DART: DECOUPLED AXIS-ADAPTIVE RECONSTRUCTION TREES FOR BUDGETED VIDEO TOKEN COMPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Video large language models incur substantial computational costs when processing dense frame-wise visual tokens. Compressing these tokens under a fixed budget requires deciding not only which regions to preserve, but also whether to retain temporal variation, spatial detail, or both. We formulate this decision as budgeted, attention-weighted reconstruction of the video feature volume and propose Decoupled Axis-Adaptive Reconstruction Trees (DART). DART represents the video as a partition of contiguous spatiotemporal regions, with one pooled token per region. A greedy planner applies temporal, spatial, and spatiotemporal partitioning to every candidate region, evaluates the reconstruction error of each resulting partition with respect to the original features within the region, and selects the region and partitioning strategy that yield the greatest reconstruction-error reduction per additional token. Adaptive-T further selects content-dependent boundaries for temporal-only splits, while spatial and spatiotemporal splits use midpoint boundaries. The resulting partition guides importance-weighted pooling of the original features before language-model prefilling, without query conditioning or updates to pretrained parameters. Averaged over four video-understanding benchmarks, DART retains 101.2% and 102.7% of the corresponding full-token baseline accuracy on LLaVA-OneVision-7B and 0.5B, respectively, while using only 25% of the visual tokens. These results support budgeted region refinement as an effective approach to allocating spatiotemporal resolution for video token compression.

1 INTRODUCTION

Video large language models (Video-LLMs) have demonstrated strong capabilities in video question answering and reasoning (Liu et al., 2023; Li et al., 2024a), but processing dense frame-wise visual tokens incurs substantial computation before text generation begins. In our LLaVA-OneVision setting, 32 sampled frames produce $32 \times 14 \times 14 = 6{,}272$ visual tokens after native spatial pooling. Reducing this visual prefix substantially decreases language-model computation, but aggressive compression must preserve the information required for downstream video understanding.

A central challenge is that different video regions require different amounts and types of representation capacity. A spatially simple region may change rapidly over time and require finer temporal resolution, whereas a persistent fine-scale pattern may benefit more from spatial detail; other regions contain both. Video token compression should therefore determine not only *where* to preserve information, but also *along which axes* resolution should be allocated under a fixed token budget.

Existing token-reduction methods primarily select individual representative tokens or group redundant ones through pruning, pooling, or similarity-based merging (Bolya et al., 2023; Marin et al., 2023; Shang et al., 2025; Yang et al., 2025; Alvar et al., 2025), and video-specific variants further exploit temporal redundancy across frames (Shao et al., 2025; Fan et al., 2026). We explore a complementary structured view: rather than forming arbitrary token groups, we maintain a complete, non-overlapping partition of the spatiotemporal feature volume and progressively allocate finer resolution to selected regions (Figure 1).

We formulate this process as budgeted, importance-weighted reconstruction and introduce **Decoupled Axis-Adaptive Reconstruction Trees (DART)**, a training-free, query-independent framework

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.

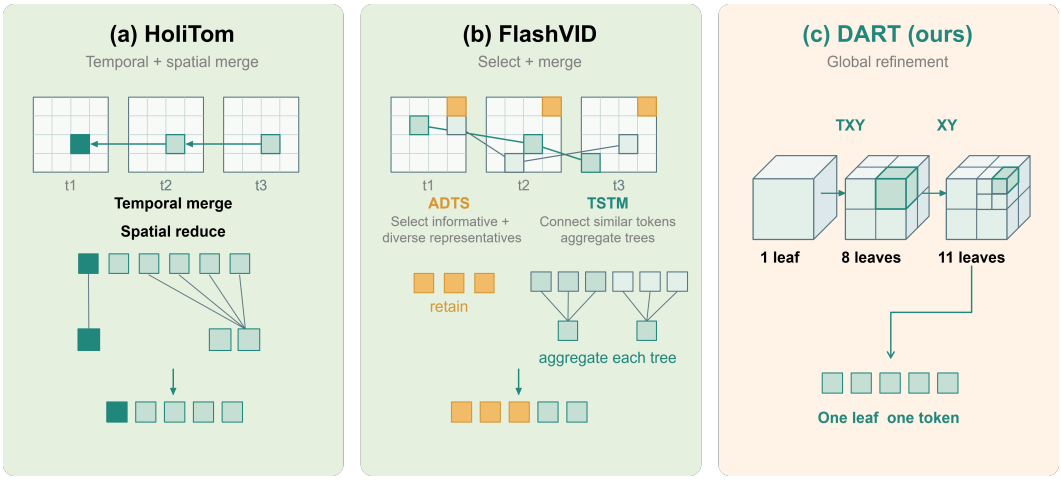


Figure 1: **Token aggregation versus region refinement.** (a) HoliTom merges and selects tokens sequentially. (b) FlashVID retains representatives and aggregates the rest via similarity trees. (c) DART maintains a complete spatiotemporal partition and refines selected regions under a global budget, with temporal (T), spatial (XY), and spatiotemporal (TXY) refinements competing under a shared reconstruction criterion and one token produced per final leaf.

in which each partition region produces one compressed token and additional capacity is allocated by reconstruction benefit per added token. Temporal (T), spatial (XY), and spatiotemporal (TXY) refinements compete under the same global budget, and **Adaptive-T** further selects content-dependent boundaries for temporal refinement. The final regions pool the original projected features into tokens that directly replace the dense visual prefix before LLM prefilling, with no learned compression module or parameter updates.

Experiments on MVBench (Li et al., 2024b), EgoSchema (Mangalam et al., 2023), LongVideoBench (Wu et al., 2024), and Video-MME (Fu et al., 2025a) demonstrate that DART preserves strong video-understanding performance under substantial visual-token reduction. At 25% pre-LLM retention, DART achieves average accuracies of 59.1 on LLaVA-OneVision-7B and 41.6 on its 0.5B counterpart (Li et al., 2024a). On the 7B model, DART obtains the highest average accuracy among the compared methods at 25%, 20%, and 15% retention, and remains within 0.1 points of the best reported result at 10% retention. These results indicate that explicitly allocating spatiotemporal representation capacity provides an effective alternative to conventional token-level selection and similarity-based aggregation.

The main contributions of this work are:

- **Budgeted spatiotemporal reconstruction.** We formulate video token compression as importance-weighted reconstruction over a complete partition of the video feature volume, allocating a fixed visual-token budget across space and time.
- **Decoupled axis-adaptive refinement.** We introduce DART, in which temporal, spatial, and spatiotemporal refinements compete by reconstruction benefit per additional token; Adaptive-T further places temporal boundaries according to video content.
- **Strong compression across model scales.** Across four benchmarks and two LLaVA-OneVision scales, DART preserves strong accuracy from 25% down to 10% token retention without additional training or query-dependent selection.

2 Related Work

Visual token compression. Visual-token reduction has been widely studied through pruning, merging, pooling, and adaptive computation. Token Pooling minimizes reconstruction error during downsampling (Marin et al., 2023), while ToMe merges similar tokens through lightweight bipartite

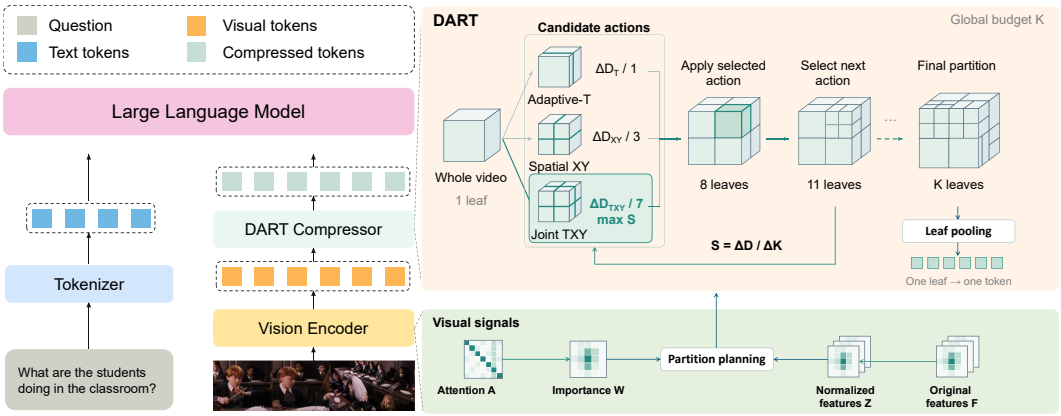


Figure 2: **Overview of the DART framework.** Given the feature volume $\mathbf{F}$ with $N = THW$ visual tokens, DART constructs a K-leaf spatiotemporal partition before LLM prefilling. A global best-first planner compares temporal (T), spatial (XY), and spatiotemporal (TXY) refinements of all leaves by reconstruction-error reduction per added token, using the importance map $\mathbf{W}$ and normalized features $\mathbf{Z}$; Adaptive-T searches content-dependent temporal boundaries while XY and TXY use midpoint cuts. Each final leaf pools the original features with importance weights into one compressed token, without query conditioning; the illustrated sequence is schematic.

matching (Bolya et al., 2023). For multimodal LLMs, FastV and SparseVLM reduce visual tokens within the language model using attention-based importance (Chen et al., 2024; Zhang et al., 2025), whereas PruMerge and VisionZip compress visual representations before LLM prefilling through saliency-aware selection and aggregation (Shang et al., 2025; Yang et al., 2025). Related approaches further exploit diversity, adaptive pruning ratios, or parameter-free pooling (Alvar et al., 2025; Ye et al., 2025; Yao et al., 2024). These methods establish token selection, merging, and reconstruction as effective compression strategies. DART instead allocates a global token budget over contiguous spatiotemporal regions, distributing capacity along temporal and spatial axes.

Efficient video understanding. Video compression additionally exploits redundancy across frames. LongVU combines frame removal, text-guided feature selection, and cross-frame spatial compression (Shen et al., 2025b), while PruneVid and FrameFusion use spatiotemporal redundancy or cross-frame similarity to reduce visual tokens (Huang et al., 2025; Fu et al., 2025b). HoliTom combines outer-LLM spatiotemporal merging with additional inner-LLM compression (Shao et al., 2025), whereas FlashVID retains selected representatives and aggregates remaining tokens through cross-frame similarity trees (Fan et al., 2026). EarlyTom moves compression into the vision encoder to reduce vision-side computation (Wang et al., 2026), and recent methods such as ForestPrune and MoPrune exploit structured spatial–temporal or scene-aware cues for token allocation (Ju et al., 2026; Hong et al., 2026). In contrast to token selection or similarity-based grouping, DART maintains a complete partition of the video feature volume, letting temporal, spatial, and spatiotemporal refinements compete under a shared reconstruction criterion and global budget.

3 Method

DART is a training-free video token compression framework that allocates a fixed visual-token budget through reconstruction-guided spatiotemporal partitioning. Rather than directly pruning individual tokens, DART represents the video feature volume as a set of contiguous regions and progressively refines regions that benefit most from additional representation capacity. The overall framework is illustrated in Figure 2.

3.1 Problem Formulation and Overview

Given T sampled video frames, the frozen vision encoder first extracts frame-wise visual features. These features are mapped into the language-model embedding space by the frozen multimodal

projector and then processed by the model's native spatial pooling, yielding a dense spatiotemporal feature volume $\mathbf{F} \in \mathbb{R}^{T\times H\times W\times d}$ with $N = THW$ visual tokens, where d is the language-model embedding dimension. For a target retention ratio $r \in (0, 1]$, we set the global visual-token budget to $K = \mathrm{round}(rN)$. All pretrained model parameters remain frozen. DART represents the complete feature volume as a partition of contiguous, non-overlapping spatiotemporal cuboids. Each leaf cuboid corresponds to one output visual token, so the number of leaves directly determines the compressed visual-sequence length. Starting from a single leaf covering the whole video, DART progressively refines selected regions while preserving complete coverage of the feature volume.

At each step, a leaf can be refined temporally (T), spatially (XY), or spatiotemporally (TXY), replacing one leaf with two, four, or eight children and increasing the token count by 1, 3, or 7, respectively. DART evaluates all feasible leaf-action pairs using a common reconstruction criterion, deciding both *where* additional representation capacity is allocated and *along which axes* the region is refined. Partition planning uses normalized projected features together with a question-independent visual importance signal; after refinement terminates, the final leaves pool the original projected features to construct the compressed visual sequence passed to the frozen language model. Thus, partition planning determines where capacity is allocated, whereas leaf pooling determines the content of the resulting tokens.

3.2 Importance-Weighted Reconstruction

Question-independent visual importance. DART uses self-attention from the frozen SigLIP vision encoder as a soft, question-independent importance prior. For each frame, we average the post-softmax attention received by each patch over all attention heads and query positions. The resulting patch scores are reshaped to the native vision grid, bilinearly resized to the $H \times W$ feature grid, and min–max normalized over all spatiotemporal positions of the video to obtain an importance map $\mathbf{W} = \{w_i\}$ with $w_i \in [0, 1]$. Importantly, the importance map does not directly select or discard visual tokens; it determines where reconstruction errors receive greater weight, so a region is favored only when subdividing it improves the importance-weighted reconstruction of its visual features.

Reconstruction distortion. Partition planning is performed on ℓ_2-normalized projected features, denoted by $\mathbf{Z} = \{z_i\}$, which reduces the influence of feature-norm variation when comparing alternative partitions. For a leaf C, the importance values w_i are used as reconstruction weights, falling back to uniform weighting if their total mass within C is numerically negligible; we denote the resulting effective weights by ω_i^C. The representative of C in the normalized planning space and its reconstruction distortion are

$$\mu_C = \frac{\sum_{i\in C} \omega_i^C z_i}{\sum_{i\in C} \omega_i^C}, \qquad D(C) = \sum_{i\in C} \omega_i^C \, \|z_i - \mu_C\|_2^2 \,. \tag{1}$$

For a partition $\mathcal{P}$, the total distortion is additive across its leaves, and DART seeks a low-distortion partition reachable through the refinement operations described below under the global token budget:

$$\min_{\mathcal{P}:\,|\mathcal{P}|\le K} \sum_{C\in\mathcal{P}} D(C). \tag{2}$$

The normalized features $\mathbf{Z}$ characterize feature variation within each region, whereas $\mathbf{W}$ determines where such variation matters more, yielding a common criterion for ranking candidate refinements.

3.3 Decoupled Axis-Adaptive Reconstruction Tree

Axis-adaptive refinement. For each leaf C, DART considers three refinement actions: temporal (T), spatial (XY), and spatiotemporal (TXY). The T action splits only the temporal interval and produces two children; XY bisects both spatial dimensions and produces four children while preserving the temporal extent; and TXY refines both the temporal and spatial dimensions to produce eight children. An action is considered only when the dimensions it refines can be split into non-empty intervals. Spatial and spatiotemporal refinements use midpoint splits. This decoupled design

allocates temporal and spatial resolution according to local content: a region with strong temporal variation but simple spatial structure is refined along time without unnecessary spatial subdivision, a stable region with fine spatial detail receives spatial refinement, and TXY remains available when both matter.

Adaptive temporal refinement. Temporal changes do not necessarily occur at the midpoint of a region. For a leaf spanning the temporal interval $[t_0, t_1)$, we consider all interior boundaries $\mathcal{B}_T(C) = \{t_0+1, \ldots, t_1-1\}$. Each candidate boundary τ produces two temporal children, C_τ^- and C_τ^+. Adaptive-T selects the boundary that yields the largest reduction in reconstruction distortion:

$$\tau^\star = \arg\max_{\tau \in \mathcal{B}_T(C)} \left[D(C) - D(C_\tau^-) - D(C_\tau^+) \right]. \tag{3}$$

Ties are resolved deterministically. Adaptive boundary selection is applied only to the temporal action, while spatial and spatiotemporal refinements retain midpoint boundaries.

Budget-aware refinement. For a candidate action a on leaf C, let $\mathrm{ch}(C, a)$ denote the resulting children. DART evaluates each candidate by the reduction in reconstruction distortion per additional output token:

$$S(C, a) = \frac{\max\left(0,\ D(C) - \sum_{C' \in \mathrm{ch}(C,a)} D(C')\right)}{|\,\mathrm{ch}(C, a)| - 1}. \tag{4}$$

This score provides a common criterion for comparing temporal, spatial, and spatiotemporal refinements with different token requirements. At each iteration, DART selects the highest-scoring valid leaf–action pair that fits within the remaining token budget and replaces the selected leaf with its children. The process is repeated over the updated partition until the target budget is reached or no further valid refinement can be applied without exceeding it.

3.4 Leaf Token Construction and LLM Integration

Once refinement terminates, each leaf of the final partition is converted into one visual token. Feature normalization is used only for partition planning; token construction operates on the original, non-normalized projected features $\mathbf{F}$ so that the resulting representations remain in the embedding space expected by the frozen language model. Using the same effective importance weights as in Section 3.2, the output representation of a final leaf C is

$$\hat{f}_C = \frac{\sum_{i \in C} \omega_i^C f_i}{\sum_{i \in C} \omega_i^C}. \tag{5}$$

When the total importance mass within a leaf is numerically negligible, the same fallback rule reduces this operation to ordinary mean pooling, and every location inside the leaf contributes to its representative. The final leaves are ordered lexicographically by their temporal and spatial starting coordinates, with tree depth used only as a tie breaker, and their pooled representations form a contiguous visual sequence of $K^\star \le K$ tokens that receive the model's ordinary consecutive positional indices, with no additional positional or region encoding.

Because $\mathbf{F}$ is already in the language-model embedding space, the compressed visual sequence directly replaces the dense visual block before prefilling, with no further multimodal projection. Partition construction depends only on the sampled video and the target token budget, not on the language query, so a partition can be constructed once for a video–budget pair and reused across all questions associated with that video.

4 Experiments

We evaluate DART as a budgeted representation of video content, focusing on accuracy across compression levels, transfer across model scales, adaptive partition behavior, and the computational cost

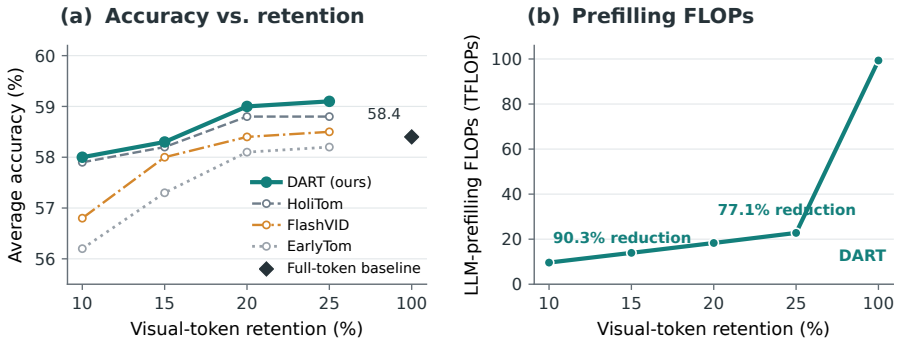


Figure 3: **Accuracy–efficiency trade-off of DART on LLaVA-OneVision-7B.** (a) Average accuracy across visual-token retention ratios for DART and representative recent baselines; the diamond denotes the full-token reference. (b) LLM-prefilling FLOPs for DART under the same retention settings, with the reduction measured relative to the full-token input.

of constructing the compressed representation. LLaVA-OneVision-7B provides the primary comparison; experiments on its 0.5B counterpart assess whether the same compression design remains effective with a smaller language model.

4.1 Experimental Setup

Models and benchmarks. We evaluate DART on four video understanding benchmarks: MVBench (Li et al., 2024b), EgoSchema (Mangalam et al., 2023), LongVideoBench (Wu et al., 2024), and Video-MME (Fu et al., 2025a). These benchmarks cover short-video understanding, egocentric reasoning, long-video question answering, and general video comprehension. We use LLaVA-OneVision-7B (Li et al., 2024a) as the primary evaluation model with all pretrained parameters frozen. We further evaluate LLaVA-OneVision-0.5B to examine whether the proposed compression strategy generalizes across model scales.

Baselines and metrics. We compare DART with representative visual-token compression methods, including FastV (Chen et al., 2024), PyramidDrop (Xing et al., 2025), DyCoke (Tao et al., 2025), VisionZip (Yang et al., 2025), PruneVid (Huang et al., 2025), FastVID (Shen et al., 2025a), HoliTom (Shao et al., 2025), EarlyTom (Wang et al., 2026), and FlashVID (Fan et al., 2026). The primary metric is multiple-choice accuracy. For efficiency evaluation, Section 4.4 reports LLM-prefilling FLOPs under the same language-model configuration. These FLOP measurements should not be interpreted as end-to-end latency or generation-throughput measurements.

Implementation details. Following LLaVA-OneVision, we uniformly sample 32 frames and obtain a $32 \times 14 \times 14$ visual feature volume containing 6,272 visual tokens. Unless otherwise specified, DART uses SigLIP layer-26 importance, normalized features, and the $\{T, XY, TXY\}$ refinement action set. We evaluate pre-LLM retention ratios of $\{25\%, 20\%, 15\%, 10\%\}$, corresponding to budgets $K \in \{1{,}568, 1{,}254, 941, 627\}$. The reported retention ratio counts visual tokens entering the LLM and excludes text and separator tokens. Detailed prompts, preprocessing settings, and implementation details are provided in Appendix A.

4.2 Main Results

Performance under aggressive compression. Beyond moderate compression, we further evaluate DART under increasingly strict token budgets. As shown in Table 1 and Figure 3, reducing the retention ratio from 25% to 10% decreases the average accuracy by only 1.1 points, while the visual token count is reduced by 60%. The degradation is mainly observed on benchmarks requiring fine-grained temporal reasoning, such as LongVideoBench and Video-MME. Nevertheless, DART maintains competitive performance, indicating that the adaptive partitioning strategy effectively preserves important spatiotemporal information under severe compression.

Table 1: Comparison of visual-token compression methods on LLaVA-OneVision-7B. Best results within each retention ratio are boldfaced.

Method	Before LLM Retained Ratio	MVBench ↑	EgoSchema ↑	LongVideo Bench↑	Video-MME ↑	Avg. Score ↑	Relative (%)↑
Full†	100%	58.3	60.4	56.4	58.6	58.4	100.0
FastV† ECCV'24	100%	55.9	57.5	56.7	56.1	56.5	96.7
PyramidDrop† CVPR'25	100%	56.1	58.0	54.1	56.4	56.2	96.2
DyCoke† CVPR'25	25%	53.1	59.5	49.5	54.3	54.1	92.6
VisionZip† CVPR'25	25%	57.9	60.3	56.5	58.2	58.2	99.7
PruneVid† ACL Findings'25	25%	57.4	59.9	55.7	57.4	57.6	98.6
FastVID† NeurIPS'25	25%	56.5	58.2	56.3	58.0	57.3	98.1
HoliTom† NeurIPS'25	25%	**58.4**	61.2	56.7	58.9	58.8	100.7
EarlyTom† CVPR'26	25%	57.4	60.5	56.3	58.5	58.2	99.7
FlashVID‡ ICLR'26	25%	58.0	60.2	**57.6**	58.3	58.5	100.2
DART (ours)	25%	**58.4**	**61.3**	57.3	**59.3**	**59.1**	**101.2**
VisionZip† CVPR'25	20%	57.7	59.8	55.2	57.9	57.7	98.8
FastVID† NeurIPS'25	20%	56.3	57.9	57.1	57.9	57.3	98.1
HoliTom† NeurIPS'25	20%	**58.7**	61.0	57.1	58.6	58.8	100.7
EarlyTom† CVPR'26	20%	57.8	60.6	55.6	58.0	58.1	99.5
FlashVID‡ ICLR'26	20%	57.8	60.1	**57.3**	58.3	58.4	100.0
DART (ours)	20%	57.9	**61.4**	**57.3**	**59.3**	**59.0**	**101.0**
VisionZip† CVPR'25	15%	56.5	59.8	54.4	56.1	56.7	97.1
FastVID† NeurIPS'25	15%	56.0	57.4	56.2	57.7	56.8	97.3
HoliTom† NeurIPS'25	15%	**58.1**	**61.2**	56.4	57.3	58.2	99.7
EarlyTom† CVPR'26	15%	57.5	60.2	54.4	56.9	57.3	98.1
FlashVID‡ ICLR'26	15%	57.3	60.0	**57.1**	57.4	58.0	99.3
DART (ours)	15%	58.0	**61.2**	56.1	**57.8**	**58.3**	**99.8**
VisionZip† CVPR'25	10%	53.5	58.0	49.3	53.4	53.5	91.6
FastVID† NeurIPS'25	10%	55.9	56.5	**56.3**	**57.3**	56.5	96.7
HoliTom† NeurIPS'25	10%	57.3	**61.2**	**56.3**	56.8	57.9	99.1
EarlyTom† CVPR'26	10%	56.5	60.1	52.4	55.8	56.2	96.2
FlashVID‡ ICLR'26	10%	56.9	59.7	54.2	56.4	56.8	97.3
DART (ours)	10%	**57.9**	60.8	56.0	**57.4**	**58.0**	**99.3**

Generalization across model scales. We further evaluate DART on LLaVA-OneVision-0.5B. As reported in Table 2, DART consistently maintains strong performance across different retention ratios. Although the smaller language model provides less reasoning capacity, DART still preserves effective visual representations, suggesting that the proposed partition strategy is not dependent on a specific model scale.

4.3 Partition Analysis

Figure 4 visualizes the spatiotemporal partitions generated by DART under different token budgets. Starting from the dense video feature volume, DART combines importance signals and feature variation to determine where additional representation capacity should be allocated. At 25% retention, DART produces fine-grained partitions for informative regions while maintaining coarse representations for less important areas. When reducing the budget to 10%, the resulting partition becomes coarser, but the allocation remains non-uniform across both spatial and temporal dimensions. The temporal allocation further shows that refinement is concentrated on specific positions rather than uniformly distributed over frames. This behavior shows that DART does not simply reduce tokens globally, but adaptively reallocates limited representation capacity according to video content.

Table 2: Accuracy on LLaVA-OneVision-0.5B across pre-LLM retention ratios. Best results within each retention ratio are boldfaced.

Method	Before LLM Retained Ratio	MVBench ↑	EgoSchema ↑	LongVideo Bench↑	Video-MME ↑	Avg. Score ↑	Relative (%)↑
Full	100%	45.5	26.8	45.8	43.7	40.5	100.0
HoliTom NeurIPS'25	25%	45.8	27.6	46.2	44.4	41.0	101.2
EarlyTom CVPR'26	25%	45.5	27.4	46.3	43.4	40.7	100.5
FlashVID ICLR'26	25%	45.6	27.4	**47.8**	44.1	41.2	101.7
DART (ours)	25%	**46.3**	**29.0**	46.5	**44.6**	**41.6**	**102.7**
HoliTom NeurIPS'25	20%	45.5	27.7	45.9	44.1	40.8	100.7
EarlyTom CVPR'26	20%	45.2	27.5	44.7	43.7	40.3	99.5
FlashVID ICLR'26	20%	45.6	27.2	**47.4**	43.6	41.0	101.2
DART (ours)	20%	**46.1**	**28.4**	45.9	**44.3**	**41.2**	**101.7**
HoliTom NeurIPS'25	15%	45.4	27.6	46.4	**43.4**	**40.7**	**100.5**
EarlyTom CVPR'26	15%	44.8	27.0	44.9	42.3	39.8	98.3
FlashVID ICLR'26	15%	44.9	27.1	**47.0**	42.4	40.4	99.8
DART (ours)	15%	**45.7**	**28.2**	45.7	43.0	**40.7**	**100.5**
HoliTom NeurIPS'25	10%	45.0	**27.3**	44.5	**43.3**	40.0	98.8
EarlyTom CVPR'26	10%	44.3	**27.3**	44.5	41.8	39.5	97.5
FlashVID ICLR'26	10%	44.7	26.7	**46.5**	41.7	40.0	98.8
DART (ours)	10%	**45.1**	27.1	46.2	42.7	**40.3**	**99.5**

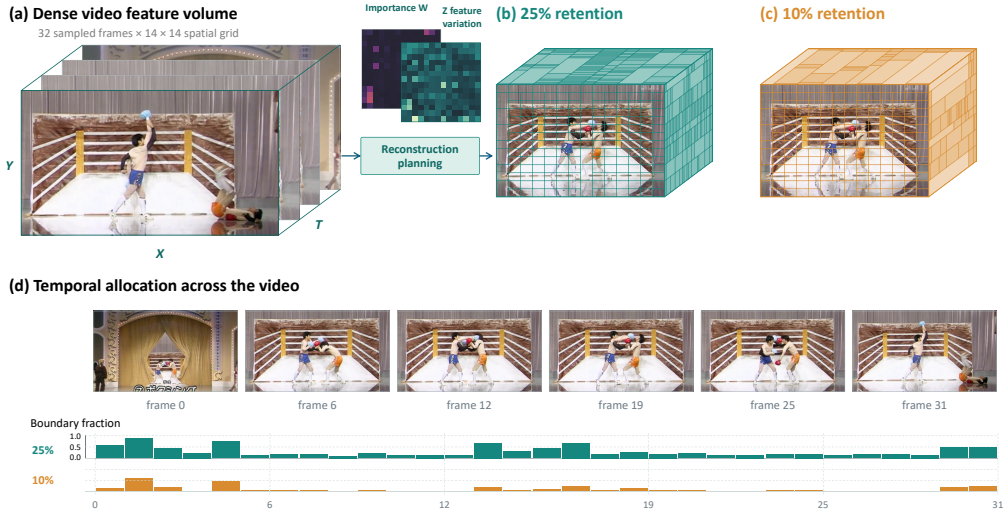


Figure 4: **Visualization of DART's adaptive spatiotemporal partitioning.** (a) Dense video feature volume from 32 sampled frames after native spatial pooling. Importance $\mathbf{W}$ and normalized feature variation derived from $\mathbf{Z}$ guide reconstruction-based planning. (b–c) Partitions of the same video at 25% and 10% pre-LLM retention. Smaller leaves correspond to finer representation, while larger leaves summarize broader spatiotemporal regions. (d) Temporal allocation for the same example at the two retention levels. For each pair of adjacent sampled positions, the bar height reports the fraction of the 14×14 spatial cells whose active final-leaf assignment changes. This quantity measures temporal-boundary density rather than token count or motion magnitude.

4.4 Inference Efficiency

To evaluate the computational benefit of visual-token compression, we measure the FLOPs required by the LLM prefilling stage under different token-retention budgets, using a structure-aware dense-prefilling FLOPs model of the Qwen2-7B decoder averaged over 100 evaluation videos (derivation

Table 3: LLM prefilling FLOPs under different DART retention ratios, averaged over 100 videos.

Retention	Tokens	Prefill FLOPs↓	Reduction↑
100%	6272	99.331	0.0%
25%	1568	22.786	77.1%
20%	1254	18.309	81.6%
15%	941	13.925	86.0%
10%	627	9.606	90.3%

Table 4: **Ablation of the importance-source layer.** Results use LLaVA-OneVision-7B at a target retention ratio of 25%.

Source	MVBench ↑	Video-MME ↑	LongVideo Bench ↑	Avg. ↑
Layer 7	58.0	57.2	56.3	57.2
Layer 10	57.9	57.3	**57.8**	57.7
Layer 14	58.1	57.9	56.3	57.4
Layer 17	57.7	57.5	57.7	57.6
Layer 26 (DART)	**58.4**	**59.3**	57.3	**58.3**

Table 5: **Ablation of the refinement design.** Results use LLaVA-OneVision-7B at a target retention ratio of 25%.

Variant	MVBench ↑	Video-MME ↑	LongVideo Bench ↑	Avg. ↑
w/o Adaptive-T	58.2	58.8	57.2	58.1
w/o XY	57.0	56.6	55.5	56.4
w/o TXY	58.3	58.7	57.1	58.0
TXY only	57.5	58.4	55.7	57.2
DART (full)	**58.4**	**59.3**	**57.3**	**58.3**

and model constants in Appendix C). All measurements use the same LLaVA-OneVision-Qwen2-7B model, prompt, input video protocol, and precision.

As shown in Table 3 and Figure 3(b), DART reduces the LLM prefilling computation by 77.1% at 25% token retention and by up to 90.3% at 10% retention; the prefilling cost decreases from 99.331 TFLOPs in the full-token setting to 9.606 TFLOPs, quantifying a substantial reduction in the LLM-prefilling stage.

4.5 Ablation Studies

Importance-source layer. We first investigate how the choice of importance source affects DART. Table 4 compares different SigLIP layers. Deeper layers generally provide more semantic information and achieve better performance, motivating our default choice of layer 26.

Axis-adaptive refinement. Table 5 isolates the effects of temporal boundary selection and the refinement action set. Replacing Adaptive-T with midpoint splitting reduces the average from 58.3 to 58.1, and removing the joint TXY action yields 58.0, while excluding the spatial XY action causes the largest degradation, to 56.4. Restricting the planner to TXY alone achieves 57.2, 1.1 points below the full configuration, confirming that spatiotemporal subdivision cannot replace decoupled temporal and spatial refinement. The best performance is obtained when DART selects among all three actions by reconstruction gain with adaptive temporal boundaries.

5 Conclusion

We presented DART, a query-independent framework for budgeted video token compression based on axis-adaptive refinement of a complete spatiotemporal partition. DART compares temporal, spatial, and spatiotemporal refinements by their attention-weighted reconstruction gain per added token, while Adaptive-T selects content-dependent temporal boundaries. The resulting leaves are pooled from the original visual features and replace the dense visual sequence before language-model prefilling. Across four video-understanding benchmarks, DART remains competitive under 25%–10% visual-token retention on both LLaVA-OneVision-7B and 0.5B. By shortening the visual sequence before language-model execution, DART reduces LLM-prefilling FLOPs by 77.1% at 25% retention and by up to 90.3% at 10% retention. These reductions require neither modifications to the pretrained language model nor query-dependent compression. More broadly, our results suggest that budgeted reconstruction over structured spatiotemporal regions offers a general principle for allocating limited representation capacity in video-language models.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, manuscript organization, retrieval and organization of related literature, experimental code development and debugging, and figure preparation. No generative AI tool was used to produce the reported experimental results; all accuracies and FLOPs measurements were obtained from actual model runs. The authors reviewed and verified all AI-assisted text, citations, figures, and code against the relevant sources and experimental records, and take full responsibility for the final content of this work, including all claims, analyses, and artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

This work uses publicly available video-understanding benchmarks and does not involve collecting new data, human-subject experiments, or interaction with individual participants. DART is a representation-level compression method for improving the efficiency of video understanding and does not introduce a deployment system for making decisions about people. We follow the intended research use of the benchmark datasets and rely on the licenses and usage conditions provided by their original creators. Potential limitations related to dataset bias and model-generated errors of the underlying video MLLMs remain applicable to our compressed models.

REPRODUCIBILITY STATEMENT

The DART framework, including the reconstruction objective, refinement actions, and Adaptive-T boundary selection, is described in Sections 3.1–3.4, with full implementation details in Appendix A. Section 4 and Appendix B specify the evaluated models, benchmarks, baselines, prompts, frame-sampling rules, and retention settings; Appendix C details the prefilling-FLOPs accounting, and Appendix D describes the deterministic planner configuration. The main results are reported in Tables 1–2. The implementation, evaluation configurations, and prompts will be made publicly available upon acceptance.

REFERENCES

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. DivPrune: Diversity-based visual token pruning for large multimodal models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

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

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, 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 *International Conference on Learning Representations*, 2026.

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

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 *IEEE/CVF International Conference on Computer Vision*, 2025b.

Wenhao Hong, Ziyang Wang, Yixin Zhang, and Zilei Wang. MoPrune: Scene-guided motion-aware token pruning for efficient video large language models. In *Findings of the Association for Computational Linguistics: ACL*, 2026.

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.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition Findings*, 2026.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024a.

Kunchang Li, Yali He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. MVBench: A comprehensive multi-modal video understanding benchmark. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024b.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, 2023.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. EgoSchema: A diagnostic benchmark for very long-form video language understanding. In *Advances in Neural Information Processing Systems*, 2023.

Dmitrii Marin, Jen-Hao Rick Chang, Anurag Ranjan, Anish Prabhu, Mohammad Rastegari, and Oncel Tuzel. Token pooling in vision transformers for image classification. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. LLaVA-PruMerge: Adaptive token reduction for efficient large multimodal models. In *IEEE/CVF International Conference on Computer Vision*, 2025.

Kele Shao, Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. HoliTom: Holistic token merging for fast video large language models. In *Advances in Neural Information Processing Systems*, 2025.

Leqi Shen, Guoqiang Gong, Tao He, Yifeng Zhang, Sicheng Zhao, Guiguang Ding, et al. FastVID: Dynamic density pruning for fast video large language models. In *Advances in Neural Information Processing Systems*, 2025a.

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, Zhuang Liu, Hu Xu, Hyunwoo J. Kim, Bilge Soran, Raghuraman Krishnamoorthi, Mohamed Elhoseiny, and Vikas Chandra. LongVU: Spatiotemporal adaptive compression for long video-language understanding. In *International Conference on Machine Learning*, 2025b.

Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. DyCoke: Dynamic compression of tokens for fast video large language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

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

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*, 2024.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, et al. PyramidDrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

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*, 2025.

Linli Yao, Lei Li, Shuhuai Ren, Lean Wang, Yuanxin Liu, Xu Sun, and Lu Hou. DeCo: Decoupling token compression from semantic abstraction in multimodal large language models. *arXiv preprint arXiv:2405.20985*, 2024.

Xubing Ye, Yukang Gan, Yixiao Ge, Xiao-Ping Zhang, and Yansong Tang. ATP-LLaVA: Adaptive token pruning for large vision language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. SparseVLM: Visual token sparsification for efficient vision-language model inference. In *International Conference on Machine Learning*, 2025.

Table 6: Legal DART refinement actions for an active spatiotemporal cuboid.

Action	Children	Token increment	Legality
T	2	1	Temporal extent ≥ 2
XY	4	3	Both spatial extents ≥ 2
TXY	8	7	Temporal and spatial extents ≥ 2

A Implementation Details

A.1 Visual Feature Pipeline

The compression pipeline operates after the frozen vision encoder, multimodal projector, and native spatial pooling, and before the frozen language model. For each video, we uniformly sample 32 RGB frames. The SigLIP So400m-patch14-384 vision encoder produces a 27×27 patch grid with 729 patch tokens per frame and no class token. The frozen multimodal projector maps the patch features into the Qwen2 hidden dimension $d = 3584$. LLaVA-OneVision's native bilinear spatial pooling subsequently produces a 14×14 grid. Consequently, DART receives

$$\mathbf{F} \in \mathbb{R}^{32\times14\times14\times3584}, \qquad N = 32 \times 14 \times 14 = 6{,}272. \tag{6}$$

DART replaces these N visual tokens with one token for each final cuboid leaf. The pooled DART features are inserted directly into the multimodal input sequence; the multimodal projector is not applied again after pooling.

A.2 Importance and Reconstruction Representations

The default importance signal is extracted from the 26th SigLIP encoder layer, corresponding to zero-based code index 25. The captured self-attention is the post-softmax attention. For each patch key position j, we average the attention received from all query positions and attention heads. The resulting 27×27 importance map is resized to 14×14 using bilinear interpolation and normalized using a global min–max operation over all spatiotemporal positions of the video. No class token or other special token is included. The numerical stabilizer used during normalization is 10^{-6}.

For partition planning, each projected feature is independently normalized as

$$\mathbf{z}_i = \frac{\mathbf{f}_i}{\max(\|\mathbf{f}_i\|_2, 10^{-6})}. \tag{7}$$

With the default importance power $p = 1.0$, the effective reconstruction weight is

$$\omega_i = w_i^p. \tag{8}$$

For a cuboid C, DART evaluates the weighted within-region reconstruction distortion defined in Section 3.2. If the total importance weight within a candidate cuboid is at most 10^{-6}, uniform weights are used for that cuboid. The same fallback reduces to ordinary mean pooling when the final visual token is constructed. Feature statistics are accumulated in float32, while the total importance-weight mass is accumulated in float64.

A.3 Refinement Planner

Each active cuboid admits three possible refinement actions, subject to the extent of the corresponding dimensions. Table 6 summarizes the legal actions and their token costs.

The planner constructs candidates for all legal action–leaf pairs and maintains them in a global best-first priority queue. For a candidate action a that partitions cuboid C into children $C_1, \ldots, C_m$, its priority is

$$S(C, a) = \frac{\max\Big(0,\ D(C) - \sum_{j=1}^{m} D(C_j)\Big)}{m - 1}, \tag{9}$$

where $D(\cdot)$ denotes the weighted reconstruction distortion. The denominator $m - 1$ is the number of additional visual tokens introduced by the split. Thus, the score measures the reduction in reconstruction distortion per additional token.

Candidates that would cause the partition to exceed the target token budget are skipped. Adaptive-T enumerates every legal interior temporal boundary for a temporal-only split. Spatial and joint spatiotemporal refinements use integer midpoint cuts. Ties among temporal boundaries are resolved first according to their proximity to the temporal midpoint and then according to the smaller boundary index. The deterministic action order used for remaining priority ties is

$$T \prec XY \prec TXY. \tag{10}$$

The four main token budgets are

$$K \in \{1{,}568,\ 1{,}254,\ 941,\ 627\}, \tag{11}$$

corresponding to retention ratios of 25%, 20%, 15%, and 10% relative to the original 6,272 visual tokens. The saved partition audits reach the requested value of K for every audited budget. Because DART is query-independent, the partition for a given video and token budget can be reused across all questions associated with that video.

A.4 Leaf Token Construction

After partition planning, each final cuboid leaf produces one visual token by pooling the original projected feature vectors $\mathbf{f}_i$ using the same effective importance weights employed by the planner:

$$\mathbf{v}_C = \frac{\sum_{i \in C} \omega_i \mathbf{f}_i}{\sum_{i \in C} \omega_i}. \tag{12}$$

When the denominator is negligible, DART applies the uniform-weight fallback described above. No additional within-leaf token pruning, learned token scorer, or language-query conditioning is applied.

B Evaluation Protocol

B.1 Benchmarks and Models

We evaluate DART on MVBench, EgoSchema, LongVideoBench, and Video-MME. The primary experiments use LLaVA-OneVision-7B, while LLaVA-OneVision-0.5B is used to evaluate transfer across model scales. All pretrained model parameters remain frozen.

Each benchmark retains its native question and answer format. Consequently, the benchmark-specific prompts, answer extraction procedures, and preprocessing protocols are not assumed to be identical across datasets. The same DART compression configuration is nevertheless used for all four benchmarks.

All main experiments use the 32-frame visual input described in Appendix A. The number of frames and the resulting $32 \times 14 \times 14$ feature grid remain fixed across benchmarks. The data loaders use benchmark-compatible video-decoding procedures. Specifically, the formal evaluation pipeline uses OpenCV-based sampling with rounded frame indices for MVBench and Video-MME, and Decord-based sampling with floor indexing for LongVideoBench and EgoSchema. This distinction does not affect the target visual-token budgets, but it is relevant for exact reproduction of the sampled frames.

B.2 Metrics and Aggregation

We report multiple-choice accuracy on each benchmark. The average score in Tables 1 and 2 is the unweighted arithmetic mean of the four displayed benchmark accuracies:

$$\mathrm{Avg} = \frac{1}{4} \sum_{b=1}^{4} \mathrm{Acc}_b. \tag{13}$$

The relative score reported together with the average is normalized by the corresponding full-token result:

$$\mathrm{Relative}(K) = 100 \times \frac{\mathrm{Avg}(K)}{\mathrm{Avg}(K_{\mathrm{full}})}. \tag{14}$$

The retention ratio is measured with respect to the visual-token sequence entering the language model. Text tokens and multimodal separator tokens are excluded from the reported token counts. The same target budgets are used for all datasets, and the final number of DART leaves equals the requested value of K in the audited partition plans.

B.3 Evaluation Consistency and Baseline Provenance

All results produced in this work use a single controlled evaluation pipeline. For each model scale, DART is evaluated with the same frozen LLaVA-OneVision checkpoint, 32-frame input, benchmark-specific prompt and answer-extraction protocol, video-decoding procedure, numerical precision, and visual feature grid across all retention ratios. The DART configuration is also fixed across benchmarks: the importance source, reconstruction objective, refinement action set, Adaptive-T rule, and leaf-pooling operation remain unchanged, and only the target token budget K is varied. The ablation experiments modify only the component identified by the corresponding ablation while retaining the same evaluation pipeline. These controls make comparisons among DART retention levels, model scales, and ablation variants internally consistent.

To place these controlled DART results in the context of recent methods, rows marked with † in Table 1 reproduce the published comparison values collected by EarlyTom for the same base model family, benchmarks, evaluation metric, and reported retention setting. We retain the method labels and averages reported in that comparison table. Recomputing an average from the displayed one-decimal benchmark scores can therefore differ slightly from an average calculated from the original unrounded values. The FlashVID rows marked with ‡ are transcribed from the corresponding updated comparison record, with retention interpreted as the fraction of visual tokens entering the language model.

The literature values are included to provide a common published reference point and were not re-evaluated in our codebase. Consequently, they should be read as literature-referenced comparisons rather than as a claim that every baseline was rerun with identical low-level implementations. Differences in prompt formatting, frame-index rounding, video decoders, or other implementation details may remain across the original studies. In contrast, all DART results and controlled analyses reported in this paper follow the unified configuration specified above and in Appendix A.

C Prefilling FLOPs Accounting

The efficiency analysis in Section 4.4 quantifies the dense LLM-prefilling computation after visual-token compression. For the Qwen2-7B language model, we use $L = 28$ transformer layers, hidden dimension $d = 3584$, SwiGLU intermediate dimension $m = 18{,}944$, 28 query heads, and 4 key/value heads. The effective key/value dimension is therefore

$$d_{\mathrm{kv}} = d\frac{h_{\mathrm{kv}}}{h} = 3584 \times \frac{4}{28} = 512. \tag{15}$$

For a total prefilling sequence length n, the structure-aware dense FLOPs estimate is

$$F_{\mathrm{prefill}}(n) = L\left(4nd^2 + 4ndd_{\mathrm{kv}} + 4n^2d + 6ndm\right). \tag{16}$$

This calculation counts one multiply–add operation as two FLOPs. The four terms respectively account for the query and output projections, the key/value projections under grouped-query attention, the two attention matrix multiplications, and the SwiGLU feed-forward network. The estimate omits normalization, nonlinear activation, embedding lookup, and output-logit operations, whose contributions are not central to the visual-token comparison.

For the efficiency subset containing $M = 100$ examples, the reported value at visual-token budget K is

$$\overline{F}_K = \frac{1}{M}\sum_{i=1}^{M} F_{\mathrm{prefill}}(n_{i,K}), \qquad n_{i,K} = N_{\mathrm{text},i} + K, \tag{17}$$

where $N_{\mathrm{text},i}$ denotes the number of non-visual input tokens for the i-th example. The corresponding relative reduction is

$$\mathrm{Reduction}(K) = 1 - \frac{\overline{F}_K}{\overline{F}_{K_{\mathrm{full}}}}. \tag{18}$$

These values isolate the computation of the dense LLM-prefilling stage. The SigLIP vision encoder is executed before DART and is unchanged across visual-token budgets. Attention extraction, partition planning, leaf pooling, video decoding, and autoregressive answer generation are not included in the reported prefilling FLOPs. The analysis therefore measures the reduction in language-model computation obtained by shortening the visual sequence, rather than treating the prefilling estimate as an end-to-end latency measurement.

D Additional Reproducibility Details

The main experiments use a single frozen DART configuration across benchmarks, model scales, and retention ratios; only the target token budget K changes. No benchmark-specific importance power, refinement action, reconstruction objective, or pooling rule is selected.

For fixed visual features, importance weights, token budget, and numerical environment, the planner is deterministic. Every legal candidate action is evaluated with the same weighted reconstruction objective, and queue ties are resolved using the deterministic rules described in Appendix A. The saved partition artifacts contain the target budget and final spatiotemporal bounds of every leaf. Together with the model configuration, frame-sampling rules, importance-layer definition, refinement action set, and pooling rule specified in this appendix, these artifacts determine the visual sequence supplied to the language model.

DART does not use the question text, answer options, or predicted answer when constructing the partition. Partition planning is performed using only the video-derived visual representations and importance weights. This separation allows the partition for a video to be reused across multiple associated questions and prevents compression from being conditioned on benchmark answer content.