EarlyTom: Early Token Compression Completes Fast Video Understanding

Supplementary Material

Overview

Due to page limitations in the main paper, we present additional quantitative experiments, detailed latency analyses, qualitative visualizations, and implementation details in this supplementary material. The content is organized as follows:

- **Section A** evaluates the generalizability of our method on a different video-LLM architecture. Specifically, we provide extensive efficiency and accuracy results on the LLaVA-Video-7B benchmark to verify the robustness of EarlyTom across different backbones. Furthermore, we extend our evaluation to the Qwen2.5-VL architecture, comparing EarlyTom against two native token reduction baselines. We also conduct fine-grained ablation studies to investigate the individual contribution of each component within our framework on Qwen architecture.
- **Section B** presents a fine-grained decomposition of the time-to-first-token (TTFT) latency. We analyze the specific contributions of vision encoding, visual token processing, and LLM prefilling to the total latency on both LLaVA-OneVision-7B and 0.5B models across different settings.
- **Section C** provides additional visualizations of the attention sink phenomenon. By visualizing attention heatmaps from the vision encoder, we further substantiate the motivation behind our decoupled spatial token selection strategy.
- **Section D** details the implementation of our framework, providing the pseudocode for the two core components: the *inner-vision encoder frame merging* and the *decoupled spatial token selection*.
- **Section E** presents the future work of EarlyTom, including potential directions for system co-design, heterogeneous inference optimization, and acceleration for the decoding stage in multimodal models.

A. Generalizability Analysis on LLaVA-Video and Qwen2.5-VL

To further verify the effectiveness and broad applicability of our framework, we extend our evaluation to the LLaVA-Video-7B model and Qwen2.5-VL-7B model.

Efficiency analysis. As detailed in Table 6, EarlyTom consistently delivers substantial improvements in computational efficiency across all tested token retention settings. By performing frame merging directly within the vision encoder, our method effectively reduces the prefilling FLOPs. For instance, at a 15% retention rate, EarlyTom reduces the FLOPs ratio to 35.1% and achieves a time-to-first-token of 947.4 ms, representing a 6.8× speedup compared to the full-token baseline (6429.3 ms). The efficiency advantages are also corroborated on the Qwen2.5-VL-7B backbone (Table 7). Specifically, while trivial baselines like Average Pooling and Uniform Subsampling result in a 16.6% FLOPs ratio, EarlyTom further optimizes this to 12.2% (67.7T), achieving a significantly faster TTFT (3667 ms) than both the full model and the native token reduction baselines.

Accuracy and trade-off. Table 6 presents a comprehensive comparison of accuracy and efficiency. EarlyTom maintains competitive performance on standard video understanding benchmarks, achieving an average score of 56.43% while operating with significantly reduced computational overhead. These results demonstrate that EarlyTom can successfully generalize to the LLaVA-Video architecture, providing an efficient inference solution that balances high throughput with reliable model performance. This robust generalizability is further evidenced by our results on the Qwen2.5-VL-7B backbone (Table 7). At a 15% token retention ratio, EarlyTom achieves an average score of 62.2%, which significantly outperforms the Uniform Subsampling and Average Pooling baselines. Notably, EarlyTom maintains higher accuracy than these trivial baselines while utilizing even fewer FLOPs, demonstrating a superior Pareto frontier in the accuracy-efficiency trade-off.

Ablation studies. To investigate the individual contribution of our proposed modules, we conduct fine-grained ablation studies on the Qwen2.5-VL architecture, as summarized in Table 7. We observe that both decoupled spatial token selection and weighted frame merging are essential for maintaining optimal performance under aggressive compression. Specifically, removing the spatial selection module leads to a performance drop from 62.2% to 61.4%, indicating its critical role in identifying and preserving informative regions across frames. Similarly, excluding the weighted merging strategy results in a decline to 61.3%, underscoring the importance of our importance-aware aggregation. These results confirm that the synergy between spatial selection and temporal merging is the key to EarlyTom's ability to preserve high-fidelity visual information while reducing token redundancy.

Table 6. **Efficiency comparison with SoTA methods on the LLaVA-Video-7B model. Best** results are in bold, second-best results are underlined. Time-to-first-token is denoted as TTFT for simplicity. All efficiency results are measured on a single NVIDIA A100 GPU.

Model	Method	Before LLM Retained Ratio	Prefilling FLOPs (T) ↓	FLOPs Ratio ↓	TTFT (ms) ↓	Throughput (tokens/s) ↑	MVBench ↑	EgoSchema ↑	LongVideo Bench ↑	VideoMME ↑	Avg. ↑ Score	Avg. ↑ %
LLaVA-Video-7B	Vanilla	100%	246.2	100%	6429.3	8.1	60.4	57.2	58.9	64.3	60.2	100
	FastV [4]$_{\text{ECCV'24}}$	100%	158.2	64.3%	3494.3	10.0	54.3	54.1	55.0	58.8	55.6	92.4
	PyramidDrop [39]$_{\text{CVPR'25}}$	100%	159.4	64.7%	3494.8	10.1	55.9	54.3	54.7	61.9	56.7	94.2
	VisionZip [42]$_{\text{CVPR'25}}$	15%	159.4	64.7%	3241.4	14.2	56.7	54.7	54.7	60.7	56.7	94.2
	HoliTom [31]$_{\text{NeurIPS'25}}$	15%	156.6	63.6%	1669.5	**17.1**	57.7	54.8	56.2	62.1	57.7	95.8
	EarlyTom	15%	**86.4**	**35.1%**	**947.4**	16.7	55.8	54.7	53.9	61.3	56.4	93.7

Table 7. **Experiment results on trivial baselines and ablation studies.** All results are obtained on Qwen2.5-VL-7B with a maximum of 768 frames and a retain ratio of 15%. Efficiency metrics are measured under a 23k-token context length on a single NVIDIA A100 GPU.

Method	Prefilling FLOPs (T) ↓	FLOPs Ratio ↓	TTFT (ms) ↓	MVBench ↑	VideoMME ↑				Avg. Score
					Short	Medium	Long	Average	
Qwen2.5-VL-7B	554.7	100%	6842	67.1	76.0	66.0	55.1	65.7	66.4
Average Pooling	91.9	16.6%	4609	56.8	66.4	57.3	51.1	58.3	57.6
Uniform Subsampling	91.9	16.6%	4578	57.7	68.6	59.6	**55.0**	60.8	59.3
EarlyTom w/o Decoupled Spatial Token Selection	67.7	12.2%	3667	60.7	**71.0**	61.6	53.6	**62.0**	61.4
EarlyTom w/o Weighted Frame Merging	67.7	12.2%	3667	60.7	70.5	**62.3**	52.7	61.8	61.3
EarlyTom	**67.7**	**12.2%**	**3667**	**62.5**	70.7	61.6	53.6	61.9	**62.2**

Table 8. Details of the hyperparameters on LLaVA-OneVision.

Retained Ratio	w. Inner LLM	EMA factor α	MVBench ↑	EgoSchema ↑	LongVideo Bench ↑	VideoMME ↑
LLaVA-OneVision-7B						
25%	✓	0.9	τ_{seg}=0.8 [6,14,20]	τ_{seg}=0.7 [10,21,23]	τ_{seg}=0.8 [6,21,23]	τ_{seg}=0.6 [10,21,23]
20%	✓	0.9	τ_{seg}=0.8 [6,14,20]	τ_{seg}=0.6 [10,21,23]	τ_{seg}=0.6 [10,21,23]	τ_{seg}=0.5 [8,21,23]
15%	✓	0.9	τ_{seg}=0.8 [6,14,20]	τ_{seg}=0.5 [10,21,23]	τ_{seg}=0.5 [10,21,23]	τ_{seg}=0.4 [8,21,23]
10%	✓	0.9	τ_{seg}=0.65 [8,14,20]	τ_{seg}=0.3 [10,21,23]	τ_{seg}=0.3 [10,21,23]	τ_{seg}=0.3 [10,21,23]
LLaVA-OneVision-0.5B						
25%	✓	0.9	τ_{seg}=0.8 [8,21,23]	τ_{seg}=0.7 [10,21,23]	τ_{seg}=0.8 [6,21,23]	τ_{seg}=0.6 [8,21,23]
20%	✓	0.9	τ_{seg}=0.8 [8,21,23]	τ_{seg}=0.6 [10,21,23]	τ_{seg}=0.6 [10,21,23]	τ_{seg}=0.5 [8,21,23]
15%	✓	0.9	τ_{seg}=0.8 [8,21,23]	τ_{seg}=0.5 [10,21,23]	τ_{seg}=0.5 [10,21,23]	τ_{seg}=0.4 [8,21,23]
10%	✓	0.9	τ_{seg}=0.65 [8,21,23]	τ_{seg}=0.3 [10,21,23]	τ_{seg}=0.3 [10,21,23]	τ_{seg}=0.3 [8,21,23]

B. Detailed Analysis of TTFT Latency Decomposition

In this section, we provide a fine-grained visualization of the Time-to-First-Token (TTFT) latency composition for both LLaVA-OneVision-7B (Figure 7) and LLaVA-OneVision-0.5B (Figure 8) under varying token retention rates (10%, 15%, 20%, and 25%). The total latency is decomposed into four components: Vision Encoding, Visual Token Processing, LLM Prefill, and System Overhead.

Analysis on LLaVA-OneVision-7B. As illustrated in Figure 7, the vision encoding stage constitutes a dominant portion of the total latency for the Baseline, HoliTom, and VisionZip. While existing methods like HoliTom and VisionZip effectively reduce the LLM prefill latency through token reduction, they fail to address the high computational cost of the vision encoder. Moreover, HoliTom introduces significant computational overhead during the Visual Token Processing stage, which partially offsets the gains from reduced prefill time. In contrast, EarlyTom directly compresses redundancy within the vision encoder, achieving a substantial reduction in encoding latency. Consequently, our method achieves the lowest total TTFT across all settings, delivering a speedup of up to 2.65× compared to the baseline at a 10% retention rate.

Analysis on LLaVA-OneVision-0.5B. The advantages of our approach are consistent across model scales. Figure 8 presents the results on the smaller 0.5B backbone. A notable observation is that on this lightweight model, the computational overhead introduced by comparison methods becomes more detrimental. Specifically, HoliTom exhibits a higher total latency than the Baseline (e.g., 0.90× speedup at 10% retention) because the time saved in the LLM prefill stage is insufficient to outweigh the extra cost of its token processing module. Conversely, EarlyTom maintains its superiority by minimizing both vision encoding time and processing overhead. Even with the smaller potential for prefill acceleration in the 0.5B model, our method achieves a robust speedup of 1.48× (at 10% retention), validating the effectiveness of our early-stage compression strategy.

C. Visualization of the Attention Sink Phenomenon Across Diverse Video Samples

In this section, we provide additional visualizations to further substantiate the analysis of the "Attention Sink" phe-

nomenon discussed in the main paper. Figure 6 displays the attention heatmaps extracted from the SigLIP vision encoder across a diverse set of video samples.

Observation. A consistent pattern emerges across all examples: distinct vertical stripes appear in the heatmaps, indicating that certain spatial tokens maintain exceptionally high attention scores throughout the entire video sequence. These tokens, often referred to as "attention sinks," act as static attractors within the feature space, dominating the attention distribution regardless of the changing visual content in dynamic frames.

Motivation for Our Design. This visualization highlights a critical insight for token compression: simply ranking tokens by attention magnitude might bias the selection towards these static sink tokens, potentially overlooking less prominent but semantically rich dynamic features. Recognizing this inherent distribution characteristic, EarlyTom adopts a Decoupled Spatial Token Selection strategy. By distinguishing between static frames (where sinks are stable) and dynamic frames, and applying tailored selection mechanisms for each, our method ensures that the compressed token set preserves both the necessary structural information (sinks) and the crucial motion-sensitive details, leading to a more robust and balanced video representation.

D. Pseudocode of EarlyTom

In this section, we provide the detailed pseudocode for the two core components of EarlyTom to facilitate implementation. Algorithm 1 outlines the inner-vision encoder frame merging process, which performs adaptive streaming segmentation and weighted merging to reduce temporal redundancy. Algorithm 2 illustrates the decoupled spatial token selection strategy, describing how dynamic and static frames are processed via distinct selection mechanisms to ensure balanced spatial information preservation.

E. Future Work

EarlyTom reveals that the inference budget is mainly dominated by the prefill stage in VLMs. Although existing methods [3, 7, 25–27, 40, 51] have proposed various techniques for efficient inference, they primarily focus on algorithm-level improvements rather than system-level optimizations. How to jointly leverage system design and algorithmic techniques in a heterogeneous manner remains an open problem. Meanwhile, recent reasoning models [8] have exhibited strong scene understanding capabilities, yet they still suffer from lengthy generation steps during the decoding stage. Therefore, accelerating inference and improving efficiency via system–algorithm co-design is essential and worthy of further exploration.

Algorithm 1 Inner-Vision Encoder Frame Merging

Input: Frame features $F \in \mathbb{R}^{B\times L\times D}$, hyperparameters $\alpha, \tau_{\text{seg}}, \tau_{\text{merge}}$.
Output: Merged frame features $\hat{F}_{\text{out}} \in \mathbb{R}^{N\times L\times D}$.
Streaming Frame Segmentation in Equation (1)
$\mathcal{S} \leftarrow \text{SegmentBySimilarity}(F, \alpha, \tau_{\text{seg}})$, $F_{\text{merged\_list}} \leftarrow [\,]$
for each segment $S_{\text{seg}} = \{F_0, \ldots, F_k\}$ in $\mathcal{S}$ **do**
  $F_{\text{mid}} \leftarrow [\,], i \leftarrow 1$
  Iterate over Middle Frames within the Segment
  while $i < k$ **do**
    Compute Pairwise Frame Similarities
    $s_i \leftarrow \text{Sim}(F_i, F_{i+1}), s_{i+1} \leftarrow \text{Sim}(F_{i+1}, F_{i+2})$
    Middle Frame Merge Condition in Equation (2)
    if $s_i > \tau_{\text{merge}}$ **and** $s_i > s_{i+1}$ **then**
      Weighted Frame Merge in Equation (3)
      $\hat{F}_m \leftarrow \text{WeightedMerge}(s_i, F_i, s_{i+1}, F_{i+1})$
      $F_{\text{mid}}.\text{append}(\hat{F}_m); i \leftarrow i + 2$
    else
      $F_{\text{mid}}.\text{append}(F_i); i \leftarrow i + 1$
    end if
  end while
  Assemble Merged Segment
  $F_{\text{seg\_out}} \leftarrow \text{Concat}(F_0, F_{\text{mid}}, F_k)$
  $F_{\text{merged\_list}}.\text{append}(F_{\text{seg\_out}})$
end for
Concatenate All Merged Segments
$\hat{F}_{\text{out}} \leftarrow \text{Concatenate}(F_{\text{merged\_list}})$
Return $\hat{F}_{\text{out}}$

Algorithm 2 Decoupled Spatial Token Selection

Input: Features $\hat{F}$ and attentions A from vision encoder, segment list $\mathcal{S}$, target ratio r.
Output: Final compressed features $\hat{\hat{F}}$.
Decouple Frames into Dynamic and Static Sets
$\hat{F}^d, A^d \leftarrow [\,], [\,]$
$\hat{F}^s, A^s \leftarrow [\,], [\,]$
for each segment $S_{\text{seg}} = \{\hat{F}_0, \ldots, \hat{F}_k\}$ in $\mathcal{S}$ **do**
  $\hat{F}^d.\text{append}(\hat{F}_0, \hat{F}_k)$; $A^d.\text{append}(A_0, A_k)$
  $\hat{F}^s.\text{append}(\hat{F}_{1:k-1})$; $A^s.\text{append}(A_{1:k-1})$
end for
Compute Re-scaled Retention Ratio in Equation (5)
$\hat{r} = \frac{r}{(\frac{B-N}{B})*L}$
Compress Dynamic Frames via Global Top-K Selection
$\hat{\hat{F}}^d \leftarrow \text{GlobalTopKSelection}(\hat{F}^d, A^d, \hat{r})$
Compress Static Frames via Local-window Selection
$\hat{\hat{F}}^s \leftarrow \text{LocalWindowSelection}(\hat{F}^s, A^s, \hat{r})$
Gather and Reorder Selected Tokens in Temporal Order
$\hat{\hat{F}} \leftarrow \text{GatherAndReorder}(\hat{\hat{F}}^d, \hat{\hat{F}}^s)$
Return $\hat{\hat{F}}$

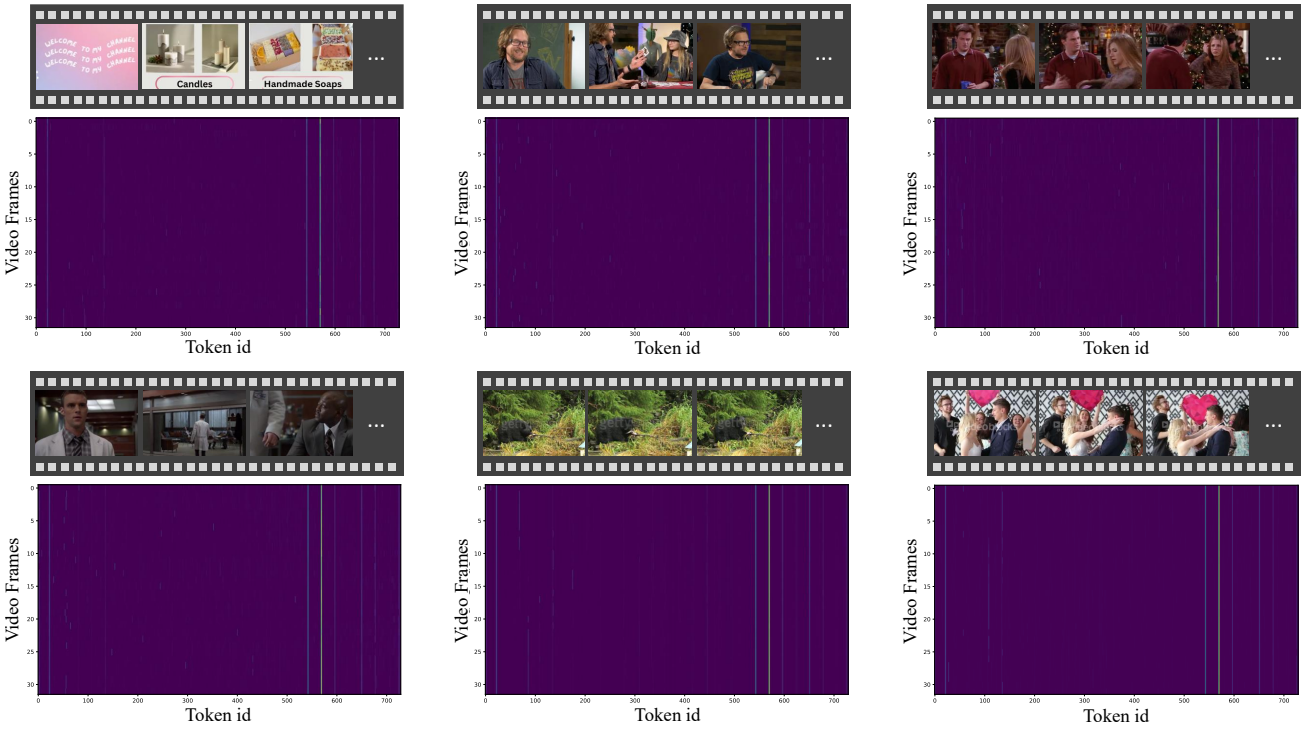
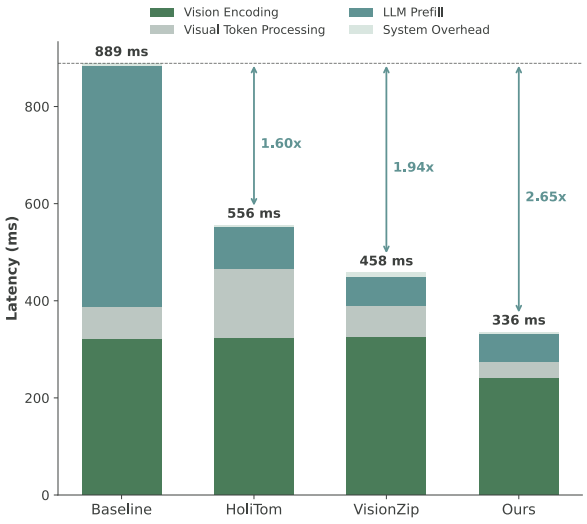
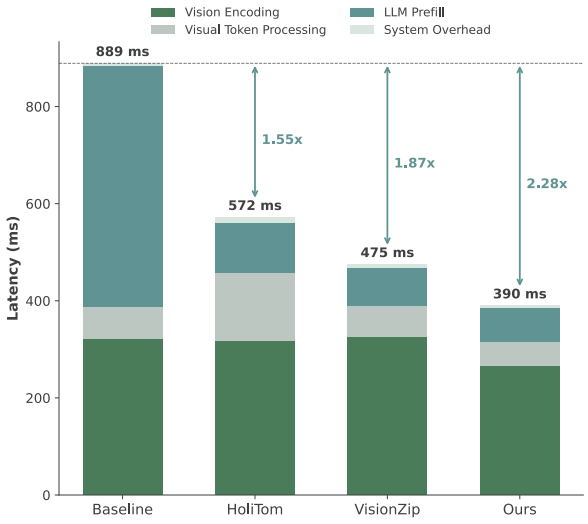


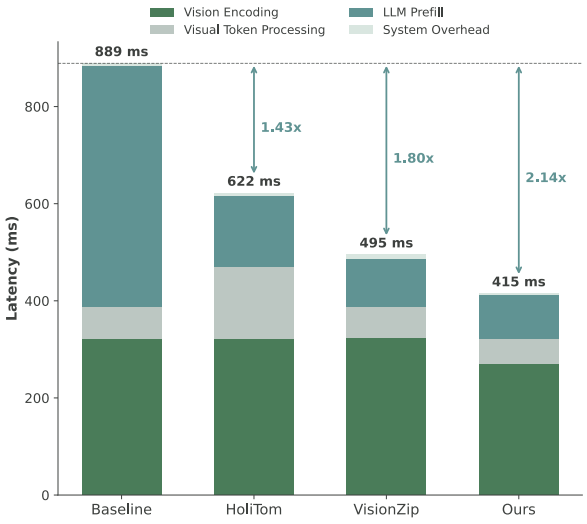
Figure 6. Additional visualizations of attention score distributions. We present the attention heatmaps from the SigLIP vision encoder across six randomly selected videos. The consistent vertical stripes (highlighted in bright colors) indicate that specific spatial tokens accumulate disproportionately high attention scores throughout the temporal sequence. This observation confirms that attention "sinks" are a widely existing structural characteristic in the vision encoder, motivating the design of our decoupled token selection strategy.



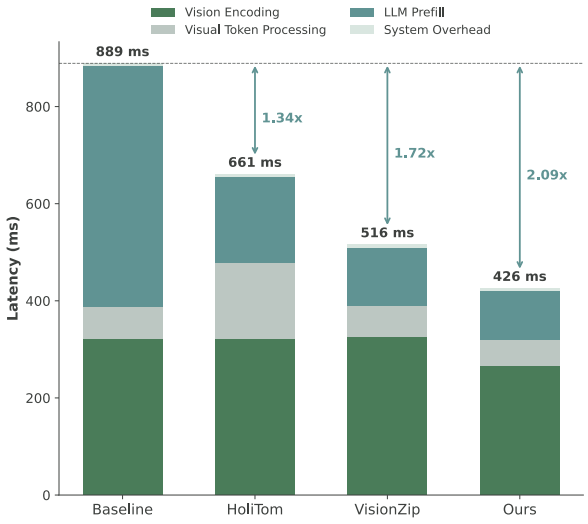
(a) 10% token retention rate.



(b) 15% token retention rate.

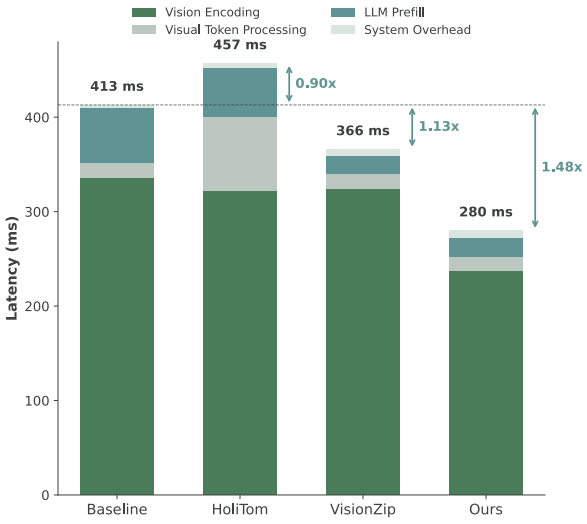


(c) 20% token retention rate.

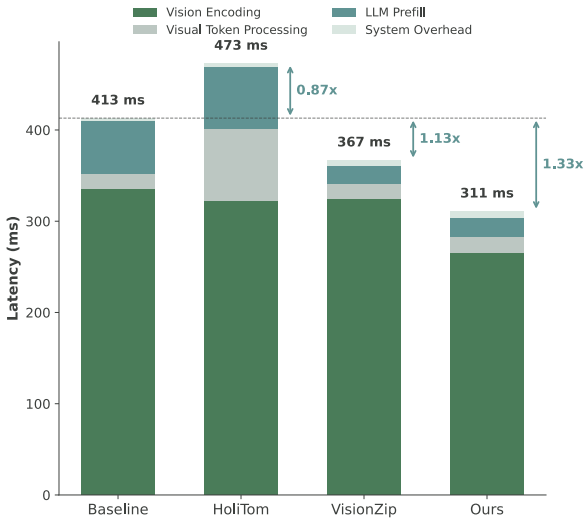


(d) 25% token retention rate.

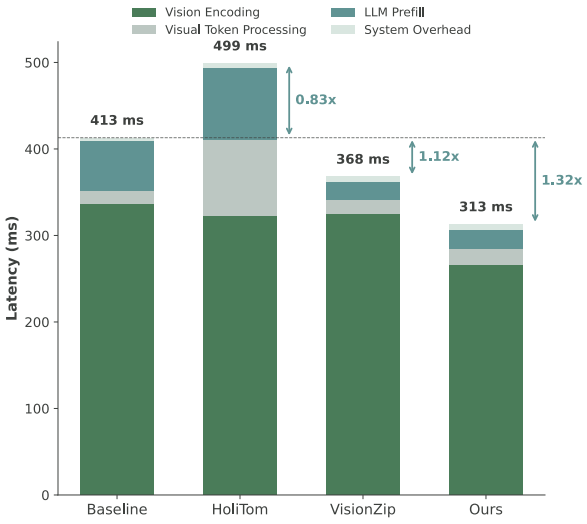
Figure 7. Time-to-first-token (TTFT) comparison on the **LLaVA-OneVision-7B** model. We report the latency breakdown (vision encoding, token processing, LLM prefill, and system overhead) across different methods.



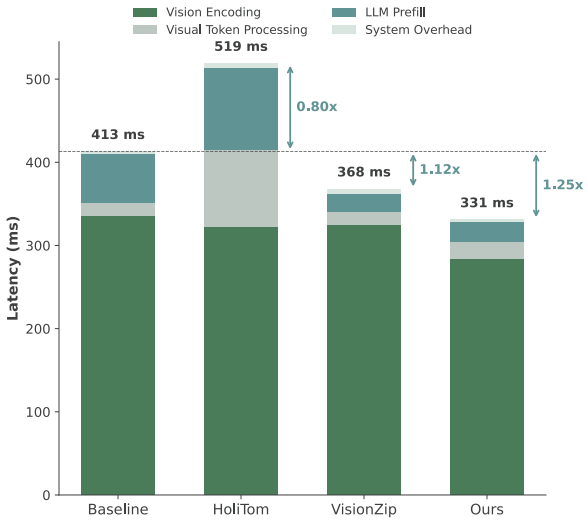
(a) 10% token retention rate.



(b) 15% token retention rate.



(c) 20% token retention rate.



(d) 25% token retention rate.

Figure 8. Time-to-first-token (TTFT) comparison on the **LLaVA-OneVision-0.5B** model. We report the latency breakdown (vision encoding, token processing, LLM prefill, and system overhead) across different methods.