GTR: Guide-Then-Refine Token Compression for Training-Free Acceleration of Video-LLMs

Sijin Zhou[1,3⋆], Wei Feng[1,3⋆], Zhuang Qi[4], Zhonghua Wang[2,3], Lie Ju[1,3], Xiang An[5], Mehrtash Harandi[1], and Zongyuan Ge[2,3†]

[1] Department of Electrical and Computer Systems Engineering, Monash University, Australia
[2] Faculty of Information Technology, Monash University, Australia
[3] Airdoc-Monash Research, Monash University, Australia
[4] School of Software, Shandong University, China
[5] Beijing DeepGlint Technology Co., Ltd, China

Abstract. Video large language models (Video-LLMs) have achieved remarkable progress in video understanding tasks, but their heavy computational overhead during inference severely limits deployment in resource-constrained scenarios. This highlights the urgent need for efficient inference acceleration frameworks tailored to Video-LLMs. Existing acceleration methods typically apply a homogeneous processing approach to all video frames, disregarding the fact that each frame within a video clip conveys distinct visual information. Additionally, many of them underutilize text queries, or add text guidance only through costly pre-training or intrusive LLM modifications, underexploiting the text-video alignment that is critical in long-form video understanding where redundant visual content is common. To address this limitation, we propose Guide-Then-Refine (GTR), a plug-and-play inference acceleration framework for Video-LLMs that realizes token compression through a two-stage mechanism. First, for all tokens in each video frame, we compute both global scores and text-aware scores. Based on the sum of these scores for all tokens in a frame, we dynamically adjust the token retention ratio for each frame. Subsequently, according to the retention ratio, we calculate the local score of each token to evaluate its importance within the current frame, thereby selecting critical visual tokens. Extensive evaluations on mainstream video benchmarks and various Video-LLMs demonstrate the effectiveness of our approach. With only 15% of visual tokens retained, our method maintains an average of 95.8% of the original performance across four benchmarks.

Keywords: Model Compression · Efficient AI · Video-LLMs

1 Introduction

Video-Large Language Models (Video-LLMs) have emerged as a cornerstone of modern multimodal AI, enabling advanced capabilities in video question answer-

⋆ Equal contribution. † Corresponding author: zongyuan.ge@monash.edu.

ing (Video QA), action recognition, real-time surveillance, and autonomous navigation [3, 9, 18, 44]. From early extensions of image-language models (e.g., VideoLLaVA) to state-of-the-art long-video specialists (e.g., FiLA-Video, CrossLMM), Video-LLMs have achieved remarkable performance by modeling spatiotemporal dynamics of visual content [13, 36]. However, the exponential growth of visual tokens generated by high-resolution videos poses prohibitive computational and memory burdens. The quadratic complexity of the Transformer's attention mechanism amplifies this issue, resulting in an excessively long inference delay for long videos [5, 25, 30, 37].

An analysis of state-of-the-art token compression methods reveals three core limitations that become more pronounced in long-video scenarios. First, many methods underuse text guidance during token evaluation. A large family of approaches, such as STTM, DynTok, and FLOC, rely solely on visual cues (e.g., inter-frame similarity and spatial density) for token pruning [8, 16, 41]. In long-video question answering (QA), this visual-only paradigm can discard tokens that seem unimportant globally but are essential for the query, leading to performance drops on benchmarks such as MLVU and LongVideoBench [33, 45]. While a growing number of methods do incorporate text or instruction signals (e.g., DyCoke, DyToK, and HoliTom [21, 30, 31]), they typically do so at extra cost—expensive pre-training (e.g., HICom [26]), reliance on LLM-guided signals, or in-LLM modifications that hurt deployment. The key gap is thus not the absence of text guidance, but the lack of a lightweight, plug-and-play mechanism that injects query relevance before the LLM.

Beyond the lack of text guidance, existing methods also struggle with dynamic video content and deployment compatibility. Static compression strategies are often misaligned with video dynamics: most frameworks, such as STTM with fixed quadtree granularity and MMG-Vid with segment-level thresholding, apply uniform compression ratios across frames [16, 27], ignoring large variations in information density over time. For instance, transition frames with static backgrounds may contain many redundant tokens, while action-intensive frames with fine-grained motion require higher token retention. Although VidCom^2 mitigates this issue via frame uniqueness quantification, it still lacks integration with text signals [25]. Besides, many methods remain poorly compatible with mainstream architectures. Approaches like CrossLMM modify LLM backbones with dual cross-attention layers [36], and HoliTom requires changes to both pre-LLM token merging and in-LLM attention filtering [30]. These modifications increase deployment complexity and hinder seamless integration with popular models (e.g., Qwen2-VL and LLaVA-OneVision) without retraining [3, 18].

To address the aforementioned limitations in existing token compression methods for Video-LLMs, we propose Guide-Then-Refine (GTR), a two-stage visual token compression framework for Video-LLMs, especially in long-video scenarios. The GTR framework first computes global scores and text scores for all tokens across video frames, then dynamically adjusts the token retention ratio for each frame based on the sum of scores of all tokens within that frame. For global score calculation, GTR first calculates the average value of visual tokens

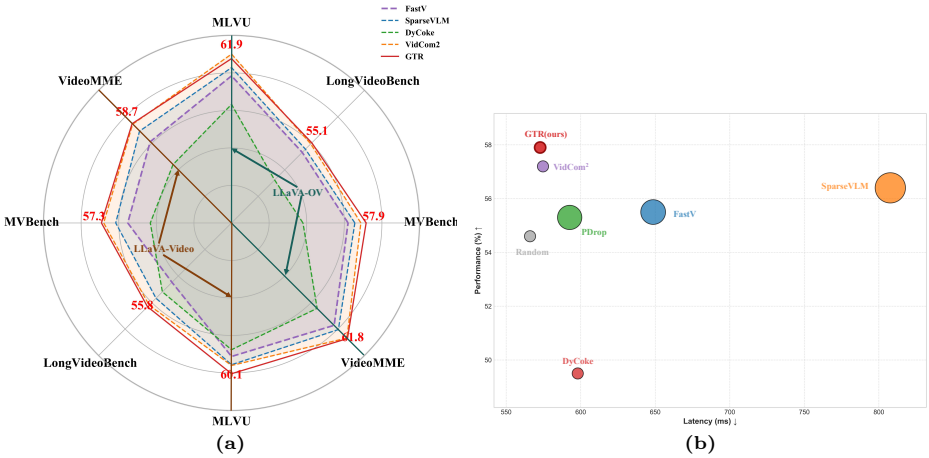


Fig. 1: Overall comparison of GTR in efficiency and performance. (a) Performance retention compared with baseline methods. (b) Inference efficiency improvement with token compression, where circle size denotes GPU memory usage.

at the same spatial position across all frames. It then computes the cosine similarity between each individual token and the corresponding position-averaged token, which reflects the importance of the token within the entire video context. For text score calculation, GTR computes the cosine similarity between each token and the text embedding of the input text instruction. The text embedding is generated via a text encoder, which helps GTR capture the relevance of the token to the text instruction. After determining the dynamic token retention ratio for each frame, GTR proceeds to the second stage by computing local scores for tokens within each frame. The local score of a token is defined as the average value of cosine similarities between the token and its adjacent tokens, which is used to evaluate the importance of the token within the local context of the current frame. This two-stage design of GTR effectively integrates text guidance into token evaluation, adapts compression strategies to the dynamic information density of video frames, and avoids modifications to the backbone architectures of mainstream Video-LLMs.

Extensive evaluations are conducted on four video understanding benchmarks [12, 20, 33, 45] and two mainstream Video-LLMs [18, 44]. As depicted in Fig. 1, GTR consistently improves inference efficiency while preserving strong task performance under aggressive token compression. In particular, it achieves a better accuracy–efficiency trade-off than prior training-free baselines, demonstrating robust generalization across both models and benchmarks.

2 Related Work

2.1 Video Large Language Models

Video-LLMs are built upon the evolution of multimodal LLMs that align vision and language representations. Early systems such as Flamingo and BLIP-2 established strong cross-modal interaction via gated cross-attention and Q-Former-style connectors [1,19], while the LLaVA line simplified adaptation with lightweight projectors and efficient instruction tuning [2,18,22–24]. This paradigm enabled scalable transfer from image-language models to video understanding by treating videos as frame/token sequences.

Recent Video-LLMs further focus on long-context and temporal reasoning. Representative models introduce stronger spatiotemporal modeling, synthetic video instruction data, and long-context mechanisms, e.g., VideoLLaMA2, LLaVA-Video, Qwen2-VL, LongVA, and LongVILA [6, 7, 32, 42, 44]. In practice, most frameworks follow a common pipeline: visual encoding (e.g., CLIP/SigLIP), projection into the LLM token space, and instruction tuning for downstream tasks such as QA and temporal reasoning [28, 29, 39, 45]. Although this design greatly improves capability, it also increases visual token volume in long videos, motivating efficient token compression methods.

2.2 Token Compression for Video-LLMs

Token compression has become essential for Video-LLMs because long videos produce massive numbers of visual tokens and incur prohibitive attention cost. Existing approaches are mainly divided into unconditional compression and conditional compression. Unconditional methods (e.g., pooling/merging and clustering pipelines) reduce redundancy from visual statistics alone [4, 15, 17, 28, 31, 35], while additional convolution or memory designs further improve temporal efficiency [7, 14, 40]. These methods are generally efficient, but they may remove tokens that are weak in visual saliency yet crucial for answering a specific query.

Conditional methods introduce text or instruction signals to alleviate this issue. Representative works such as HICom, FiLA-Video, VidCom^2, and CrossLMM improve semantic alignment through instruction-aware scoring, adaptive frame selection, or cross-modal interaction [13, 25, 26, 36]. Training-free plug-and-play approaches, including HoliTom and STTM, also provide practical acceleration without full retraining [16, 30]. More recent methods explore text- and frame-adaptive compression, e.g., DyToK with an LLM-guided keyframe prior [21] and HoliTom with a hybrid pre-LLM/in-LLM design [30]. However, such LLM-guided or in-LLM operations raise integration complexity and reduce compatibility with optimized kernels. In contrast, GTR compresses *entirely before the LLM*, leaving its layers, attention, and KV-cache untouched for plug-and-play deployment. Nevertheless, existing methods still struggle to jointly satisfy three goals: strong text-aware relevance, frame-adaptive compression for long videos, and seamless compatibility with efficient operators (e.g., FlashAttention [10]).

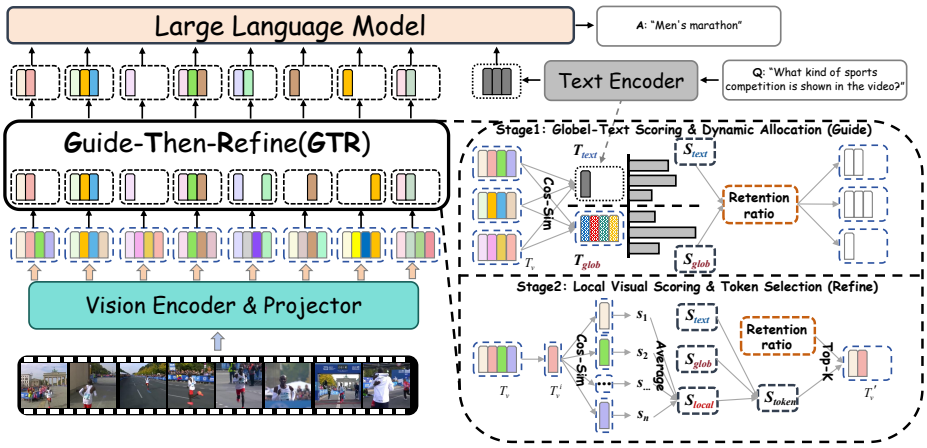


Fig. 2: Overview of the proposed Guide-Then-Refine (GTR) framework for training-free token compression in Video-LLMs. Given video frames and a text query, Stage 1 (**Guide**) computes global uniqueness and text-aware relevance scores to estimate frame-level informativeness and allocate dynamic retention budgets across frames. Stage 2 (**Refine**) then performs local structure-aware scoring within each frame and preserves the most critical visual tokens under the assigned budget.

3 Methodology

Fig. 2 illustrates the overall architecture of Guide-Then-Refine (GTR). The framework operates through a dual-stage workflow comprising global-text scoring (Guide) and local visual scoring (Refine), which are detailed in the following sections. This framework is plug-and-play: it integrates into existing pipelines without changing model weights or retraining, optimizing token density through hierarchical selection while preserving performance.

3.1 Global-Text Scoring & Dynamic Allocation

This stage computes initial visual token importance by fusing cross-frame uniqueness and text relevance, then adaptively allocates retention budgets per frame. This ensures information-rich frames (e.g., those matching text instructions) retain more tokens while redundant ones are compressed. The input comprises I video frames, each encoded into N spatial-semantic tokens via a pre-trained visual encoder (e.g., CLIP-ViT).

Global Visual Scoring We denote $\mathcal{F}_{i,j} \in \mathbb{R}^D$ as the feature token corresponding to the j-th spatial patch in the i-th frame, where D is the feature dimension determined by the pre-trained visual encoder. The global visual score $S_{\text{glob}}(i,j)$ quantifies how unique the token $\mathcal{F}_{i,j}$ is across the entire video; tokens that are highly similar to their counterparts in other frames are considered redundant and assigned lower scores.

To compute $S_{\text{glob}}(i,j)$, we first calculate the cross-frame average feature of the j-th token position across all I frames as $\bar{\mathcal{F}}_j = \frac{1}{I}\sum_{i=1}^{I}\mathcal{F}_{i,j}$. This average token $\bar{\mathcal{F}}_j$ serves as a reference for the typical feature of that spatial position in the video, capturing the consistent information (e.g., static background) across frames.

Next, we measure the similarity between $\mathcal{F}_{i,j}$ and $\bar{\mathcal{F}}_j$ using cosine similarity, which quantifies the angular distance between two high-dimensional vectors. The cosine similarity between $\mathcal{F}_{i,j}$ and $\bar{\mathcal{F}}_j$ is computed as:

$$\cos\left(\mathcal{F}_{i,j}, \bar{\mathcal{F}}_j\right) = \frac{\mathcal{F}_{i,j} \cdot \bar{\mathcal{F}}_j}{\|\mathcal{F}_{i,j}\|_2 \cdot \|\bar{\mathcal{F}}_j\|_2} \tag{1}$$

where $\cdot$ denotes the dot product, and $\|\cdot\|_2$ denotes the L2 norm of the vector. A higher cosine similarity indicates that $\mathcal{F}_{i,j}$ is more consistent with the cross-frame average, thus carrying less unique information. The global visual score is defined as the complement of this similarity to map higher uniqueness to higher scores:

$$S_{\text{glob}}(i,j) = 1 - \cos(\mathcal{F}_{i,j}, \bar{\mathcal{F}}_j) \tag{2}$$

Dynamic regions generate high $S_{\text{glob}}(i,j)$ scores as $\mathcal{F}_{i,j}$ diverges from the background-inclusive average $\bar{\mathcal{F}}_j$, whereas static regions yield low scores due to feature stability across frames.

Text Relevance Scoring The text relevance score $S_{\text{text}}(i,j)$ measures how closely the visual information encoded in $\mathcal{F}_{i,j}$ aligns with the input text instruction, ensuring that tokens relevant to the task are prioritized. To maintain consistency in the embedding space, the text encoder is selected to match the pre-trained visual encoder. In practice we instantiate it with SigLIP, which is naturally matched for SigLIP-based towers (e.g., LLaVA-OneVision) and acts as an external aligned encoder for models without a paired text encoder (e.g., Qwen2-VL). Since the text score is only one of three fused components, with the global and local scores being query-agnostic, GTR does not require exact text-vision alignment.

Let $Q = \{q_1, q_2, \ldots, q_M\}$ be the input text instruction. We encode Q using the matched text encoder to obtain a single aggregated text embedding $T \in \mathbb{R}^D$, which integrates the semantic information of the entire instruction into a high-dimensional vector. This aggregation avoids noise from individual text tokens and ensures alignment with the overall task objective. The text relevance score is computed as the cosine similarity between $\mathcal{F}_{i,j}$ and T, with the cosine similarity calculated as:

$$S_{\text{text}}(i,j) = \cos\left(\mathcal{F}_{i,j}, T\right) = \frac{\mathcal{F}_{i,j} \cdot T}{\|\mathcal{F}_{i,j}\|_2 \cdot \|T\|_2} \tag{3}$$

This score quantifies the semantic alignment between local visual patches $\mathcal{F}_{i,j}$ and the text instruction T. Consequently, regions semantically consistent with the query yield high $S_{\text{text}}(i,j)$ values, while irrelevant areas produce low scores.

Dynamic Frame-Level Budget Allocation After computing the global visual score $S_{\text{glob}}(i,j)$ and text relevance score $S_{\text{text}}(i,j)$ for each token, we derive the comprehensive score $S_{\text{guide}}(i,j)$ as $S_{\text{guide}}(i,j) = \alpha \cdot S_{\text{glob}}(i,j) + \beta \cdot S_{\text{text}}(i,j)$, governed by fixed hyperparameters α and β selected via validation experiments. For all experiments, we fixed the hyperparameters at $\alpha = 0.375$ and $\beta = 0.625$, which yielded the best performance on the validation set.

To allocate the token retention budget across frames, we compute a frame-level significance score $S_{\text{guide}}(i)$ by aggregating the comprehensive token scores within frame i as $S_{\text{guide}}(i) = \sum_{j=1}^{N} S_{\text{guide}}(i,j)$. The total score $S_{\text{guide}}(i)$ serves as a proxy for the frame's information value, as frames with higher total scores contain more task-relevant and unique tokens and thus require a higher retention ratio. We then normalize $S_{\text{guide}}(i)$ across all frames to compute the dynamic retention ratio r_i for frame i:

$$r_i = r_{\text{base}} + (r_{\max} - r_{\text{base}}) \times \frac{S_{\text{guide}}(i) - S_{\min}}{S_{\max} - S_{\min}} \tag{4}$$

where $S_{\min} = \min_{i \in [1,I]} S_{\text{guide}}(i)$ and $S_{\max} = \max_{i \in [1,I]} S_{\text{guide}}(i)$ are the minimum and maximum total scores across all frames, respectively. r_{base} (set to 10%) and $r_{\max}$ (set to 40%) are pre-defined hyperparameters that enforce a minimum token retention to avoid complete information loss in low-value frames and a maximum retention to ensure compression efficiency. This dynamic allocation prioritizes key frames with rich semantic content while aggressively compressing redundant ones, effectively balancing compression efficiency and information preservation.

3.2 Local Visual Scoring & Token Selection

Leveraging the frame-level budgets from Stage 1, this stage refines token selection by integrating local spatial context. This addresses the limitation of purely global scoring, which often discards locally critical tokens (e.g., small objects) due to low global uniqueness. By incorporating fine-grained spatial information, our method preserves essential details for task accuracy, effectively complementing the global, text-driven selection of Stage 1. The input to this stage includes the original visual tokens $\mathcal{F}_{i,j} \in \mathbb{R}^d$ (for $i \in [1,I]$ frames and $j \in [1,N]$ tokens per frame), the global visual scores $S_{\text{glob}}(i,j)$, text relevance scores $S_{\text{text}}(i,j)$, and dynamic retention ratios r_i for each frame.

Local Visual Scoring The local visual score $S_{\text{local}}(i,j)$ evaluates the importance of token $\mathcal{F}_{i,j}$ within the spatial context of its current frame. Tokens that are distinct from their neighbors often correspond to foreground objects or fine-grained details (e.g., edges, small tools), whereas tokens highly similar to their surroundings typically belong to homogeneous background regions (e.g., solid-color walls) that contribute minimally to task performance.

To capture this spatial context, we define a *local spatial neighborhood* $\mathcal{N}_{i,j}$ for each token based on its grid position. Specifically, let the token $\mathcal{F}_{i,j}$ be located at grid coordinates (u,v) within a frame of height H and width W. The

neighborhood $\mathcal{N}_{i,j}$ is defined as the set of tokens at positions (u', v') satisfying the condition of immediate spatial adjacency:

$$\mathcal{N}_{i,j} = \{\mathcal{F}_{i,k} \mid \max(|u - u'|, |v - v'|) = 1,\ 1 \le u' \le H,\ 1 \le v' \le W\}, \tag{5}$$

where k denotes the linearized index corresponding to the grid position (u', v'). This formulation utilizes the Chebyshev distance to identify all valid adjacent tokens within the immediate vicinity. Crucially, by restricting the set to the valid grid bounds $[1, H] \times [1, W]$, the definition automatically adapts to boundary conditions: tokens located at edges or corners naturally possess a smaller neighborhood cardinality without requiring artificial padding or explicit case handling.

Notably, this neighborhood relies solely on the intrinsic 2D grid topology of the feature map, making it **agnostic to the positional encoding scheme** (e.g., learnable embeddings or RoPE) used in subsequent layers, so that grid-based spatial adjacency remains a robust indicator of local semantic coherence.

The local score is computed as the average cosine similarity between $\mathcal{F}_{i,j}$ and each of its valid neighbors in $\mathcal{N}_{i,j}$. For any neighbor $\mathcal{F}_{i,k} \in \mathcal{N}_{i,j}$, the cosine similarity is calculated as:

$$\mathrm{sim}(\mathcal{F}_{i,j}, \mathcal{F}_{i,k}) = \frac{\mathcal{F}_{i,j} \cdot \mathcal{F}_{i,k}}{\|\mathcal{F}_{i,j}\|_2 \cdot \|\mathcal{F}_{i,k}\|_2}. \tag{6}$$

Consequently, the local visual score $S_{\text{local}}(i, j)$ is defined as the mean similarity over the adaptive neighborhood:

$$S_{\text{local}}(i, j) = 1 - \frac{1}{|\mathcal{N}_{i,j}|} \sum_{\mathcal{F}_{i,k} \in \mathcal{N}_{i,j}} \mathrm{sim}(\mathcal{F}_{i,j}, \mathcal{F}_{i,k}), \tag{7}$$

where $|\mathcal{N}_{i,j}|$ denotes the cardinality of the neighborhood set, which varies dynamically depending on the token's spatial position. A higher $S_{\text{local}}(i, j)$ indicates higher local distinctiveness, thereby assigning greater importance to the token for downstream tasks.

Multi-Score Fusion and Frame-wise Pruning The final step integrates the global, text, and local scores to select the most critical tokens for each frame. We compute the final importance score $S_{\text{final}}(i, j)$ for each token by linearly combining the three metrics:

$$S_{\text{final}}(i, j) = \lambda_g S_{\text{glob}}(i, j) + \lambda_t S_{\text{text}}(i, j) + \lambda_l S_{\text{local}}(i, j), \tag{8}$$

where $\lambda_g, \lambda_t, \lambda_l$ are balancing coefficients. In our experiments, we set $\lambda_g = 0.3$, $\lambda_t = 0.5$, and $\lambda_l = 0.2$. This configuration prioritizes **text alignment** ($\lambda_t = 0.5$), consistent with the design philosophy in Stage 1 where text relevance was dominant ($\beta = 0.625$), ensuring that tokens semantically relevant to the instruction are preserved. The **global uniqueness** term ($\lambda_g = 0.3$) maintains cross-frame key information, while the **local context** term ($\lambda_l = 0.2$) retains fine-grained

spatial details that might be overlooked by global or text-based scoring alone. These weights were empirically determined to balance inference efficiency and task accuracy across diverse benchmarks.

For each frame i, we sort all N tokens in descending order of $S_{\text{final}}(i,j)$ and retain the top K_i tokens, where the budget K_i is determined by the dynamic ratio r_i computed in Eq. 4 (Stage 1) as $K_i = \lceil N \cdot r_i \rceil$. This strategy ensures that the compression rate is tailored to each frame's content: key frames with high information density preserve more tokens, while redundant frames undergo aggressive pruning.

After selection, we concatenate the retained tokens across frames in temporal order to form the final visual sequence $\mathcal{F}_{\text{retained}} = [\mathcal{F}_1^*, \mathcal{F}_2^*, \ldots, \mathcal{F}_I^*]$, where $\mathcal{F}_i^* \subset \{\mathcal{F}_{i,j}\}_{j=1}^N$ denotes the subset of selected tokens for frame i with $|\mathcal{F}_i^*| = K_i$. The resulting sequence $\mathcal{F}_{\text{retained}}$ effectively preserves task-relevant information from three complementary perspectives: global cross-frame uniqueness, text instruction alignment, and local spatial details.

4 Experiments

In this section, we comprehensively evaluate the effectiveness and efficiency of our proposed Guide-Then-Refine (GTR) framework. We conduct extensive experiments on multiple video understanding benchmarks, comparing GTR against state-of-the-art token pruning baselines. Furthermore, we perform ablation studies to validate the contribution of each component, analyze the compatibility with different Video-LLM architectures, and provide qualitative visualizations to interpret the token selection behavior.

4.1 Experimental Setup

We evaluate our method on four challenging video understanding benchmarks covering diverse tasks and video lengths including MVBench [20], MLVU [45], LongVideoBench [33] and VideoMME [12]. We note that recent studies question how faithfully such benchmarks isolate temporal understanding [11, 38]; we therefore evaluate across four diverse benchmarks and complement them with qualitative analyses to reduce single-benchmark bias. Besides, we compare GTR with several representative token pruning and compression methods including Random, FastV [5], SparseVLM [43], PDrop [34], DyCoke [31] and VidCom^2 [25]. We integrate GTR into two popular Video-LLM backbones: LLaVA-OneVision-7B and LLaVA-Video-7B. For the dynamic budget allocation (Stage 1), we set $r_{\text{base}} = 10\%$ and $r_{\max} = 40\%$. In the fusion stage (Stage 2), the weights are fixed at $\lambda_g = 0.3, \lambda_t = 0.5, \lambda_l = 0.2$. We evaluate two compression settings: retaining **25%** and **15%** of the original tokens on average, which is consistent with previous work [25]. All experiments are conducted on NVIDIA A100-SXM4-80GB GPUs. Due to space constraints, we further provide in the supplementary material: 1) details of the evaluation benchmarks, 2) the impact of different hyperparameter settings, and 3) robustness to the number of input frames.

Table 1: Performance comparison with other baselines with LLaVA-OV-7B across different benchmarks.

Method	MVBench	LongVideoBench	MLVU	VideoMME				Avg
				Overall	Short	Medium	Long	(%)
Full Tokens (100%)								
LLaVA-OV-7B	56.9	56.4	63.0	58.6	70.3	56.6	48.8	100.0
Retention Ratio = 25%								
Random	54.2	52.7	59.7	55.6	65.4	53.0	48.3	94.8
FastV [5]	55.5	53.3	59.6	55.3	65.0	53.8	47.0	94.9
SparseVLM [43]	56.4	53.9	60.7	57.3	68.4	55.2	48.1	97.5
PDrop [34]	55.3	51.3	57.1	55.5	64.7	53.1	48.7	94.4
DyCoke [31]	49.5	48.1	55.8	51.0	61.1	48.6	43.2	87.0
VidCom^2 [25]	57.2	54.9	**62.5**	58.6	**69.8**	56.4	49.4	99.6
GTR (Ours)	**57.9**	**55.1**	61.9	**58.7**	69.6	**56.7**	**49.9**	**99.8**
Retention Ratio = 15%								
FastV [5]	51.6	48.3	55.0	48.1	51.4	49.4	43.3	85.0
SparseVLM [43]	52.9	49.7	57.4	53.4	61.0	52.1	47.0	91.2
PDrop [34]	53.2	47.6	54.7	50.1	58.7	48.7	45.0	87.4
VidCom^2 [25]	54.3	52.0	58.9	56.2	**65.8**	54.8	48.1	95.1
GTR (Ours)	**55.7**	**52.6**	**59.3**	**56.4**	65.5	**55.1**	**48.7**	**95.8**

4.2 Main Results

Table 1 and Table 2 present the performance comparison on various benchmarks with LLaVA-OneVision-7B and LLaVA-Video-7B backbones, respectively.

Superior Accuracy at High Compression Rates. As shown in Table 1, with a retention ratio of **25%**, GTR achieves an average score of **57.9%** on MVBench and **55.1%** on LongVideoBench, outperforming the strongest baseline VidCom^2 (57.2% and 54.9%) and significantly surpassing FastV and SparseVLM. Notably, on the challenging VideoMME-Long subset, GTR reaches **49.9%**, demonstrating its ability to preserve critical long-range temporal dependencies.

When the retention ratio is aggressively reduced to **15%**, the advantage of GTR becomes even more pronounced. For MVBench, while other methods suffer from severe performance degradation (e.g., FastV drops to 51.6% on average), GTR maintains a robust average score of **55.7%**, which is comparable to the 25% performance of many baselines. This confirms that our multi-score fusion strategy effectively identifies the most informative tokens even under extreme compression.

Generalization Across Architectures. Table 2 verifies the generalization capability of GTR on the LLaVA-Video-7B backbone. At 25% retention, GTR achieves an average score of **94.0%** (relative to full token performance), outperforming VidCom^2 (93.6%) and SparseVLM (91.6%). Even at 15% retention, GTR retains **89.1%** of the full performance, whereas FastV collapses to 78.0%. These results indicate that GTR's token selection mechanism is architecture-agnostic and consistently effective across different Video-LLM variants.

Table 2: Performance comparison with other baselines with LLaVA-Video-7B across different benchmarks.

Method	MVBench	LongVideoBench	MLVU	VideoMME				Avg
				Overall	Short	Medium	Long	(%)
Full Tokens (100%)								
LLaVA-Video-7B	60.4	59.6	70.3	64.3	77.2	62.1	53.4	100.0
Retention Ratio = 25%								
FastV [5]	53.8	51.2	57.8	59.3	67.1	60.0	50.8	89.7
SparseVLM [43]	55.4	54.2	58.9	60.1	71.1	59.1	50.1	91.6
DyCoke [31]	50.8	53.0	56.9	56.1	65.8	53.6	48.9	86.3
VidCom^2 [25]	57.0	55.5	59.0	61.7	**73.0**	**61.7**	50.0	93.6
GTR (Ours)	**57.3**	**55.8**	**60.1**	**61.8**	72.8	61.5	**51.2**	**94.0**
Retention Ratio = 15%								
FastV [5]	44.0	44.6	53.8	51.3	56.4	51.1	46.2	78.0
SparseVLM [43]	53.1	52.7	56.2	55.7	65.0	53.9	48.3	86.3
VidCom^2 [25]	53.3	51.5	56.8	58.3	**68.0**	57.3	49.7	88.5
GTR (Ours)	**54.1**	**52.1**	**58.2**	**58.5**	67.6	**57.5**	**50.4**	**89.1**

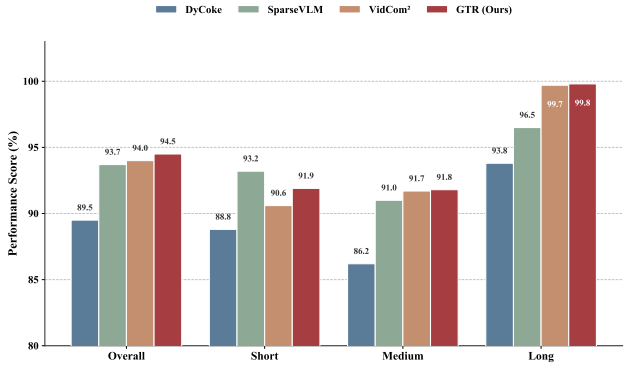


Fig. 3: Comparison on VideoMME with Qwen2-VL at a 25% token retention rate. GTR consistently achieves the best or tied-best results across all evaluated sequence lengths.

Generalization to Qwen2-VL. To verify that GTR is not tied to the LLaVA family, we further evaluate it on Qwen2-VL, whose visual encoder and token interface differ substantially from LLaVA-based backbones. Following the setup of VidCom^2 [25], we compare against representative baselines on VideoMME at 25% retention. As shown in Fig. 3, GTR attains the strongest overall performance (**94.5%**), surpassing DyCoke (89.5%), SparseVLM (93.7%), and VidCom^2 (94.0%), with the largest margin on long videos (**99.8%**). This confirms that GTR generalizes across heterogeneous Video-LLM architectures.

4.3 Efficiency Analysis

Table 3 reports the efficiency–accuracy trade-off of GTR on LLaVA-OV-7B. At 25% retention, GTR reduces total latency from **26min03s** to **18min44s** (about

Table 3: Efficiency comparisons on LLaVA-OV-7B.

Method	Latency (LLM) (ms)	Latency (Model) (ms)	Total Latency (min:s)	GPU Memory (GB)	Throughput (sample/s)	Performance (Avg %)
Full Tokens						
LLaVA-OV-7B	618.0	1008.4	26:03	17.7	0.64	56.9
Retention Ratio = 25%						
Random	178.2	566.0	18:44	16.0	0.89	54.6
FastV [5]	260.9	648.6	20:07	24.7	0.83	55.5
SparseVLM [43]	410.6	807.7	25:03	27.1	0.67	56.4
PDrop [34]	205.6	592.6	18:50	24.5	0.88	55.3
DyCoke [31]	205.2	598.0	18:56	16.1	0.88	49.5
VidCom^2 [25]	180.7	574.7	18:46	**16.0**	0.88	57.2
GTR (Ours)	**179.8**	**572.8**	**18:44**	16.1	**0.88**	**57.9**
Retention Ratio = 15%						
FastV [5]	172.4	599.3	18:19	24.6	0.91	51.6
SparseVLM [43]	370.4	764.8	24:09	27.1	0.69	52.9
PDrop [34]	165.3	552.6	18:32	24.5	0.90	53.2
VidCom^2 [25]	**129.2**	**533.0**	**18:11**	15.8	0.92	54.3
GTR (Ours)	130.8	535.1	18:15	**15.8**	**0.92**	**55.7**

1.4×) while slightly improving average performance (**57.9%** vs. 56.9% for full tokens). This result is consistent with our design: text-guided dynamic budgeting preserves semantically important frames, and local refinement maintains fine-grained evidence within retained frames.

Compared with representative baselines, GTR provides a better speed–quality balance. SparseVLM reaches 56.4% at 25% retention but incurs higher end-to-end latency (25min03s), suggesting that selection overhead can offset compression gains; FastV and PDrop are faster than full-token inference but lose more accuracy. Although GTR adds multi-dimensional scoring and a text encoder, its Guide/Refine overhead is only 8.5/2.3 ms and it remains 75.8 ms faster than FastV, which processes all tokens through the first LLM layers and relies on in-LLM attention incompatible with FlashAttention. As this preprocessing is linear while attention is quadratic in token count, the gap widens for longer videos. VidCom^2 is computationally competitive, yet GTR still attains higher average performance (57.9% vs. 57.2%) at similar throughput, indicating that text-aware guidance is more task-aligned than visual-only heuristics.

An additional finding is the memory advantage of GTR. At 25% retention, FastV, SparseVLM, and PDrop require 24.5–27.1 GB GPU memory, whereas GTR and VidCom^2 remain around 16.0–16.1 GB. This gap is consistent with better compatibility with FlashAttention-style kernels, which reduces activation and attention-memory overhead during decoding. The same trend holds at 15% retention: GTR keeps **55.7%** average performance at practical latency (18min15s), and although VidCom^2 is slightly faster (18min11s), GTR delivers higher accuracy (55.7% vs. 54.3%), yielding a stronger utility–efficiency trade-off.

4.4 Ablation Studies and Compatibility

Impact of Individual Score Components. To validate the contribution of each scoring term in Eq. 8, we conduct ablation experiments by removing one component at a time. As shown in Fig. 4(a)–(b), using any single score alone is insuf-

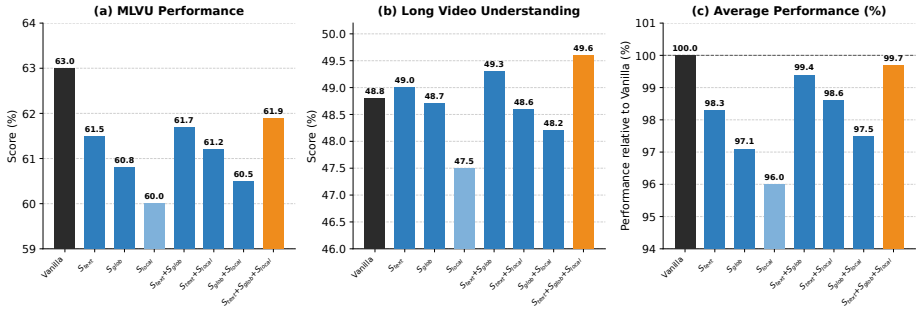


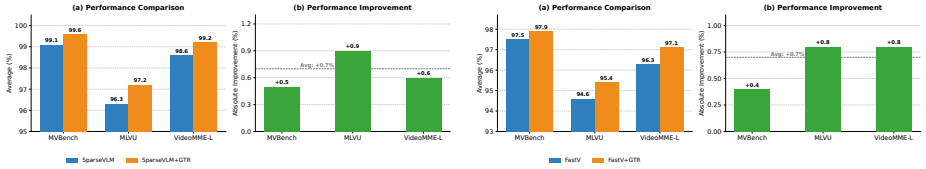
Fig. 4: Ablation results of different scoring components. (a) Absolute performance on MLVU. (b) Absolute performance on long-video understanding. (c) Average performance relative to the vanilla baseline.

ficient to recover the full capability of the model. Among single-term variants, S_{text} is the strongest (61.5% on MLVU and 49.0% on long-video understanding), while S_{local} is the weakest (60.0% and 47.5%). This trend is consistent with our motivation: in long videos, instruction-aware relevance is the primary signal for avoiding semantically incorrect pruning, whereas purely local redundancy cues are not robust enough when temporal context is complex.

Combining scores consistently enhances performance, verifying their complementarity. The pairing of $S_{\text{text}} + S_{\text{glob}}$ achieves the strongest two-term results (61.7% on MLVU; 49.3% on LongVideoBench), demonstrating that text alignment and cross-frame uniqueness form a synergistic core, where the former preserves query-relevant content while the latter suppresses repeated background information. Integrating S_{local} further refines this selection, pushing the full model to peak accuracy (61.9% on MLVU; 49.6% on LongVideoBench), which recovers 99.7% of the vanilla performance (Fig. 4(c)). This confirms our two-stage design: Stage 1 establishes a task-aware frame budget via global and text signals, while Stage 2 leverages local context to preserve fine-grained details, collectively optimizing the accuracy-efficiency trade-off.

Plug-and-Play Compatibility. One of the key advantages of GTR is its architecture-agnostic design, which enables straightforward integration with existing token compression pipelines. To further validate this property, we combine GTR with two representative training-free baselines, SparseVLM and FastV, and report the resulting performance in Fig. 5.

As shown in Fig. 5(a), integrating GTR into SparseVLM brings consistent gains on all evaluated benchmarks: MVBench improves from 99.1% to 99.6% (+0.5%), MLVU from 96.3% to 97.2% (+0.9%), and VideoMME-L from 98.6% to 99.2% (+0.6%), with an average absolute improvement of approximately +0.7%. Similar trends are observed in Fig. 5(b) for FastV. These results confirm that GTR is not tied to a specific pruning strategy. Instead, its Guide-Then-Refine mechanism provides complementary benefits on top of existing methods: the



(a) SparseVLM + GTR **(b)** FastV + GTR

Fig. 5: Plug-and-play integration results of GTR with existing methods. In both cases, adding GTR consistently improves performance across benchmarks.

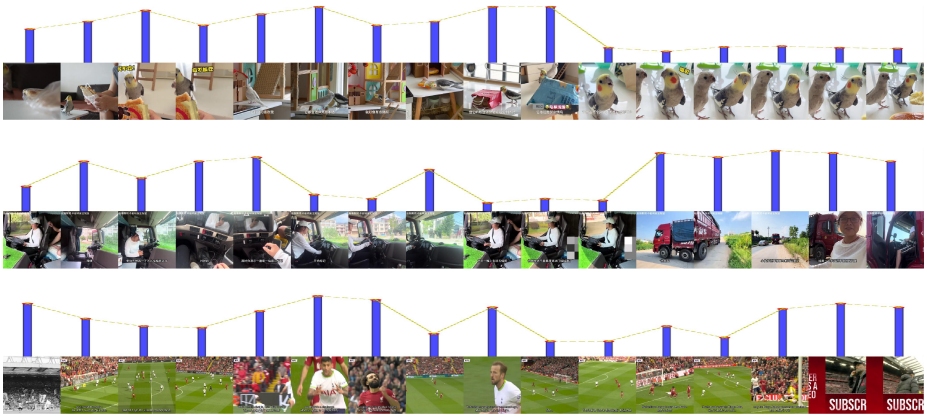


Fig. 6: Visualization of dynamic frame retention ratio by the GTR framework.

guide stage injects text-aware and global video-level signals to improve semantic relevance, while the refine stage preserves local discriminative details after compression. Therefore, even when strong baselines are already applied, GTR remains effective as an orthogonal enhancement module.

4.5 Visualization and Qualitative Analysis

To better reveal the mechanism of GTR from concrete visual evidence, we analyze what is directly shown in Figs. 6 and 7. In Fig. 6, each sampled frame is paired with a blue bar indicating the retained token ratio. Across diverse scenes (e.g., a cockatiel video, a driving scene, and a soccer broadcast), the bars rise on action-intensive or query-relevant segments and drop on near-duplicate or static frames. These patterns demonstrate that GTR does not apply a fixed budget, but reallocates tokens to moments with stronger task information.

Fig. 7 provides spatial evidence for the refine stage under the question **"What food is being cooked in the video?"**. The overlaid sparse tokens and dashed boxes show that preserved tokens are concentrated on fire-related and cooking-related regions, including the stove flame, furnace opening, pots, and manipulated tools near the heat source. In contrast, tokens are much sparser on back-

Fig. 7: Visualization of visual token selection by the GTR framework.

ground walls, empty tabletop areas, and other visually salient but question-irrelevant regions. This selective focus explains how GTR maintains strong QA performance at high compression ratios, as it preserves the minimal yet sufficient evidence essential for answering the query.

Taken together, the two figures show a consistent temporal-spatial behavior: Fig. 6 decides *when to keep more* tokens, and Fig. 7 determines *where to keep them*. This concrete evidence supports the claim that GTR performs structured, query-aware compression instead of generic saliency filtering, which is the key reason for its robust accuracy-efficiency trade-off.

5 Conclusion

In this paper, we proposed Guide-Then-Refine (GTR), a training-free and plug-and-play token compression framework for Video-LLMs that addresses long-video inference inefficiency through a two-stage design: a guide stage that uses global and text-aware signals to allocate dynamic frame-wise retention budgets, and a refine stage that performs local structure-aware token selection within each frame. Extensive experiments on multiple benchmarks and mainstream Video-LLMs demonstrate that GTR achieves a strong accuracy-efficiency trade-off under different compression ratios. With only 15% visual tokens retained, it preserves 95.8% of the original performance while remaining highly compatible with existing training-free baselines. As future work, we will extend GTR to jointly optimize token compression with latency-aware scheduling on streaming and ultra-long videos, enabling adaptive real-time deployment under dynamic compute budgets.

Acknowledgements

We would like to thank Airdoc for the philanthropic funding.

References

1. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a visual language model for few-shot learning. Advances in neural information processing systems **35**, 23716–23736 (2022)
2. An, X., Xie, Y., Yang, K., Zhang, W., Zhao, X., Cheng, Z., Wang, Y., Xu, S., Chen, C., Wu, C., et al.: Llava-onevision-1.5: Fully open framework for democratized multimodal training. arXiv preprint arXiv:2509.23661 (2025)
3. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., et al.: Qwen3-vl technical report. arXiv preprint arXiv:2511.21631 (2025)
4. Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token merging: Your vit but faster. In: International Conference on Learning Representations (2023)
5. Chen, L., Zhao, H., Liu, T., Bai, S., Lin, J., Zhou, C., Chang, B.: An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In: European Conference on Computer Vision. pp. 19–35. Springer (2024)
6. Chen, Y., Xue, F., Li, D., Hu, Q., Zhu, L., Li, X., Fang, Y., Tang, H., Yang, S., Liu, Z., He, E., Yin, H., Molchanov, P., Kautz, J., Fan, L., Zhu, Y., Lu, Y., Han, S.: Longvila: Scaling long-context visual language models for long videos. arXiv preprint arXiv:2408.10188 (2024)
7. Cheng, Z., Leng, S., Zhang, H., Xin, Y., Li, X., Chen, G., Zhu, Y., Zhang, W., Luo, Z., Zhao, D., Bing, L.: Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. arXiv preprint arXiv:2406.07476 (2024)
8. Cho, J., Lee, J., Hayat, M., Hwang, K., Porikli, F., Choi, S.: Floc: Facility location-based efficient visual token compression for long video understanding. arXiv preprint arXiv:2511.00141 (2025)
9. Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., et al.: Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261 (2025)
10. Dao, T., Fu, D., Ermon, S., Rudra, A., Ré, C.: Flashattention: Fast and memory-efficient exact attention with io-awareness. Advances in neural information processing systems **35**, 16344–16359 (2022)
11. Feng, B., Lai, Z., Li, S., Wang, Z., Wang, S., Huang, P., Cao, M.: Breaking down video llm benchmarks: Knowledge, spatial perception, or true temporal understanding? arXiv preprint arXiv:2505.14321 (2025)
12. Fu, C., Dai, Y., Luo, Y., Li, L., Ren, S., Zhang, R., Wang, Z., Zhou, C., Shen, Y., Zhang, M., et al.: Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 24108–24118 (2025)
13. Guo, Y., Dong, W., Song, J., Zhu, S., Zhang, X., Yang, H., Wang, Y., Du, Y., Chen, X., Zheng, B.: Fila-video: Spatio-temporal compression for fine-grained long video understanding. arXiv preprint arXiv:2504.20384 (2025)
14. He, B., Li, H., Jang, Y.K., Jia, M., Cao, X., Shah, A., Shrivastava, A., Lim, S.N.: Ma-lmm: Memory-augmented large multimodal model for long-term video understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13504–13514 (2024)

15. Huang, X., Zhou, H., Han, K.: Prunevid: Visual token pruning for efficient video large language models. In: Findings of the Association for Computational Linguistics: ACL 2025. pp. 19959–19973 (2025)
16. Hyun, J., Hwang, S., Han, S.H., Kim, T., Lee, I., Wee, D., Lee, J.Y., Kim, S.J., Shim, M.: Multi-granular spatio-temporal token merging for training-free acceleration of video llms. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 23990–24000 (2025)
17. Jin, P., Takanobu, R., Zhang, W., Cao, X., Yuan, L.: Chat-univi: Unified visual representation empowers large language models with image and video understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13700–13710 (2024)
18. Li, B., Zhang, Y., Guo, D., Zhang, R., Li, F., Zhang, H., Zhang, K., Zhang, P., Li, Y., Liu, Z., et al.: Llava-onevision: Easy visual task transfer. Transactions on Machine Learning Research (2025)
19. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: International conference on machine learning. pp. 19730–19742. PMLR (2023)
20. Li, K., Wang, Y., He, Y., Li, Y., Wang, Y., Liu, Y., Wang, Z., Xu, J., Chen, G., Luo, P., et al.: Mvbench: A comprehensive multi-modal video understanding benchmark. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22195–22206 (2024)
21. Li, Y., Gui, H., Fan, Z., Wang, J., Kang, B., Chen, B., Tian, Z.: Less is more, but where? dynamic token compression via llm-guided keyframe prior. Advances in Neural Information Processing Systems **38**, 156861–156904 (2026)
22. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 26296–26306 (2024)
23. Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., Lee, Y.J.: Llava-next: Improved reasoning, ocr, and world knowledge (January 2024), `https://llava-vl.github.io/blog/2024-01-30-llava-next/`, last accessed 2026/06/30
24. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. Advances in neural information processing systems **36**, 34892–34916 (2023)
25. Liu, X., Wang, Y., Ma, J., Zhang, L.: 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. pp. 1910–1924 (2025)
26. Liu, Z., Xie, C.W., Li, P., Zhao, L., Tang, L., Zheng, Y., Liu, C., Xie, H.: Hybrid-level instruction injection for video token compression in multi-modal large language models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 8568–8578 (2025)
27. Ma, J., Zhang, Q., Lu, M., Wang, Z., Zhou, Q., Song, J., Zhang, S.: Mmg-vid: Maximizing marginal gains at segment-level and token-level for efficient video llms. arXiv preprint arXiv:2508.21044 (2025)
28. Maaz, M., Rasheed, H., Khan, S., Khan, F.: Video-chatgpt: Towards detailed video understanding via large vision and language models. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 12585–12602 (2024)
29. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning. pp. 8748–8763. PMLR (2021)

30. Shao, K., Tao, K., Qin, C., You, H., Sui, Y., Wang, H.: Holitom: Holistic token merging for fast video large language models. Advances in Neural Information Processing Systems **38**, 135547–135570 (2026)
31. Tao, K., Qin, C., You, H., Sui, Y., Wang, H.: Dycoke: Dynamic compression of tokens for fast video large language models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 18992–19001 (2025)
32. Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., Fan, Y., Dang, K., Du, M., Ren, X., Men, R., Liu, D., Zhou, C., Zhou, J., Lin, J.: Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. arXiv preprint arXiv:2409.12191 (2024)
33. Wu, H., Li, D., Chen, B., Li, J.: Longvideobench: A benchmark for long-context interleaved video-language understanding. Advances in Neural Information Processing Systems **37**, 28828–28857 (2024)
34. Xing, L., Huang, Q., Dong, X., Lu, J., Zhang, P., Zang, Y., Cao, Y., He, C., Wang, J., Wu, F., et al.: Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2025)
35. Xu, L., Zhao, Y., Zhou, D., Lin, Z., Ng, S.K., Feng, J.: Pllava: Parameter-free llava extension from images to videos for video dense captioning. arXiv preprint arXiv:2404.16994 (2024)
36. Yan, S., Han, J., Tsai, J., Xue, H., Fang, R., Hong, L., Guo, Z., Zhang, R.: Crosslmm: Decoupling long video sequences from lmms via dual cross-attention mechanisms. arXiv preprint arXiv:2505.17020 (2025)
37. Yang, S., Chen, Y., Tian, Z., Wang, C., Li, J., Yu, B., Jia, J.: Visionzip: Longer is better but not necessary in vision language models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19792–19802 (2025)
38. Yang, S., Yang, J., Huang, P., Brown II, E.L., Yang, Z., Yu, Y., Tong, S., Zheng, Z., Xu, Y., Wang, M., et al.: Cambrian-s: Towards spatial supersensing in video. In: The Fourteenth International Conference on Learning Representations (2025)
39. Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid loss for language image pre-training. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11975–11986 (2023)
40. Zhang, H., Wang, Y., Tang, Y., Liu, Y., Feng, J., Dai, J., Jin, X.: Flash-vstream: Memory-based real-time understanding for long video streams. arXiv preprint arXiv:2406.08085 (2024)
41. Zhang, H., Zhang, J., Ji, X., Wang, Q., Zhang, F.: Dyntok: Dynamic compression of visual tokens for efficient and effective video understanding. arXiv preprint arXiv:2506.03990 (2025)
42. Zhang, P., Zhang, K., Li, B., Zeng, G., Yang, J., Zhang, Y., Wang, Z., Tan, H., Li, C., Liu, Z.: Long context transfer from language to vision. arXiv preprint arXiv:2406.16852 (2024)
43. Zhang, Y., Fan, C.K., Ma, J., Zheng, W., Huang, T., Cheng, K., Gudovskiy, D., Okuno, T., Nakata, Y., Keutzer, K., et al.: Sparsevlm: Visual token sparsification for efficient vision-language model inference. In: International Conference on Machine Learning. pp. 74840–74857. PMLR (2025)
44. Zhang, Y., Wu, J., Li, W., Li, B., Ma, Z., Liu, Z., Li, C.: Llava-video: Video instruction tuning with synthetic data. arXiv preprint arXiv:2410.02713 (2024)
45. Zhou, J., Shu, Y., Zhao, B., Wu, B., Liang, Z., Xiao, S., Qin, M., Yang, X., Xiong, Y., Zhang, B., et al.: Mlvu: Benchmarking multi-task long video understanding.

In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13691–13701 (2025)