TeMPO: Frame-Causal Token Compression for Efficient Video Large Language Models

Yingxin Lai[1*] **Bo Xu**[2,3*] **Yun-ze Pan**[2,3]
Zhiliang Zhu[2,3] **Baigui Sun**[2,3†] **Yang Liu**[2,3†]
[1]Xiamen University
[2]IROOTECH TECHNOLOGY [3]Wolf 1069 b Lab, Sany Group
laiyingxin2@gmail.com

Abstract

Video large multimodal models incur substantial inference cost because visual tokens accumulate across sampled frames. Training-free token compression can reduce this cost, but it often degrades video reasoning performance when later frames must contribute new evidence over time. We identify a common failure mode behind this degradation: existing compressors do not track the committed support, namely the token support already retained and delivered to the language model by earlier frames. As a result, selectors with different scoring rules repeatedly retain overlapping supports across neighboring frames, a phenomenon we call Selector Collapse. To address this issue, we propose TEMPO, a frame-causal token compression framework that combines committed-support memory, residual-energy greedy selection, and parameter-free neighborhood fusion. Across four video understanding benchmarks and multiple backbones, TEMPO consistently improves the accuracy-efficiency trade-off of training-free compression; at 10% retention, it preserves 99.8% of vanilla performance on LLaVA-OneVision and improves fixed-budget frame scaling on Qwen2.5-VL to 109.3% relative accuracy, without retraining, backbone modification, or additional token budget.

1 Introduction

Recent advances in Video Large Language Models (VLLMs) [2, 9, 31, 32] have demonstrated promising capabilities in video understanding. However, existing VLLMs typically encode each sampled frame as a sequence of visual tokens and feed the resulting tokens into the language model. As the number of frames increases, the visual-token sequence grows rapidly, leading to substantial inference overhead. Visual token compression has therefore become a key direction for efficient video reasoning.[1]

Recent training-free methods reduce visual tokens without retraining the model or modifying the backbone. Existing approaches select or merge tokens according to attention scores [3, 24], visual similarity and diversity [1, 17, 28, 29], text-conditioned relevance [30], residual-energy criteria [12], or cross-frame statistics [10, 18, 19, 21]. While these training-free approaches indeed improve model inference efficiency by reducing the number of visual tokens, a critical limitation remains: most of them still make retention decisions primarily from current-frame information. Such frame-local selection is insufficient for video reasoning, where later frames should contribute new evidence rather than repeatedly preserve visual support already delivered to the language model.

*Equal contribution.
†Corresponding authors.
[1]Code: https://github.com/laiyingxin2/TeMPO; project page: https://laiyingxin2.github.io/TeMPO/.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

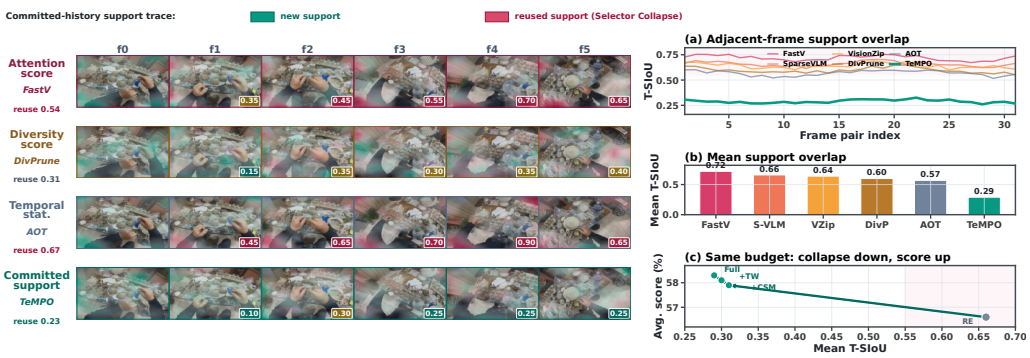
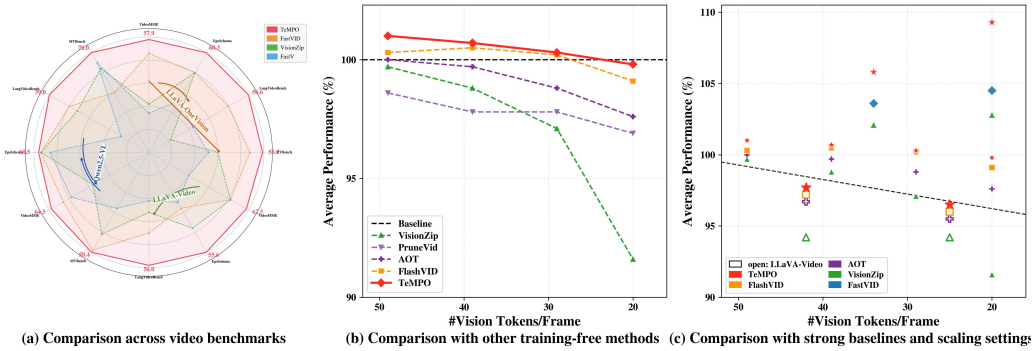


Figure 1: Selector Collapse in video token compression. Existing selectors show high adjacent-frame T-SIoU, indicating repeated support reuse across nearby frames, whereas TEMPO reduces this overlap by incorporating committed-support memory.



(a) Comparison across video benchmarks (b) Comparison with other training-free methods (c) Comparison with strong baselines and scaling settings

Figure 2: Overall comparison of TEMPO with representative token compression methods across video reasoning benchmarks, token budgets, and backbone settings.

Motivation. To diagnose this issue, we introduce Temporal Support IoU (T-SIoU), which measures the relative overlap between retained token supports in adjacent frames. A higher T-SIoU indicates more repeated support, whereas a lower value indicates greater temporal diversity. As shown in Fig. 1, representative selectors on LLaVA-Video-7B at 10% retention, including FastV [3], SparseVLM [30], VisionZip [25], DivPrune [1], and AOT [10], consistently exhibit high adjacent-frame support overlap. The support traces show that, although their scoring rules differ, these selectors repeatedly retain tokens whose features are close to those already committed in earlier frames. Thus, the limited token budget is often spent on visual evidence that has already been delivered to the language model. The T-SIoU curves and averages further show that this behavior persists across frames and across different selectors, while the same-budget ablation links lower support overlap with stronger video reasoning accuracy. We refer to this phenomenon as Selector Collapse. This observation suggests that the missing factor is not another local scoring rule, but a temporal state that records previously committed visual support. *How can we compress video tokens while avoiding repeated committed support across frames?*

Our Solution. We propose TEMPO, a training-free video token compression framework that turns frame-wise selection into a frame-causal decision process. TEMPO consists of two plug-and-play components. First, it maintains committed-support memory from the retained metric features of the previous frame and injects this memory into a residual-energy greedy selector. Each candidate token is evaluated by both its current-frame contribution and its novelty relative to previously committed support. Second, after the retained support is fixed, TEMPO applies parameter-free neighborhood fusion to aggregate discarded tokens into nearby retained tokens, thereby recovering local visual coverage without increasing the token count. In this way, TEMPO reduces temporal repetition while preserving spatial information under the same token budget.

Extensive experiments on multiple video reasoning benchmarks show that TEMPO consistently improves the accuracy-efficiency trade-off across token budgets and backbones. Our contributions are summarized as follows:

1. We identify Selector Collapse, a common failure mode in training-free video token compression, where selectors repeatedly retain overlapping supports across adjacent frames.
2. We propose TEMPO, a frame-causal compression framework that combines committed-support memory with residual-energy greedy selection and neighborhood fusion to reduce temporal repetition while preserving local visual coverage.
3. We validate TEMPO on multiple video reasoning benchmarks, demonstrating consistent improvements without retraining, backbone modification, or additional token budget.

2 Related Work

Frame-local token selection. Many training-free methods compress visual tokens independently for each image or frame. Attention-based approaches, including FastV [3], SparseVLM [30], VisionZip [25] and HiPrune [13] prune tokens using attention or attention-derived saliency. Diversity- and redundancy-aware methods, such as DivPrune [1], CDPruner [29], LLaVA-PruMerge [17], VisPruner [28], and FiCoCo [6] preserve representative visual coverage by removing duplicate tokens. Residual-energy methods such as ResPrune [12] select tokens by the residual directions they explain under a subspace reconstruction objective. These methods provide effective within-frame compression, but their decision state resets at each frame, allowing similar evidence to be committed repeatedly across neighboring frames.

Video token compression with temporal statistics. Video-specific compressors incorporate temporal information into token allocation or merging. DyCoke [21], PruneVid [7], FastVID [19], HoliTom [18], AOT [10], MMG-Vid [15], VidCom^2 [14], and StreamForest [26] exploit cross-frame structure through windows, anchors, clusters, global redundancy statistics, trajectories, or segment-level representations. Closely related methods such as SURGE [20] and STC [22] further use past features to estimate surprise or temporal context. However, these histories are typically represented as feature statistics or continuous temporal priors, rather than as an explicit record of the support already delivered to the language model. TEMPO differs in the state it preserves: the previous frame's retained support is stored as a permutation-invariant set of metric features and written back into the current selection rule.

3 Method

3.1 Overview

The preceding analysis suggests that the main limitation of existing training-free compressors is not the lack of a strong within-frame score, but the absence of a causal record of committed support. TEMPO therefore keeps a residual-energy selector and augments it with this missing temporal state. The selection rule changes from an open-loop per-frame decision to a frame-causal decision: a token is valuable if it explains the current frame and contributes evidence not already covered by recent committed support.

Let a video contain F frames. For frame t, the vision encoder produces visual features

$$X_t = [x_1^t, \dots, x_T^t]^\top \in \mathbb{R}^{T \times D},$$

where T is the number of visual tokens and D is the feature dimension consumed by the language model. Selection is performed in a metric space

$$M_t = [m_1^t, \dots, m_T^t]^\top \in \mathbb{R}^{T \times d},$$

where $m_i^t \in \mathbb{R}^d$ is the metric representation of token i. When the vision tower exposes a suitable key projection, we use it as M_t; otherwise, we set $M_t = X_t$ and $d = D$. Given retention ratio $\rho \in (0, 1]$, the per-frame budget is

$$k = \max\big(1,\ \lfloor \rho T \rfloor\big). \tag{1}$$

The compressor outputs a support $S_t \subseteq \{1, \dots, T\}$ with $|S_t| = k$ and refined features $\widetilde{X}_t = \{\widetilde{x}_a^t : a \in S_t\}$, which are forwarded to the language model. The metric space $\mathbb{R}^d$ is used for scoring and routing, whereas the feature space $\mathbb{R}^D$ is used for the final visual tokens consumed by the language model.

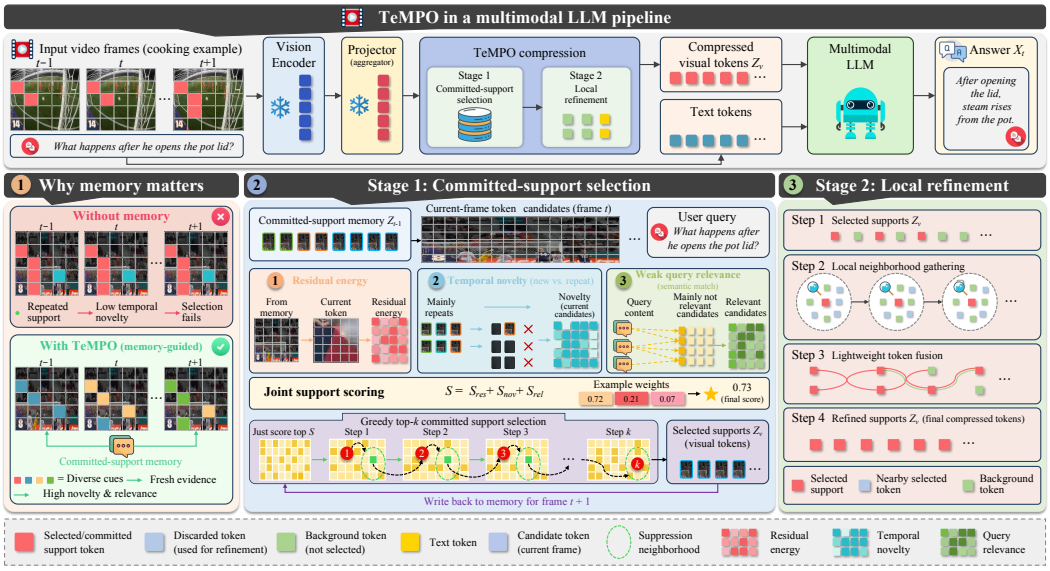


Figure 3: The framework of TEMPO. The upper row presents the overall frame-causal video token compression pipeline. Committed-support memory guides residual-energy token selection. Joint scoring combines residual energy, temporal novelty, and text guidance. Neighborhood fusion restores local coverage without increasing the token count.

3.2 Backbone: residual-energy greedy selection

The backbone of TEMPO is residual-energy greedy selection, which encourages within-frame complementarity. Given M_t, token selection is formulated as a reconstruction problem:

$$\min_{S_t,\, W_t} \; \|M_t - W_t M_t[S_t]\|_F^2, \quad \text{s.t. } |S_t| = k, \tag{2}$$

where $M_t[S_t] \in \mathbb{R}^{k\times d}$ stacks the selected metric vectors and $W_t \in \mathbb{R}^{T\times k}$ is the reconstruction weight matrix. This objective selects a support set that explains the current-frame representation, rather than a set of independently high-saliency tokens.

At greedy step s, let $S_t^{(s)}$ be the current support and $M_t^{(s)} = M_t[S_t^{(s)}]$. For $s = 0$, the residual is initialized as $R_t^{(0)} = M_t$. For $s > 0$, the reconstruction coefficients and residual are

$$W_t^{(s)} = M_t (M_t^{(s)})^\top \big(M_t^{(s)} (M_t^{(s)})^\top + \epsilon I\big)^{-1}, \qquad R_t^{(s)} = M_t - W_t^{(s)} M_t^{(s)}, \tag{3}$$

where $\epsilon > 0$ is a small stabilizer. For a candidate $i \notin S_t^{(s)}$, the residual-energy score is

$$c_i^{(s)} = \left\| R_t^{(s)} (m_i^t)^\top \right\|_2. \tag{4}$$

A larger $c_i^{(s)}$ indicates that token i explains directions not yet covered by the current support. Because the score is recomputed after each greedy update, this backbone discourages redundant selections within the same frame.

However, this residual state is local to a single frame. At frame $t+1$, the selector restarts from $R_{t+1}^{(0)} = M_{t+1}$. When neighboring frames are visually similar, this reset can make the greedy trajectory repeat, causing S_{t+1} to overlap strongly with S_t. Thus, residual-energy selection addresses within-frame redundancy but leaves cross-frame repetition unresolved. Committed-support memory is introduced to fill this gap.

3.3 Temporal novelty from committed support

To make selection frame-causal, TEMPO stores the metric features of the previous frame's retained support:

$$Z_{t-1} = [z_1^{t-1}, \ldots, z_k^{t-1}]^\top \in \mathbb{R}^{k\times d}, \qquad z_j^{t-1} = m_{a_j}^{t-1}, \qquad a_j \in S_{t-1}. \tag{5}$$

This memory is compact and causal: its size is $\mathcal{O}(kd)$, it does not grow with the number of frames, and it depends only on previous decisions. Most importantly, each row corresponds to the original selected support of a token delivered to the language model after neighborhood fusion, rather than to a pooled or clustered temporal summary. By default we use $K = 1$; the generalization $Z_{t-1}^{(K)} = \bigcup_{\tau=1}^{K} Z_{t-\tau}$ extends the memory to a window of K frames, replacing Z_{t-1} in Eq. (6); the maximum then ranges over all slots in that window. The K ablation is reported in Sec. 4.4.

Given Z_{t-1}, we measure whether a candidate repeats previously committed evidence by comparing it with the closest memory slot:

$$\sigma_i^t = \max_{1 \le j \le k} \cos\big(\overline{m}_i^t, \overline{z}_j^{t-1}\big), \qquad n_i^t = \frac{1 - \sigma_i^t}{2}, \qquad \widetilde{n}_i^t = \frac{n_i^t}{\max_{i'} n_{i'}^t + \epsilon}, \tag{6}$$

where overlines denote ℓ_2 normalization. The maximum is deliberate: if a candidate is close to any committed slot, it should be treated as repeated evidence even if it is dissimilar to other slots. The novelty term therefore answers a per-candidate question: has this kind of evidence already been committed to the language model? For the first frame, $Z_0 = \emptyset$ and $\widetilde{n}_i^1 = 0$, so the temporal branch contributes no score.

Mechanism check: temporal eligibility decay. In event-dense scenarios, the same spatial region may carry new evidence across frames, and rigid suppression by committed-support memory can discard such evidence. As a diagnostic, we decay each slot's suppressive weight by its persistence:

$$\sigma_i^t = \max_j w_j \cos(\overline{m}_i^t, \overline{z}_j^{t-1}), \qquad w_j = \exp(-\gamma p_j),$$

where p_j counts how many consecutive frames slot j has persisted. A larger γ progressively reduces the suppressive influence of stale slots, so a candidate similar to a long-lived anchor remains eligible for re-selection. We expect this to benefit event-dense benchmarks more than static long-context ones; with $\gamma = 0$, the decay reduces to the default committed-support memory rule.

3.4 Text warm-start and joint scoring

Residual energy captures visual explanatory power, but it can favor visually dominant background regions that are irrelevant to the query. We therefore add a task-aware text warm-start. Let $Q = \{q_\ell\}_{\ell=1}^{N_q}$ be query tokens projected into the same metric space as M_t. We compute a normalized query prototype and a clipped alignment score:

$$\bar{q} = \overline{\frac{1}{N_q} \sum_{\ell=1}^{N_q} q_\ell}, \qquad a_i^t = \max\big(0, \cos(\overline{m}_i^t, \bar{q})\big), \qquad \widetilde{a}_i^t = \frac{a_i^t}{\max_{i'} a_{i'}^t + \epsilon}. \tag{7}$$

The clipping prevents negatively correlated tokens from receiving negative scores. In text-free settings, this branch can be disabled by setting $\lambda_{\text{text}} = 0$.

At step s of frame t, the residual score is normalized as

$$\widetilde{c}_i^{(s)} = \frac{c_i^{(s)}}{\max_{i'} c_{i'}^{(s)} + \epsilon}.$$

The joint score is then

$$\boxed{\psi_i^{(s,t)} = \underbrace{\widetilde{c}_i^{(s)}}_{\text{current residual}} + \lambda_{\text{tmp}} \underbrace{\widetilde{n}_i^t}_{\text{committed-support novelty}} + \lambda_{\text{text}} \underbrace{\widetilde{a}_i^t}_{\text{text guidance}}} \tag{8}$$

where λ_{tmp} and λ_{text} control temporal novelty and text guidance. All three terms are normalized before combination, so the weights act as explicit trade-off coefficients. The greedy update is

$$i^\star = \arg\max_{i \notin S_t^{(s)}} \psi_i^{(s,t)}, \qquad S_t^{(s+1)} = S_t^{(s)} \cup \{i^\star\}, \tag{9}$$

followed by the residual update in Eq. (3). After k steps, the selected support is $S_t = S_t^{(k)}$.

The placement of novelty inside the greedy loop is important. A pre-filter that removes memory-similar tokens may discard evidence needed to reconstruct the current frame, whereas a post-hoc replacement is applied only after the budget has already been spent. In Eq. (8), temporal novelty competes with residual energy at every selection step, turning the selector from an open-loop per-frame rule into a closed-loop frame-causal rule. When $\lambda_{\text{tmp}} = \lambda_{\text{text}} = 0$, TEMPO reduces exactly to the residual-energy backbone.

3.5 Neighborhood fusion: restoring local coverage

Committed-support memory changes which tokens are retained across frames, but more novel supports may leave nearby discarded regions underrepresented. After the final support is fixed, TEMPO restores local coverage by routing each discarded token $u \notin S_t$ to its most similar retained token in metric space:

$$\pi(u) = \arg\max_{a \in S_t} \cos\big(\overline{m}_u^t, \overline{m}_a^t\big). \tag{10}$$

This assignment defines the neighborhood of each retained token,

$$\mathcal{N}(a) = \{u \notin S_t : \pi(u) = a\}.$$

We then aggregate the original D-dimensional features within each neighborhood:

$$\widetilde{x}_a^t = \begin{cases} \frac{1}{2}\left(x_a^t + \frac{1}{|\mathcal{N}(a)|}\sum_{u \in \mathcal{N}(a)} x_u^t\right), & |\mathcal{N}(a)| > 0, \\ x_a^t, & \text{otherwise.} \end{cases} \tag{11}$$

Routing uses the metric space $\mathbb{R}^d$, aligned with the novelty computation, whereas aggregation updates the D-dimensional features consumed by the language model. The memory passed to the next frame remains the selected metric support,

$$Z_t = M_t[S_t], \tag{12}$$

rather than the fused features. Thus, the language model receives locally enriched tokens, while the next-frame selector records which original support was committed.

4 Experiments

4.1 Experimental Settings

Benchmarks. We evaluate TEMPO on four widely used video understanding benchmarks: MVBench [11], EgoSchema [16], LongVideoBench [23], and VideoMME [5], providing a comprehensive testbed across video lengths and reasoning settings.

Implementation Details. Our method is implemented on LLaVA-OneVision-7B [9] and LLaVA-Video-7B [31], and is further extended to Qwen2.5-VL-7B [2] for the fixed-budget frame-scaling study. Both evaluation and inference use NVIDIA H200 GPUs. Inference cost is measured by prefilling FLOPs, with baselines configured under comparable FLOP budgets following prior work [3, 18, 21, 24]. Unless otherwise stated, the weighting coefficients of TEMPO are fixed to $\lambda_{\text{tmp}} = 0.2$ and $\lambda_{\text{text}} = 0.3$, with anchor decay disabled by default ($\gamma = 0$). The same configuration is shared across all retention ratios and backbones without task-specific tuning. All benchmarks are conducted using LMMs-Eval [8, 27].

4.2 Main Results

Results on LLaVA-OneVision. Tab. 1 compares TEMPO with representative methods on LLaVA-OneVision-7B under four token retention budgets. Inner-LLM pruning methods such as FastV and PDrop show a weaker accuracy-efficiency trade-off at low retention ratios, while frame-local compressors such as VisionZip retain only 91.6% of vanilla performance at 10% retention. In contrast, TEMPO achieves the best average score across all four budgets, with larger gains under tighter compression. At 25% retention, it surpasses the vanilla model by 0.6 points; at 10% retention, it preserves 99.8% of vanilla performance and outperforms FlashVID, AOT, and VisionZip by 0.4, 1.3, and 4.8 points, respectively.

Table 1: Comparison with representative methods on LLaVA-OneVision [9] across video benchmarks. The best performance among methods with similar retention ratios is highlighted in **bold**, while the second best is denoted as underlined.

Method	Prefilling FLOPs (T) ↓	FLOPs Ratio ↓	Before LLM Retained Ratio	MVBench ↑	EgoSchema ↑	LongVideo Bench ↑	VideoMME ↑	Avg. ↑ Score	Avg. ↑ %
LLaVA-OV-7B	40.8	100%	100%	58.3	60.4	56.4	58.6	58.4	100
FastV [3]	9.3	22.8%	100%	55.9	57.5	56.7	56.1	56.5	96.7
PDrop [24]	10.5	25.7%	100%	56.1	58.0	54.1	56.4	56.2	96.2
DyCoke [21]	8.7	21.3%	25%	53.1	59.5	49.5	54.3	54.1	92.6
VisionZip [25]	8.7	21.3%	25%	57.9	60.3	56.5	58.2	58.2	99.7
PruneVid [7]	8.7	21.3%	25%	57.4	59.9	55.7	57.4	57.6	98.6
FastVID [19]	8.7	21.3%	25%	56.5	–	56.3	58.0	–	–
AOT [10]	8.7	21.3%	25%	58.7	61.3	56.3	57.5	58.5	100.0
FlashVID [4]	–	–	25%	58.0	60.4	56.8	**59.2**	58.6	100.3
TeMPO (Ours)	8.7	21.3%	25%	**58.8**	**61.4**	**57.0**	58.9	**59.0**	**101.0**
VisionZip [25]	7.0	17.2%	20%	57.7	59.8	55.2	57.9	57.7	98.8
PruneVid [7]	7.0	17.2%	20%	57.2	59.7	54.7	56.9	57.1	97.8
FastVID [19]	7.0	17.2%	20%	56.3	–	57.1	57.9	–	–
AOT [10]	7.0	17.2%	20%	58.1	**61.3**	56.2	57.2	58.2	99.7
FlashVID [4]	–	–	20%	58.2	60.1	**58.5**	58.2	58.7	100.5
TeMPO (Ours)	7.0	17.2%	20%	**58.6**	61.0	57.4	**58.5**	**58.9**	**100.7**
VisionZip [25]	5.2	12.7%	15%	56.5	59.8	54.4	56.1	56.7	97.1
PruneVid [7]	5.2	12.7%	15%	56.8	59.7	55.4	56.6	57.1	97.8
FastVID [19]	5.2	12.7%	15%	56.0	–	56.2	57.7	–	–
AOT [10]	5.2	12.7%	15%	57.8	**61.3**	55.2	56.6	57.7	98.8
FlashVID [4]	–	–	15%	57.9	60.4	**57.5**	**58.2**	58.5	100.2
TeMPO (Ours)	5.2	12.7%	15%	**58.5**	60.7	57.0	58.1	**58.6**	**100.3**
VisionZip [25]	3.4	8.3%	10%	53.5	58.0	49.3	53.4	53.5	91.6
PruneVid [7]	3.4	8.3%	10%	56.2	59.8	54.5	56.0	56.6	96.9
FastVID [19]	3.4	8.3%	10%	55.9	–	56.3	57.3	–	–
AOT [10]	3.4	8.3%	10%	57.0	**60.6**	54.2	56.1	57.0	97.6
FlashVID [4]	–	–	10%	57.4	60.0	56.5	57.8	57.9	99.1
TeMPO (Ours)	3.4	8.3%	10%	**58.5**	60.3	**56.6**	**57.9**	**58.3**	**99.8**

Table 2: Comparison with representative methods on LLaVA-Video [31] across video benchmarks.

Method	Prefilling FLOPs (T) ↓	FLOPs Ratio ↓	Before LLM Retained Ratio	MVBench ↑	EgoSchema ↑	LongVideo Bench ↑	VideoMME ↑	Avg. ↑ Score	Avg. ↑ %
LLaVA-Video-7B	80.2	100%	100%	60.4	57.2	58.9	64.3	60.2	100
FastV [3]	17.1	21.3%	100%	54.3	54.1	55.0	58.8	55.6	92.4
PDrop [24]	19.5	24.3%	100%	55.9	54.3	54.7	61.9	56.7	94.2
DyCoke [21]	15.2	18.9%	25%	50.8	–	53.0	56.9	–	–
VisionZip [25]	15.2	18.9%	25%	56.7	54.7	54.7	60.7	56.7	94.2
AOT [10]	15.2	18.9%	25%	58.8	55.4	56.2	62.4	58.2	96.7
FlashVID [4]	–	–	25%	59.0	55.6	56.7	62.5	58.5	97.2
TeMPO (Ours)	15.2	18.9%	25%	**59.4**	**55.9**	**57.2**	**62.8**	**58.8**	**97.7**
VisionZip [25]	9.3	11.6%	15%	56.7	54.7	54.7	60.7	56.7	94.2
AOT [10]	9.3	11.6%	15%	57.8	55.2	55.0	62.0	57.5	95.5
FlashVID [4]	–	–	15%	58.0	55.3	55.5	62.2	57.8	96.0
TeMPO (Ours)	9.3	11.6%	15%	**58.4**	**55.6**	**56.0**	**62.5**	**58.1**	**96.5**

Results on LLaVA-Video. LLaVA-Video poses a stronger compression challenge due to its more aggressive initial pooling (169 vs. 196 tokens per frame). As shown in Tab. 2, TEMPO achieves the best average score at both 25% (97.7%) and 15% (96.5%) retention, outperforming AOT and FlashVID by 0.6 and 0.3 points at both budgets. These results show that frame-causal novelty selection remains effective across different video backbones and token budgets.

Superior performance after token reduction. As shown in Tab. 1, TEMPO can surpass the vanilla baseline on several benchmarks at moderate retention, suggesting that excessive or redundant visual tokens can dilute task-relevant evidence at the LLM input. By allocating the limited budget to tokens carrying new temporal evidence and recovering local context through parameter-free fusion, TEMPO improves the effective visual signal under compression.

Results on Qwen2.5-VL under a fixed token budget. Existing VLLMs often process a limited number of sampled frames due to compute constraints and may therefore miss important visual cues. Token compression allows more frames to be processed under the same budget. As shown in Tab. 3, Qwen2.5-VL consistently benefits from longer temporal context when equipped with compression.

Table 3: Comparison with representative methods on Qwen2.5-VL [2] under a fixed token budget.

Method	#Frames	Retention Ratio R	MVBench ↑	EgoSchema ↑	LongVideo Bench ↑	VideoMME ↑	Avg. ↑ Score	Avg. ↑ Rel. Acc (%)
Vanilla	16 (1×)	100%	65.0	56.0	55.0	56.5	58.1	100.0
VisionZip [25]	80 (5×)	20%	65.5	57.0	55.5	59.0	59.3	102.1
FastVID [19]	80 (5×)	20%	66.5	57.5	56.5	60.5	60.2	103.6
TeMPO (Ours)	80 (5×)	20%	**67.5**	**58.5**	**57.5**	**62.5**	**61.5**	**105.8**
VisionZip [25]	160 (10×)	10%	66.5	57.5	56.0	59.0	59.7	102.8
FastVID [19]	160 (10×)	10%	67.0	58.0	57.0	60.7	60.7	104.5
TeMPO (Ours)	160 (10×)	10%	**70.0**	**60.5**	**59.0**	**64.5**	**63.5**	**109.3**

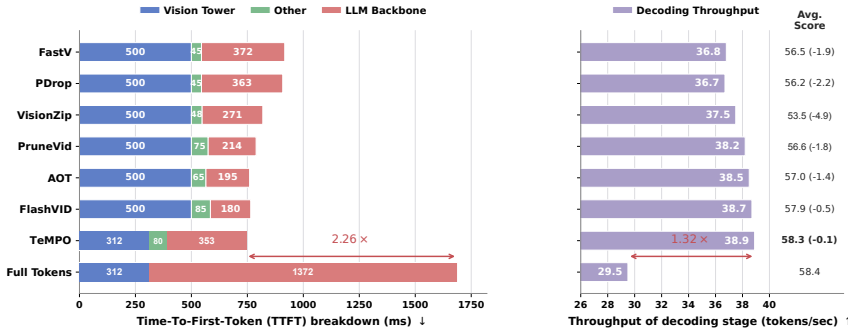


Figure 4: Efficiency comparison of TEMPO on LLaVA-OneVision-7B at 10% retention. Left: Time-To-First-Token (TTFT) breakdown across vision tower, other pre-LLM operations, and the LLM backbone (lower is better). Right: Decoding throughput in tokens per second (higher is better) and the corresponding average accuracy.

Table 4: End-to-end efficiency analysis of TEMPO on LLaVA-OneVision-7B at two retention budgets, measured on a single NVIDIA H200 GPU. We report prefilling FLOPs, prefilling time, and Time-To-First-Token (TTFT) in milliseconds; speedups are computed against the vanilla baseline.

Method	Retention Ratio R	Prefilling FLOPs (T) ↓	Vision Encoding	Prefilling Time ↓			TTFT ↓	Avg. ↑	
				Compression	LLM Forward	Total		Score	Rel. Acc (%)
Vanilla	100%	40.8	312	–	1372	1372 (1.0×)	1684 (1.0×)	58.4	100.0
TeMPO (Ours)	25%	8.7	312	80	499	579 (2.37×)	891 (**1.89**×)	**59.0**	**101.0**
TeMPO (Ours)	10%	3.4	312	80	353	433 (**3.17**×)	**745** (**2.26**×)	58.3	99.8

Among the compared methods, TEMPO delivers the largest gain, surpassing FastVID by 2.8 points at 10× frames and reaching 109.3% relative accuracy with respect to the 16-frame vanilla baseline.

4.3 Efficiency Analysis

Fig. 4 compares TEMPO with seven training-free methods on LLaVA-OneVision-7B at 10% retention using a single H200 GPU. Inner-LLM methods (FastV and PDrop) incur substantial LLM-backbone costs, whereas pre-LLM compressors substantially reduce this component. TEMPO introduces an 80 ms compression overhead, which is more than offset by the reduced LLM-forward cost. It achieves the lowest TTFT of 745 ms, corresponding to a 2.26× speedup over the vanilla model. Decoding throughput reaches 38.9 tokens/s (1.32× vanilla), the highest among the evaluated methods, while accuracy drops by only 0.1 points.

Tab. 4 further reports end-to-end timing at two operating points. At 25% retention, TEMPO achieves a 1.89× TTFT speedup while improving average accuracy by 0.6 points over the vanilla model. At 10% retention, it further improves TTFT and prefilling speedups to 2.26× and 3.17×, respectively, while preserving 99.8% of vanilla performance.

4.4 Ablation Studies

We conduct ablation studies on LLaVA-OneVision-7B with the token retention budget set to 10% to quantify the contribution of each proposed component.

Table 5: Ablation study on TEMPO components. **RE**, **CSM**, **TW**, and **NF** denote the Residual-Energy backbone, Committed-Support Memory, Text Warm-start, and Neighborhood Fusion, respectively. T-SIoU is the Temporal Support IoU between adjacent frames; higher T-SIoU indicates more committed-support repetition across frames (lower is better).

Method	Prefilling FLOPs (T) ↓	FLOPs Ratio ↓	Before LLM Retained Ratio	T-SIoU ↓	MVBench ↑	EgoSchema ↑	LongVideo Bench ↑	VideoMME ↑	Avg. ↑
Vanilla	40.8	100%	100%	–	58.3	60.4	56.4	58.6	58.4
RE only	3.4	8.3%	10%	0.66	56.8	59.5	54.0	56.0	56.6
RE + CSM	3.4	8.3%	10%	0.31	58.1	60.1	56.0	57.4	57.9
RE + CSM + TW	3.4	8.3%	10%	0.30	58.3	60.2	56.3	57.7	58.1
TeMPO (Full)	3.4	8.3%	10%	**0.29**	**58.5**	**60.3**	**56.6**	**57.9**	**58.3**
TeMPO w/o CSM	3.4	8.3%	10%	0.65	57.0	59.7	54.4	56.4	56.9
TeMPO w/o NF	3.4	8.3%	10%	0.30	58.3	60.2	56.3	57.7	58.1

Table 6: Ablation studies on three core design choices of TEMPO at 10% retention. Top: fusion strategy for handling discarded tokens after the support is fixed. Middle: aggregation function over committed-support memory slots in the novelty score (Eq. 6). Bottom: storage space of committed-support memory.

Variant	Prefilling FLOPs (T) ↓	FLOPs Ratio ↓	Before LLM Retained Ratio	MVBench ↑	EgoSchema ↑	LongVideo Bench ↑	VideoMME ↑	Avg. ↑
Fusion strategy for discarded tokens								
Drop only	3.4	8.3%	10%	58.3	60.2	56.3	57.7	58.1
Cosine merging	3.4	8.3%	10%	58.4	60.2	56.4	57.8	58.2
Neighborhood Fusion (Ours)	3.4	8.3%	10%	**58.5**	**60.3**	**56.6**	**57.9**	**58.3**
Novelty aggregation function over memory slots								
Mean over slots	3.4	8.3%	10%	57.6	59.9	55.5	57.0	57.5
Min over slots	3.4	8.3%	10%	57.2	59.8	55.0	56.7	57.2
Max over slots (Ours)	3.4	8.3%	10%	**58.5**	**60.3**	**56.6**	**57.9**	**58.3**
Memory storage space								
Feature space X_{t-1}	3.4	8.3%	10%	57.8	60.0	55.7	57.0	57.6
Metric space Z_{t-1} (Ours)	3.4	8.3%	10%	**58.5**	**60.3**	**56.6**	**57.9**	**58.3**

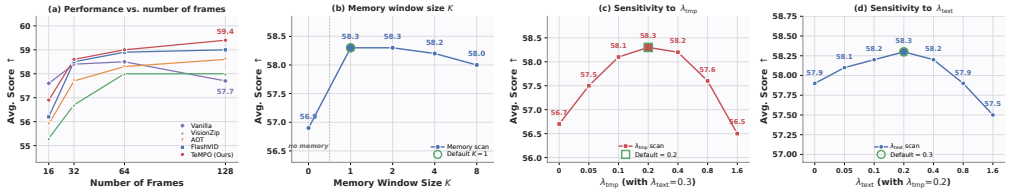


Figure 5: Scaling and sensitivity analysis on LLaVA-OneVision-7B. **(a)** Performance vs. number of input frames at 15% retention. **(b)** Memory window size K at 10% retention; $K = 0$ disables memory. **(c, d)** Sensitivity to the weighting coefficients λ_{tmp} and λ_{text} at 10% retention.

Effect of TEMPO components. TEMPO is built from four components: residual-energy backbone (RE), committed-support memory (CSM), text warm-start (TW), and neighborhood fusion (NF). As shown in Tab. 5, adding CSM raises the average score by 1.3 points and reduces T-SIoU from 0.66 to 0.31, directly addressing Selector Collapse. Removing CSM increases T-SIoU to 0.65 and reduces the score by 1.4 points. Removing NF leaves T-SIoU almost unchanged but lowers the score by 0.2 points, consistent with its role in restoring visual coverage rather than reshaping the selected support.

Core design choices. Tab. 6 validates three design choices of TEMPO. Neighborhood fusion outperforms drop-only and cosine merging by 0.2 and 0.1 points, respectively, by separating metric-space routing from feature-space aggregation. For memory aggregation, replacing $\max$ with mean or min reduces the score by 0.8 and 1.1 points, confirming that a candidate should be treated as repeated once it matches any committed slot. Storing memory in metric space rather than feature space yields a 0.7 point gain, aligning the comparison space with the novelty signal.

Scaling and sensitivity analysis. We analyze TEMPO along three axes in Fig. 5. **(a)** TEMPO consistently outperforms competing compressors across all frame counts, exceeding vanilla by 1.7

points at 128 frames, where vanilla degrades under context-length saturation. **(b)** Increasing the memory window from $K{=}0$ to $K{=}1$ gives the largest gain (1.4 points), confirming that even a single-frame causal record can reduce Selector Collapse, while $K{\geq}4$ slightly degrades performance. **(c, d)** The chosen defaults $\lambda_{\text{tmp}} = 0.2$ and $\lambda_{\text{text}} = 0.3$ fall in the high-performance region. Even with the text branch removed (RE+CSM only), TEMPO preserves more than 99% of vanilla performance.

5 Conclusion

We introduced TEMPO, a training-free framework for frame-causal video token compression. Our analysis identifies Selector Collapse as a common failure mode of existing compressors: different selection rules can repeatedly retain overlapping support across neighboring frames, wasting limited token budgets on evidence already delivered to the language model. At 10% retention, TEMPO preserves 99.8% of vanilla performance on LLaVA-OneVision without retraining or an additional token budget.

References

[1] Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. DivPrune: Diversity-based visual token pruning for large multimodal models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9392–9401, 2025.

[2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[3] 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*, pages 19–35. Springer, 2024.

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

[5] 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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24108–24118. IEEE, 2025.

[6] Yuhang Han, Xuyang Liu, Zihan Zhang, Pengxiang Ding, Junjie Chen, Honggang Chen, Donglin Wang, Qingsen Yan, and Siteng Huang. Filter, correlate, compress: Training-free token reduction for mllm acceleration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 4601–4609, 2026.

[7] 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*, pages 19959–19973, 2025.

[8] Bo Li, Peiyuan Zhang, Kaichen Zhang, Fanyi Pu, Xinrun Du, Yuhao Dong, Haotian Liu, Yuanhan Zhang, Ge Zhang, Chunyuan Li, and Ziwei Liu. LMMs-Eval: Accelerating the development of large multimodal models, March 2024. URL `https://github.com/EvolvingLMMs-Lab/lmms-eval`.

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

[10] Jinlong Li, Liyuan Jiang, Haonan Zhang, and Nicu Sebe. Token reduction via local and global contexts optimization for efficient video large language models. *arXiv preprint arXiv:2603.01400*, 2026.

[11] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. MVBench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Li_MVBench_A_Comprehensive_Multi-modal_Video_Understanding_Benchmark_CVPR_2024_paper.html.

[12] Xu Li, Yi Zheng, Yuxuan Liang, Zhe Liu, Xiaolei Chen, Haotian Chen, Rui Zhu, and Xiangyang Xue. ResPrune: Text-conditioned subspace reconstruction for visual token pruning in large vision-language models, 2026. URL https://arxiv.org/abs/2603.21105.

[13] Jizhihui Liu, Feiyi Du, Guangdao Zhu, Niu Lian, Jun Li, and Bin Chen. HiPrune: Training-free visual token pruning via hierarchical attention in vision-language models. *arXiv preprint arXiv:2508.00553*, 2025.

[14] Xuyang Liu, Yiyu Wang, Junpeng Ma, and Linfeng Zhang. Video compression commander: Plug-and-play inference acceleration for video large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 1910–1924, 2025.

[15] Junpeng Ma, Qizhe Zhang, Ming Lu, Zhibin Wang, Qiang Zhou, Jun Song, and Shanghang Zhang. MMG-Vid: Maximizing marginal gains at segment-level and token-level for efficient video llms, 2025. URL https://arxiv.org/abs/2508.21044.

[16] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. EgoSchema: A diagnostic benchmark for very long-form video language understanding, 2023. URL https://arxiv.org/abs/2308.09126.

[17] Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. LLaVA-PruMerge: Adaptive token reduction for efficient large multimodal models. In *ICCV*, 2025.

[18] 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.

[19] Leqi Shen, Guoqiang Gong, Tao He, Yifeng Zhang, Pengzhang Liu, Sicheng Zhao, and guiguang ding. Fastvid: Dynamic density pruning for fast video large language models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 123553–123581. Curran Associates, Inc., 2025. doi: 10.52202/085713-4118. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/b2e63e36c57e153b9015fece2352a9f9-Paper-Conference.pdf.

[20] Chong Tang, Sannara Ek, Dirk Koch, Robert D. Mullins, Alex S. Weddell, and Jagmohan Chauhan. SURGE: Surprise-guided token reduction for efficient video understanding with vlms. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=rCFPSIjqMf.

[21] Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. DyCoke: Dynamic compression of tokens for fast video large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18992–19001, 2025.

[22] Yiyu Wang, Xuyang Liu, Xiyan Gui, Xinying Lin, Boxue Yang, Chenfei Liao, Tailai Chen, and Linfeng Zhang. Accelerating streaming video large language models via hierarchical token compression. *arXiv preprint arXiv:2512.00891*, 2025.

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

[24] Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. PyramidDrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *CoRR*, abs/2410.17247, 2024. URL https://doi.org/10.48550/arXiv.2410.17247.

[25] 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. *arXiv preprint arXiv:2412.04467*, 2024.

[26] Xiangyu Zeng, Kefan Qiu, Qingyu Zhang, Xinhao Li, Jing Wang, Jiaxin Li, Ziang Yan, Kun Tian, Meng Tian, Xinhai Zhao, Yi Wang, and Limin Wang. StreamForest: Efficient online video understanding with persistent event memory, 2025. URL https://arxiv.org/abs/2509.24871.

[27] Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. LMMs-Eval: Reality check on the evaluation of large multimodal models, 2024. URL https://arxiv.org/abs/2407.12772.

[28] Qizhe Zhang, Aosong Cheng, Ming Lu, Renrui Zhang, Zhiyong Zhuo, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. Beyond text-visual attention: Exploiting visual cues for effective token pruning in vlms. *arXiv preprint arXiv:2412.01818*, 2025.

[29] Qizhe Zhang, Mengzhen Liu, Lichen Li, Ming Lu, Yuan Zhang, Junwen Pan, Qi She, and Shanghang Zhang. Beyond attention or similarity: Maximizing conditional diversity for token pruning in mllms. *arXiv preprint arXiv:2506.10967*, 2025.

[30] Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. SparseVLM: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*, 2024.

[31] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. LLaVA-Video: Video instruction tuning with synthetic data, 2025. URL https://arxiv.org/abs/2410.02713.

[32] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

Funding and Competing Interests

The authors received no third-party funding or support for this work during the 36 months prior to submission. The authors declare no relevant competing financial interests outside their stated primary affiliations during this period.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 clearly state our contributions and scope regarding the TEMPO framework and its performance in video token compression.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [No]

   Justification: The paper does not include a dedicated limitations discussion.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [No]

Justification: Formal proofs for the residual-energy reduction are not included in this version.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [No]

   Justification: Section 4 describes the experimental setup and hyperparameters, but further implementation details are not included in this version.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Code: `https://github.com/laiyingxin2/TeMPO`. Project page: `https://laiyingxin2.github.io/TeMPO/`.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: The evaluation settings, including retention ratios and hyperparameters (λ_{tmp}, λ_{text}), are described in Section 4.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We report average performance across multiple distinct benchmarks and VLLM backbones to ensure reliability, though explicit error bars for each individual run are omitted due to the large computational scale of VLLM inference.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Section 4.3 reports the GPU and inference costs, but does not specify memory use and execution time for each experiment.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work complies fully with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: The paper does not include a separate discussion of societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our work introduces a training-free compression module and does not release new high-risk foundational models or sensitive datasets.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: Models and benchmarks are cited in Section 4; license terms are not documented in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper does not introduce new datasets or foundational models.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: No crowdsourcing or human subjects were involved; all results are based on computational benchmarks.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The pretrained video-language backbones used in the experiments are described in Section 4.1.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.