatives. When $0 < k_t < |\mathcal{C}_t|$, it uses Eq. (3): the first representative maximizes mean cosine distance to the other eligible candidates, and later representatives maximize minimum distance to the already selected representatives of the same P-frame. This stage uses no attention or codec-score weighting, no I-anchor or earlier-P-frame seed, and no selection-time merging. When $k_t = |\mathcal{C}_t|$, the entire eligible pool is retained without greedy traversal. Ties in the traversal favor the first candidate in ascending original-index order.

B.2 GOP Bounds, Integer Allocation, and Completion

Provisional allocation. GOP budgets are proportional to their original token capacities. Integer allocation first takes the floors and then distributes the remaining slots by descending fractional remainder, breaking ties by larger GOP capacity and then earlier GOP order. The allocated budgets sum to B. The default I-frame attention reservation is $\lceil N\rho_v \rceil$, clipped by the GOP budget and the frame's spatial capacity. This is an offered budget, not a guaranteed number of accepted anchors.

For a GOP with n_g^P P-frames and A_g accepted I anchors, the provisional P cap and candidate-exposure floor are

$$C_g^P = \mathrm{clip}\left(\left\lfloor \frac{n_g^P B_g}{n_g^P + \alpha_{\min}} \right\rfloor, 0, B_g - A_g\right), \tag{19}$$

$$F_g^P = \mathrm{clip}\left(\left\lceil \frac{n_g^P B_g}{n_g^P + \alpha_{\max}} \right\rceil, 0, C_g^P\right). \tag{20}$$

These are provisional constraints, not a final hard I:P ratio. The released full-method allocator fixes $\lambda = 0.50$ in Eq. (5). Capacity-bounded water filling is followed by largest-remainder rounding, with ties resolved by larger weight and then earlier frame index. Frame capacity counts all $N - \lfloor 0.04N \rfloor$ root-eligible P positions, not only the smaller codec candidate pool. The allocated budgets sum to C_g^P; an infeasible cap raises an error. A local P budget can remain unused because its eligible pool is smaller than the budget.

Update-guided reference recovery. Let $\mathrm{anc}(v)$ denote the I-frame ancestor of a predictive position v, obtained by following codec prediction links. Using the calibrated update score in Eq. (17), the support for an I-frame position w is

$$h_w = \max_{v:\mathrm{anc}(v)=w} s_v^P, \tag{21}$$

with zero support when no valid predictive descendant exists. The implementation computes this map through chronological parent ownership and scatter-maximum aggregation.

The candidate pool $\mathcal{U}_I^+$ contains non-retained I-frame positions satisfying $h_w > h_{\min}$, with $h_{\min} = 0$. Eligibility uses the original, undilated support map. Unlike final completion, this pool does not exclude every previously considered position: an unretained I-frame position may be reconsidered when supported by later updates.

At each proposal, let $\mathcal{U}_I^+$ contain the remaining eligible candidates and let $\mathcal{S}$ contain all currently retained representatives. The next candidate is

$$v^* = \arg\max_{v \in \mathcal{U}_I^+} \min_{w \in \mathcal{S}} d(v, w). \tag{22}$$

Support determines candidate eligibility but does not weight this diversity objective. Same-frame proposals with cosine similarity at least τ_{sel} are recorded for merging rather than retained independently. Proposed I additions are then capped per frame at $\lceil \gamma |\mathcal{S}_t^{I,0}| \rceil$, with $\gamma = 1$ by default. Excess proposals can be recorded for same-frame merging at $\tau_{\text{I-rem}}$. A subsequent same-frame cleanup removes retained I additions whose similarity to another retained representative reaches that threshold, releasing their slots for final completion.

Video-global completion. Remaining capacity is shared across the video rather than forced back into the local frame or GOP that left it unused. The candidate pool excludes retained positions, previously processed positions, and recorded merge sources. It can include P positions outside the original codec-gated pools. Candidates compete by unweighted max–min diversity against all retained representatives; compatible same-frame duplicates are recorded for merging at τ_{sel}. Attention-excluded P positions may be processed as merge members but can never become independent representatives.

The I-anchor and global selection routines use full-dimensional features. Global candidates use uniform calibration, while local P selection uses a separate framewise routine.

B.3 Merge Routes and Precedence

Aggregation starts only after the final representative set has been fixed and follows the route precedence summarized in Table 9. Every similarity decision uses original token features, not progressively averaged representatives. Recorded assignments are applied only if the source is not a retained representative, the destination is still retained, and the source has not already been assigned. Invalid recorded assignments are skipped; their sources may still be considered by later routes.

Table 9: Aggregation routes in execution order for the reported experiments. Local distance is Chebyshev distance on the token grid.

Source / stage	Destination and acceptance condition
Recorded assignment	Valid same-frame assignment from I-anchor selection, global selection, or I-addition cleanup. Selection uses τ_{sel}; I-addition cleanup uses $\tau_{\text{I-rem}}$. Local P selection records none.
Remaining I position	Most similar same-frame representative; cosine ≥ 0.85 and grid distance ≤ 4.
Unselected P codec candidate	Most similar same-frame P representative; cosine ≥ 0.80 within distance 4, otherwise ≥ 0.92. The candidate pool includes attention-excluded positions.
Predictable P background	Outside the codec candidate pool and at or below the video-wide score gate; follow codec parents to the first retained ancestor and require source–destination cosine ≥ 0.80.

The background score gate is the 65th percentile over the full video's unified score vector. P positions use s_v^P; I positions use the sum of ordinal information from attention, a density-peak score, and spatially pooled future support. The default density score uses seven nearest neighbors and support pooling uses a 3×3 maximum filter. This composite affects the shared merge gate, not the I-anchor ranking or the unweighted reference-completion objective.

All accepted assignments must satisfy the corresponding source-to-destination similarity criterion. Final coverage is applied to positions that remain unassigned after the preceding routes. A source is discarded if no compatible destination is found. Accepted members use unit weights, except for the predictable P-background route, which uses $\omega_{vw} = \text{clip}(\exp(-s_v^P), 10^{-6}, 1)$. Each accepted non-representative contributes at most once to the weighted average in Eq. (6), and the retained representative itself has unit weight.

C Qwen3-VL Source-Frame Adaptation

The adapter uses Qwen3-VL's native temporal patch size $k = 2$, without duplicating input frames or splitting a fused feature map into independent visual units. Inputs are prepared at 384×384 with

no further processor resize. The 24×24 patch grid is spatially merged into 12×12 native features per temporal unit. The standard 32-source-frame setting therefore produces 16 native visual grids; MotionBench uses eight source frames and produces four grids. Subsequent compression operates on these native feature positions.

Source-frame ledger and physical roots. Let $N = 144$ denote the number of native positions per temporal unit. Each source frame contributes $N/2$ positions of bookkeeping capacity. Source GOP budgets are allocated proportionally to their source-frame lengths, with integer allocation preserving the fixed video-level retention budget. The final budget counts unique physical positions, not source records.

The half-grid credit is an accounting weight rather than a requirement that both members of a temporal unit receive equal numbers of retained roots. Their realized allocations depend on codec roles, GOP budgets, candidate availability, and roots already retained. All source-role selections operate on the same fused features and share a global retained-root set. A position retained by an earlier role is passed as a preselected representative to subsequent selection calls and cannot consume another retained slot.

Codec evidence and source provenance. Codec scores, candidate masks, and prediction relationships are constructed on the source timeline before projection. Candidate records that map to the same physical position retain their contributing source identities. During local selection for a source P-frame, only candidates attributable to that frame are eligible, and calibration uses that frame's own codec scores.

Prediction links are also resolved on the source timeline. When a link maps within the same temporal unit, the source ancestor chain is followed before constructing a surviving link to an earlier physical unit. In particular, the P member of a PI unit remains assigned to the preceding source GOP and cannot provide future support to the new I-frame.

Reference and predictive selection. Source GOPs are processed in temporal order. For each GOP, the source I-frame first selects reference anchors on its mapped native grid, conditioned on the roots already retained. The provisional predictive budget is then computed using the source P-frame count and the same I:P-bound formulation as the framewise method, with the parameters reported in Table 8.

All source P-frames participate in this stage, including the P members of IP and PI units. Thus IP supports both an I role and a P role on one fused grid. In PI, the old-GOP P role and the new-GOP I role are processed within their respective source GOPs. For II, both source I-frames retain their own reference selection opportunities on the same physical grid.

P candidates are assembled within each source GOP, including provenance-preserving floor completion. The allocator distributes the GOP's provisional predictive budget among individual source P-frames using their individual packet costs q_t; it does not replace the two costs of a PP unit with $q_{2u} + q_{2u+1}$. The allocator retains the configured mixture of uniform and robust packet-bit weighting.

Source P-frames are then processed in temporal order. Each local selection uses its assigned cap and its own candidate provenance and calibration. Its diversity seed contains all physical roots retained so far, including representatives selected by preceding P-frame roles. Newly selected roots are immediately added to this shared set. Unspent local caps are left for the subsequent video-global completion stages.

Role-dependent attention exclusion. During selection for any source P-frame, the highest-attention $\lfloor 0.04N \rfloor = 5$ positions of its native grid are ineligible as new P roots. This leaves 139 positions before candidate, provenance, and already-selected-root restrictions.

For PP units, this exclusion remains active throughout root selection, including global completion. For mixed IP and PI units, it is applied only during their P-role selection: the same positions remain available to their I roles and to global completion. Exclusion concerns root eligibility, not whether a position can contribute as a merged member.

In the evaluated implementation, the provisional-cap allocator is passed the nominal per-source candidate-domain capacity $N = 144$; attention exclusion is enforced by the selector. Consequently, a provisional cap need not be fully consumed.

Update-guided reference recovery. After source-ledger reference and predictive selection, remaining video-level capacity enters I-remaining selection. A physical position in an I-containing unit is eligible when its projected future support exceeds the configured threshold and it has not already been retained.

The evaluated support mapping preserves source-I attribution. For II, support is traced separately to each source I-frame and combined by a pointwise maximum after mapping. For IP, support from later P descendants outside the current temporal unit is primary; internal-P support is used only when the mapped later-support pool is empty. The IP configuration uses candidate-gated source support. For PI, only valid later descendants of the new I-frame contribute; its old-GOP P member is excluded from that support.

I-remaining selection uses uniform calibration and is seeded with the already retained physical roots. The selected additions are capped per native reference unit according to the number of initial attention anchors retained there. They then undergo the same-unit root-similarity cleanup, using $\tau_{\text{I-rem}}$. Positions released by this cleanup do not increase the final retention budget.

Global completion and aggregation. Any capacity still remaining is filled from native positions that are neither retained, already processed, nor designated as forced-merge sources. The persistent PP attention-exclusion mask remains active during this stage; mixed-unit P-role masks do not become global exclusions.

The adapter requires the final root set to contain exactly the requested number of unique native positions. Main-branch aggregation, including the configured root-fixed GOP coverage aggregation, assigns eligible non-root features to the selected representatives without increasing the root count.

D Supplementary Ablation Protocol

Core controls. The ablations in Table 3 use LLaVA-Video and report the equal-weight average over VideoMME, MVBench, MotionBench, and LongVideoBench at each retention setting. All controls preserve the paired inputs, nominal outer token budget, inner-pruning schedule, and evaluator.

Attention-based P selection replaces within-frame diversity with descending-attention selection within the same eligible candidate pool $\mathcal{C}_t = \mathcal{A}_t \setminus \mathcal{E}_t$, subject to the same P-frame budget.

Without update-guided P candidates replaces $\mathcal{A}_t$ with the complete P-frame grid. Diversity selection is then applied to positions outside the unchanged attention-exclusion set $\mathcal{E}_t$. The budget-allocation and aggregation rules remain unchanged.

Uniform P-frame allocation replaces packet-dependent weights with equal weights while retaining the GOP P cap and legal capacities. The no-Update-guided-reference-recovery control removes the support-gated I-completion stage while preserving initial I anchors and final video-global completion.

Sink policy and rate. The attention-policy controls remove exclusion, extend it to both I and P frames, or replace it with equal-count random exclusion on P frames. Random masks are matched over the complete frame grid, not only over the codec-gated pool. Independent-token eligibility is distinguished from aggregation membership in every control. Floor rounding excludes seven positions on a 14×14 grid, six on a 13×13 grid, ten on a 16×16 grid, and five on a 12×12 grid. These counts specify the intervention and are not estimates of a universal visual-sink frequency.

E Limitations and Discussion

Reliability and scope of codec cues. Codec motion vectors are produced for predictive coding rather than precise object correspondence (Shou et al., 2019). Noisy or coarse displacements can affect both update evidence and proposed reference links. We combine camera-compensated motion and residual cues with feature diversity and feature-checked aggregation, but these mechanisms do

not make codec links exact semantic correspondences. The reported experiments use prepared H.264 streams with I/P prediction and no B-frames. Robustness across alternative coding configurations and reference structures remains to be evaluated.

Accuracy across compression budgets. Changing the token budget can alter reference–update allocation, the selected representatives, and the features aggregated into each representative. Representations at different budgets are therefore not necessarily nested, nor is frozen-model answer accuracy constrained to improve monotonically with token count. These changes offer a possible explanation for the nonmonotonic entries in Table 1, but the aggregate results alone do not identify which representation changes help or harm individual predictions.

Offline processing and future extensions. The current method is evaluated for offline video understanding. Its budget allocation, update-guided reference recovery, and global completion use information from the available clip. A streaming extension would require support accumulated from observed frames, online budget allocation, and bounded memory, building on causal token-compression approaches (Chen et al., 2026b). The reference–update organization provides a potential basis for such an extension, but its streaming effectiveness has not been evaluated here. Extending codec-guided allocation to audio–visual models would additionally require modality-specific cues and preservation of cross-modal temporal alignment.