STAGE-ADAPTIVE TOKEN SELECTION FOR EFFICIENT OMNI-MODAL LLMS

Anonymous authors
Paper under double-blind review

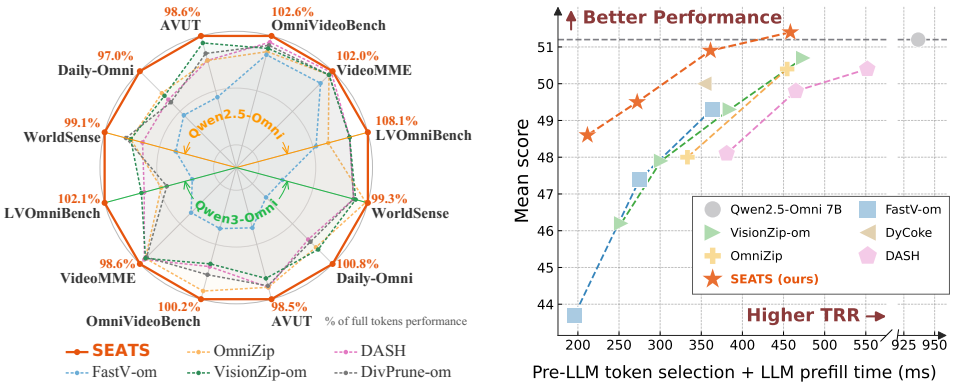


Figure 1: **Efficiency–performance trade-off** of training-free token selection methods for om-LLMs.

ABSTRACT

The unified audio-visual understanding capability of omni-modal large language models (om-LLMs) comes at the high cost of processing tens of thousands of visual and audio tokens throughout the LLM. To reduce the cost, training-free methods for om-LLM token selection are being actively developed, operating either pre-LLM or inner-LLM. Recognizing that the non-textual token redundancy varies across stages and LLM layers, we propose in this paper a novel stage-adaptive token selection (SEATS) method that operates in both pre-LLM and inner-LLM stages. Given an overall token budget measured by the token retention rate (TRR), SEATS performs top-down token budget allocation in an input-dependent manner. Specifically, a larger TRR is used for pre-LLM token selection, whilst within the LLM, all tokens are retained in shallow layers, increasingly decreased TRRs are applied in middle layers, and the non-textual tokens are fully removed in late layers. Extensive experiments on three representative om-LLMs and six benchmarks verify the viability of the proposed method. By retaining only 10% of visual and audio tokens, SEATS preserves 94.9% of the full-token performance on Qwen2.5-Omni while achieving a 9.3× FLOPs reduction and a 4.9× prefill speedup.

1 INTRODUCTION

Omni-modal large language models (om-LLMs) have shown great potential for unified audio-visual understanding (Qwen Team, 2026; Ye et al., 2026; Li et al., 2025; Tang et al., 2025). They encode video frames and audio tracks into temporally aligned token sequences and concatenate them with text tokens for joint LLM reasoning. Their performance relies on dense video frame sampling and high-resolution audio encoding, typically yielding tens of thousands of visual and audio tokens to be fed into an LLM. Processing such lengthy tokens throughout the LLM incurs substantial computation and memory overhead. Therefore, selecting compact yet semantically sufficient visual and audio tokens is crucial for efficient om-LLM inference.

Token selection (Shao et al., 2026; Yao et al., 2026) has been widely studied for image-LLMs (Li et al., 2024a;b) and video-LLMs (Zhang et al., 2025e; Wang et al., 2025; Bai et al., 2025; Yu et al., 2025; Ge et al., 2025). Depending on where selection is performed, existing methods can be

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

broadly categorized into pre-LLM methods and inner-LLM methods. Pre-LLM methods (Yang et al., 2025a; Alvar et al., 2025; Deng et al., 2025) reduce input length using encoder-side signals before LLM computation, but are often query-agnostic and may discard task-critical tokens. Inner-LLM methods (Chen et al., 2024; Wu et al., 2026; Xing et al., 2025) exploit text-to-vision attention for query-aware pruning, but shallow-layer attention is noisy, while late pruning limits computational savings. For video-LLMs, spatiotemporal redundancy further motivates frame-aware selection (Shen et al., 2025; Fan et al., 2026) and hybrid pre-/inner-LLM strategies (Shao et al., 2025; Tao et al., 2025; Du et al., 2026). Despite these advances, existing methods are mainly developed for image- and video-LLMs, so they naturally fall short when applied to the temporally interleaved audio-visual tokens of om-LLMs.

Recent studies have begun to explore token selection for om-LLMs, the majority of which operate in the pre-LLM stage. OmniZip (Tao et al., 2026a) and DASH (Li & Huang, 2026), for instance, use audio encoder attention to guide video token pruning. EchoingPixels pools audio and video tokens for cross-modal joint filtering (Gong et al., 2026). OmniSIFT performs spatiotemporal video pruning followed by visual-semantic-guided audio token selection (Ding et al., 2026). By contrast, OmniFit performs token reduction inside an om-LLM (Wang et al., 2026), but lacks a unified and adaptive token-retention budget allocation strategy across LLM layers, temporal windows, and modalities. Our empirical analysis reveals a clear block-wise dependence pattern: LLM layers in *shallow* blocks strongly rely on non-textual tokens for cross-modal fusion, layers in *middle* blocks gradually reduce this dependence, while for *late* blocks where the fusion is largely completed, the non-textual tokens are dispensable. We note that the design of existing methods actually runs counter to this pattern: FastV prunes visual tokens in shallow layers, where such tokens remain crucial for cross-modal fusion (Chen et al., 2024), while HiDrop (Wu et al., 2026) retains more visual tokens in deeper layers, where these tokens have already become redundant. This motivates us to search for a stage-adaptive, depth-aware, and modality-flexible token selection strategy for om-LLMs.

Designing such a strategy is non-trivial due to challenges as follows. First, the nature of token redundancy differs across stages: pre-LLM tokens mainly exhibit spatiotemporal similarity, whereas inner-LLM tokens become query-aligned and should be selected in an *input-dependent* manner. Second, reliance on non-textual tokens decreases with the LLM's layer depth, making a uniform pruning ratio either too aggressive for shallow layers or too conservative for deeper layers. Third, audio-visual importance varies across temporal windows, where either modality may provide the key evidence. Such varied importance cannot be handled by a fixed token retention ratio (`TRR`).

To address these challenges, we propose `SEATS`, a training-free StagE-Adaptive Token Selection method for efficient om-LLM inference. Before the LLM, `SEATS` applies attention-weighted diversity selection within each temporal window to remove spatiotemporal redundancy and shorten the input sequence. Inside the LLM, it adopts a model-agnostic block-wise token-retention-ratio (`TRR`) decay schedule, progressively increasing pruning strength as the dependence on non-textual tokens decreases. It further distributes the retention budget through a top-down two-level allocation strategy, first across temporal windows and then across modalities, guided by query relevance scores. In late layers, where cross-modal fusion is largely completed, `SEATS` removes all remaining non-textual tokens so that subsequent layers process only text tokens. Together, these stages enable token selection that adapts to both layer-wise dependency and cross-modal dynamics without retraining.

Extensive experiments on six audio-visual benchmarks and three representative om-LLMs, Qwen2.5-Omni-7B, OmniVinci-9B and Qwen3-Omni-30B, verify the viability of `SEATS`. It is comparable to the full-token performance while using only 33% computational cost on Qwen2.5-Omni-7B, see Fig. 1. At a `TRR` of 0.1, it achieves a $9.3\times$ FLOPs reduction and a $4.9\times$ prefill speedup while preserving 94.9% of the original performance. To sum up, our main contributions are as follows:

- **Insight**. We reveal a block-wise modality dependence pattern in current om-LLMs (OmniVinci and Qwen-Omni series), where visual and audio tokens become less important for deeper LLM layers. This finding sheds light on a more aggressive yet effective path to om-LLM token selection.
- **Solution**. The proposed `SEATS` operates in both pre-LLM and inner-LLM stages. For pre-LLM token selection, we extend DivPrune (Alvar et al., 2025) to the omni-modal context and in a window-based manner. Our inner-LLM token selection follows a block-wise `TRR` decay schedule and query-guided top-down token budget allocation.
- **Results**. Experiments on three om-LLMs and six benchmarks show that `SEATS` achieves a superior efficiency-performance trade-off, compared with a number of strong *open-source* baselines including FastV (Chen et al., 2024), VisionZip (Yang et al., 2025a), and DivPrune (Alvar et al., 2025) originally

for image-LLMs, DyCoke (Tao et al., 2025) and FastVID (Shen et al., 2025) developed for video-LLMs, and OmniZip (Tao et al., 2026a) and DASH (Li & Huang, 2026) dedicated to om-LLMs.

2 Related Work

As we aim for training-free token selection, we discuss recent progress in this line of research (Tab. 1).

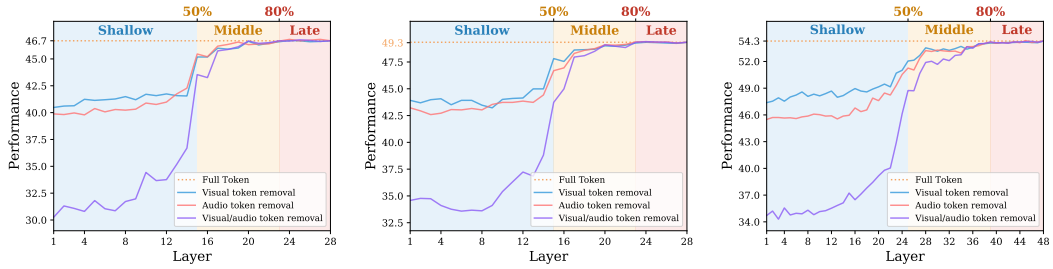
Table 1: **Training-free token selection for image-(🖼), video-(🎞), and om-(🎞+🔊) LLMs** .

Method	Open-source	External-model Free	Targeted MLLM	Selection Stage		Selection Criterion		Adaptive TRR	Late Removal
				Pre-LLM	*Inner-LLM*	*Relevance*	*Diversity*		
VisionZip (Yang et al., 2025a) CVPR'25	✓	✓	🖼	✓	✗	🖼	✗	✗	✗
DivPrune (Alvar et al., 2025) CVPR'25	✓	✓	🖼	✓	✗	✗	🖼	✗	✗
CDPruner (Zhang et al., 2025c) NIPS'25	✓	✗	🖼	✓	✗	✗	🖼	✗	✗
SCOPE (Deng et al., 2025) NIPS'25	✓	✓	🖼	✓	✗	🖼	🖼	✗	✗
FastV (Chen et al., 2024) ECCV'24	✓	✓	🖼	✗	✓	🖼	✗	✗	✗
PDrop (Xing et al., 2025) CVPR'25	✓	✓	🖼	✗	✓	🖼	✗	✓	✗
HiDrop (Wu et al., 2026) ICLR'26	✓	✓	🖼	✗	✓	🖼	✗	✓	✓
FastVID (Shen et al., 2025) NIPS'25	✓	✓	🎞	✓	✗	🎞	🎞	✗	✗
DyCoke (Tao et al., 2025) CVPR'25	✓	✓	🎞	✓	✓	🎞	🎞	✗	✗
HoliTom (Shao et al., 2025) NIPS'25	✓	✓	🎞	✓	✓	🎞	🎞	✗	✗
FlashVID (Fan et al., 2026) ICLR'26	✓	✓	🎞	✓	✓	🎞	🎞	✗	✗
UniST (Du et al., 2026) CVPR'26	✗	✓	🎞	✓	✓	🎞	🎞	✗	✗
OmniZip (Tao et al., 2026a) CVPR'26	✓	✓	🎞+🔊	✓	✗	🔊	🎞	✗	✗
DASH (Li & Huang, 2026) ECCV'26	✓	✓	🎞+🔊	✓	✗	🔊	🎞+🔊	✗	✗
OmniScope (Su et al., 2026) MM'26	✓	✗	🎞+🔊	✓	✗	🎞+🔊	🎞+🔊	✗	✗
OmniFit (Wang et al., 2026) ICML'26	✗	✓	🎞+🔊	✗	✓	✗	🎞+🔊	✗	✗
SEATS (*ours*)	✓	✓	🎞+🔊	✓	✓	🎞+🔊	🎞+🔊	✓	✓

For image-LLMs. Depending on whether token selection is performed before or inside the LLM, existing methods can be divided into two groups: pre-LLM (Yang et al., 2025a; Shang et al., 2025; Zhang et al., 2025b; Deng et al., 2025; Zhang et al., 2025c; Dong et al., 2026) and inner-LLM (Chen et al., 2024; Xing et al., 2025; Zhang et al., 2025d; Wu et al., 2026). For pre-LLM token selection, VisionZip (Yang et al., 2025a), LLaVA-PruMerge (Shang et al., 2025), and VisPruner (Zhang et al., 2025b) measure token saliency via `[CLS]` attention. DivPrune (Alvar et al., 2025) formulates token selection as a max-min diversity problem. SCOPE (Deng et al., 2025) and CDPruner (Zhang et al., 2025c) consider both saliency and diversity, whilst MMTok (Dong et al., 2026) performs multimodal coverage-based selection. Since visual and textual tokens are not semantically aligned in the pre-LLM stage, these methods are typically user-query agnostic. By contrast, inner-LLM methods prune visual tokens at specific LLM layers based on text-to-vision attention, making them inherently query-aware. FastV (Chen et al., 2024) performs one-shot pruning at a shallow layer. PDrop (Xing et al., 2025) and SparseVLM (Zhang et al., 2025d) perform token selection across multiple layers with a fixed `TRR`. HiDrop (Wu et al., 2026) operates at middle-to-deep layers with a concave schedule such that deeper layers are assigned larger `TRR`s. Different from HiDrop, `SEATS` employs a model-agnostic stage-adaptive `TRR` decay schedule, where `TRR` progressively decreases as LLM layers go deep.

For video-LLMs. Pre-LLM methods have been extended to the video domain by exploiting inter-frame token redundancy, see for instance FastVID (Shen et al., 2025), FlashVID (Fan et al., 2026), and VidCom2 (Liu et al., 2025). Meanwhile, we observe a growing interest in jointly using pre-LLM and inner-LLM approaches (Tao et al., 2025; Shao et al., 2025; Huang et al., 2025). DyCoke first merges temporally redundant tokens in the pre-LLM stage, and then dynamically reduces the KV cache within the LLM (Tao et al., 2025). HoliTom performs both pre-LLM and inner-LLM token merging (Shao et al., 2025). PruneVID (Huang et al., 2025) and UniST (Du et al., 2026) first perform spatial-temporal merging in the pre-LLM stage, and then conduct query-aware token selection inside the LLM. As these methods are designed for single-modality (visual) token pruning, directly applying them to om-LLMs, say by handling the visual and audio tokens in parallel, is suboptimal.

For om-LLMs. Token selection for om-LLMs is mostly conducted in the pre-LLM stage. OmniZip uses audio saliency to guide video pruning in fixed-length windows (Tao et al., 2026a). DASH replaces the windows by variable-length audio-semantic segments (Li & Huang, 2026). OmniScope resorts to multiple external models for token-to-query relevance estimation, thus incurring extra computation overhead (Su et al., 2026). In contrast to the above works, OmniFit performs layer-wise token selection within the LLM (Wang et al., 2026). Recognizing that token redundancy grows with layer depth, OmniFit allocates smaller token retention budgets to deeper layers. However, its budget allocation scheme is computed offline and thus cannot adapt to individual inputs.



(a) Qwen2.5-Omni-7B (b) OmniVinci-9B (c) Qwen3-Omni-30B

Figure 2: **The impact of full visual / audio token removal on the performance of an om-LLM**. Test set: WorldSense. Depending on the impact, we roughly divide the LLM layers into three blocks: *shallow* layers, where removal causes a collapse in model performance, *middle* layers, where removal leads to moderate performance loss, and *late* layers, where removal has no impact on performance.

3 Token Selection for Om-LLM: Preliminaries and Observations

3.1 Preliminaries

Let $\mathcal{V}$ be a specific video accompanied with an audio track $\mathcal{A}$. Given a user-provided prompt query $\mathcal{Q}$, an om-LLM answers with respect to the video by first encoding the video content as a sequence of N_v visual tokens, the audio track as a sequence of N_a audio tokens and the query as a sequence of N_q textual tokens. Each token is a d-dimensional vector, denoted by z. When necessary, we use z_v, z_a and z_q to denote visual, audio and textual tokens, respectively. These token sequences are then concatenated and fed into an L-layer LLM, which generates a response to the query by producing a new sequence of textual tokens in an autoregressive manner.

For temporal alignment between the visual and audio modalities, the visual and audio token sequences are first partitioned using a fixed-size window, resulting in T non-overlapping windows. For each window t, the visual and audio tokens that fall within it are grouped as $[z_{v,1}^{(t)}, \ldots, z_{v,n_v}^{(t)}, z_{a,1}^{(t)}, \ldots, z_{a,n_a}^{(t)}]$, where n_v and n_a indicate the number of visual and audio tokens in that window, respectively. These T groups are then concatenated in chronological order, followed by the textual tokens, to form the input sequence of length $N_v + N_a + N_q$ to the LLM. Since $N_o = N_v + N_a \gg N_q$, token selection for efficient LLM prefill effectively reduces to selecting the visual and audio tokens only, with the textual tokens kept entirely intact.

For each layer l in the LLM, let r_l be the token retention ratio (`TRR`) applied to its input, which reduces the input length from $N_o + N_q$ to $N_o \cdot r_l + N_q$. The value of r_l governs the trade-off between model performance and efficiency. Intuitively, r_l needs to be proportional to the importance of layer l. Given the overall `TRR` R as a token-budget indicator, *i.e.* $\sum_{l=1}^{\mathrm{L}} r_l \leq \mathrm{L} \cdot R$, more important layers should be assigned larger r_l values. Meanwhile, given R_v and R_a as the overall `TRR` for visual and audio tokens, respectively, we have $N_o \cdot R = N_v \cdot R_v + N_a \cdot R_a$.

3.2 Observations

To empirically identify layer importance, we examine the effect of removing all visual and/or audio tokens at a specific LLM layer of an om-LLM. This approach allows us to measure the extent to which each layer relies on these non-textual tokens.

As shown in Fig. 2, a consistent trend emerges across three representative om-LLMs (Qwen2.5-Omni-7B (Xu et al., 2025a), OmniVinci-9B (Ye et al., 2026), and Qwen3-Omni-30B (Xu et al., 2025b)). When l falls within the first 50% of layers, which we term the *shallow* block, removal causes a clear performance collapse, indicating that the visual and audio information has not yet been absorbed by the textual tokens. As l goes beyond 50%, *i.e.* into the *middle* block, model performance recovers rapidly, suggesting that intensive cross-modal fusion is underway and the textual tokens are progressively acquiring the needed audio-visual semantics. Once l exceeds roughly 80% of the

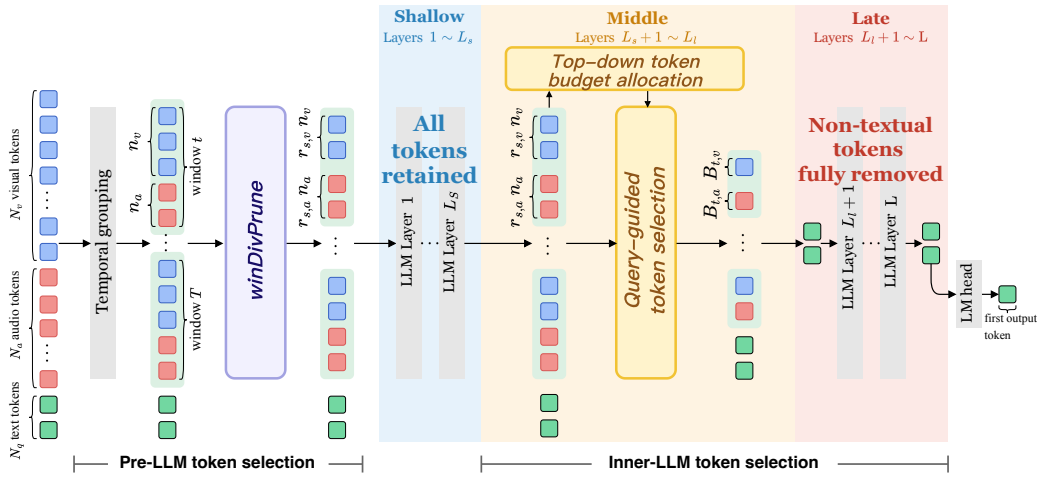


Figure 3: **SEATS for om-LLM token selection**. Given an L-layer om-LLM, the boundaries between the shallow, middle and late blocks are determined in terms of prespecified ratios (derived from Fig. 2), *i.e.* $L_s = 0.5 \times \mathrm{L}$ and $L_l = 0.8 \times \mathrm{L}$. Such a simple block partition works for Qwen2.5-Omni-7B ($\mathrm{L} = 28$), OmniVinci-9B ($\mathrm{L} = 28$) and Qwen3-Omni-30B ($\mathrm{L} = 48$).

total depth, entering the *late* block, removal causes almost no performance drop, indicating that the non-textual tokens are no longer needed.

The above results reveal a clear block-wise pattern of layer importance. Layers in the shallow block critically depend on the visual and audio tokens and thus demand a relatively high TRR. By contrast, layers in the middle block are more resistant to token removal as cross-modal fusion proceeds, so they can be allocated smaller TRR values. As for the late-block layers, the non-textual tokens can be safely removed without affecting model performance.

4 Proposed Method

As illustrated in Fig. 3, SEATS is a three-stage method. The first stage performs pre-LLM token selection (Sec. 4.1), the second stage performs inner-LLM token selection (Sec. 4.2), whilst the last stage simply removes all *non-text* tokens at the late LLM layers.

4.1 Stage I: Pre-LLM Token Selection by Window-based DivPrune

Much redundancy exists in both visual and audio tokens in the pre-LLM stage. For instance, visual tokens within a given window typically show high inter-token affinity, especially in low-motion regions. In order to select a compact yet diverse subset, we extend DivPrune (Alvar et al., 2025), originally proposed for image token selection, to the omni-modal context. DivPrune selects tokens by greedily solving a max-min diversity problem, where the objective is to maximize the minimum inter-token distance within the selected subset. To that end, a token-wise distance matrix is computed. We adapt DivPrune for omni-modal token selection as follows. First, for efficiency, instead of computing the distance matrix for all input tokens, we restrict the computation to a per-window and per-modality basis. Second, to encourage the selection of salient tokens, the matrix is row-wise reweighed by each token's attention scores. We term the adapted DivPrune winDivPrune.

Recall that in our design, the TRR progressively goes down as the tokens propagate forward. Therefore, the pre-LLM TRR, denoted by r_s, shall be larger than R. To this end, letting $r_{s,v}$ and $r_{s,a}$ be the visual and audio pre-LLM TRRs, respectively, we set $r_{s,v} = \lambda R_v$ and $r_{s,a} = \lambda R_a$, where $\lambda > 1$ is a pre-specified scale factor. Consequently, after the winDivPrune operation, the number of non-textual tokens to be forwarded to the LLM is reduced from N_o to $r_{s,v}N_v + r_{s,a}N_a$.

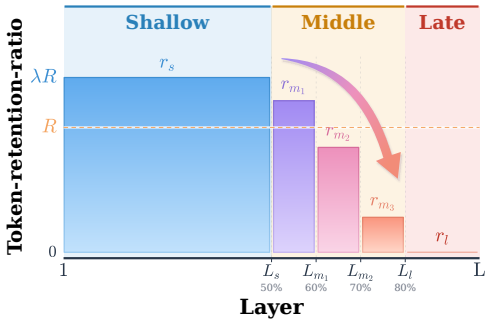
4.2 Stage II: Inner-LLM Token Selection with Top-down Token Budget Allocation

4.2.1 Block-wise TRR Decay Schedule

Based on the pattern of block-wise layer importance (Sec. 3.2), we roughly divide the L layers of the LLM into three blocks, *i.e. shallow*, *middle*, and *late*, with two hyperparameters L_s and L_l indicating the shallow-middle and middle-late boundary layers, respectively. Consequently, we propose a block-wise decay schedule for per LLM-layer TRR allocation, as detailed in Fig. 4.

Since layers in the shallow block are critical for cross-modal fusion, no token pruning is performed in these layers. The visual and audio TRRs are kept identical to their pre-LLM counterparts, $r_{s,v}$ and $r_{s,a}$. For notational simplicity, we omit the modality subscript and simply write r_s in the following.

The middle block is responsible for token selection with progressively decayed TRRs. As the layer importance diminishes with depth, deeper layers can afford more aggressive token pruning. For fine-grained TRR allocation, we define alongside L_s two extra TRR-transition layers, L_{m_1} and L_{m_2}. Accordingly, the middle block is divided into three sub-blocks with layer ranges $(L_s, L_{m_1}]$, $(L_{m_1}, L_{m_2}]$, and $(L_{m_2}, L_l]$. The TRR decreases across sub-blocks with a linearly increasing step. In particular, let r_{m_i} be the TRR of sub-block i (=1, 2, 3). Our decay schedule is defined as $r_{m_i} = r_{m_{i-1}} - i \cdot \delta$, with $r_{m_0} = r_s$ and δ a scale factor. This schedule enables earlier sub-blocks to undergo relatively mild token pruning while later sub-blocks discard tokens more aggressively, see Fig. 4. Note that with λ provided, δ can be computed analytically (Appendix C.2). All the transition layers, *i.e.* $\{L_s, L_{m_1}, L_{m_2}, L_l\}$, are selected using a fixed layer-to-L ratio, *i.e.* $\lfloor \mathrm{L} \cdot p \rfloor + 1$, where p is taken from $\{0.5, 0.6, 0.7, 0.8\}$, respectively (Tab. 10).



Block	Layer range	TRR per LLM-layer
Shallow	$[1, L_s]$	$r_s = \lambda \cdot R$
Middle	$(L_s, L_{m_1}]$	$r_{m_1} = r_s - \delta$
	$(L_{m_1}, L_{m_2}]$	$r_{m_2} = r_{m_1} - 2\delta$
	$(L_{m_2}, L_l]$	$r_{m_3} = r_{m_2} - 3\delta$
Late	$(L_l, \mathrm{L}]$	$r_l = 0$

Figure 4: **Proposed block-wise decay schedule for per-LLM-layer TRR.**

4.2.2 Top-down Token Budget Allocation

For each layer in the middle block, substituting R_v and R_a for R in Fig. 4 yields its visual and audio TRRs, denoted as r_v and r_a, respectively. The layer then accepts $r_v N_v$ visual tokens and $r_a N_a$ audio tokens as input. Recall that the input tokens are grouped into windows along the temporal dimension. Intuitively, windows containing more relevant information w.r.t. to the user query should be allocated a higher token budget. Similarly, within every window, the modality (visual or audio) that is more relevant w.r.t. the user query should also receive a larger budget relative to the other modality. In that regard, we propose a top-down strategy for query-guided token budget allocation.

Inter-window token budget allocation. For each window t (=$1, \ldots, T$), we measure its relevance to the user query, denoted as S_t, based on the causal self-attention scores between the query and the visual and audio tokens within the window. Specifically, the query is represented by the last textual token, which has attended to all preceding tokens under causal attention. The visual-based window-query relevance score $S_{t,v}$ is computed as the mean of the query-to-visual-tokens attention scores, and then normalized using a softmax. In a similar manner, we obtain the audio-based relevance score $S_{t,a}$. The overall window-query relevance S_t is then defined as the average of $S_{t,v}$ and $S_{t,a}$. The token budget B_t allocated to window t is computed as $(r_v N_v + r_a N_a) S_t$.

Intra-window token budget re-allocation. For token budget re-allocation within each window, we jointly consider each modality's layer-wise budget and its relevance to the query, computing the

window-wise visual and audio token budgets, $B_{t,v}$ and $B_{t,a}$, as follows:

$$\begin{cases} B_t &= (r_v \cdot N_v + r_a \cdot N_a) S_t \\ B_{t,v} &= \dfrac{S_{t,v} \cdot r_v \cdot N_v}{S_{t,v} \cdot r_v \cdot N_v + S_{t,a} \cdot r_a \cdot N_a} B_t \\ B_{t,a} &= B_t - B_{t,v}. \end{cases} \tag{1}$$

Note that if Eq. (1) does not fully allocate the budget, the remaining tokens will be re-allocated proportionally to S_t to ensure $\sum_{t=1}^{T} B_t = r_v N_v + r_a N_a$.

4.2.3 Query-guided Visual and Audio Token Selection

In order to select $B_{t,v}$ visual tokens from window t, we sort the visual tokens in descending order by the previously computed query-to-visual-tokens attention scores, and consequently retain the top $B_{t,v}$ tokens. Audio tokens are selected in a similar vein.

5 Experiments

5.1 Experimental Setup

Test sets. We evaluate SEATS on the following six test sets, commonly used to evaluate an om-LLM's audio-visual understanding abilities: *WorldSense* (Hong et al., 2026), *Daily-Omni* (Zhou et al., 2025), *AVUT* (Yang et al., 2025b), *OmniVideoBench* (Li et al., 2026), *Video-MME* (Fu et al., 2025), and *LVOmniBench* (Tao et al., 2026b).

Choice of om-LLM. We experiment with three open-source om-LLMs, *i.e.* Qwen2.5-Omni-7B (28-layer LLM) (Xu et al., 2025a), OmniVinci-9B (28-layer LLM) (Ye et al., 2026), and Qwen3-Omni-30B (A3B-Instruct, 48-layer MoE-based LLM) (Xu et al., 2025b). Note that Qwen3-Omni-30B has an audio token rate of 13 tokens per second, lower than Qwen2.5-Omni-7B's and OmniVinci's 25 tokens per second. Consequently, for the same overall TRR (R), the visual TRR (R_v) and audio TRR (R_a) differ between the om-LLMs.

Baselines. To ensure a fair and reproducible comparison, we opt for baseline methods that are training-free, applicable either before or during the prefill stage, requiring no *external* model, and *open-source*. Eventually, we compile a list of seven recent methods, adapting them as needed for om-LLM. Depending on their targeted modalities, *i.e.* image, video or omni-modal, the baselines are categorized into the following three groups:

- *Image*: FastV (Chen et al., 2024), VisionZip (Yang et al., 2025a) and DivPrune(Alvar et al., 2025). Applying each method in parallel to visual and audio tokens yields an omni-modal variant that we refer to as FastV-om, VisionZip-om, and DivPrune-om, respectively.
- *Video*: DyCoke (Tao et al., 2025), FastVID (Shen et al., 2025). Following (Tao et al., 2026a; Ding et al., 2026), for DyCoke we use its prefill-stage TTM module only.
- *Omni-modal*: OmniZip (Tao et al., 2026a), DASH (Li & Huang, 2026) and Random that randomly selects tokens at a given ratio.

Implementation. We refer to Appendix C.2.

5.2 SEATS *versus* SOTA

Results on Qwen2.5-Omni-7B. As shown in Tab. 2, SEATS achieves the best average performance across all retention ratios, surpassing the full-token baseline at 35% (51.4 *vs*. 51.2) and retaining 94.9% of it at 10% with only 11% of the FLOPs. Among the baselines, FastV and FastV-om rank lowest, indicating that shallow-layer relevance scores are too noisy for reliable one-shot pruning, whereas VisionZip-om and DivPrune-om match OmniZip and DASH, suggesting that pre-LLM local saliency selection already captures information comparable to joint audio-visual compression. Daily-Omni is the exception, as its reliance on audio evidence allows video-only methods that keep all audio tokens to score higher, yet SEATS still reaches the best mean score.

Efficiency analysis is provided in Tab. 3. The reported prefill time includes inner-LLM token selection overhead, whereas TTFT further accounts for encoder forward and pre-LLM compression. Owing to careful code optimization with vectorized tensor operations, the token selection overhead

Table 2: **Results on Qwen2.5-Omni-7B**. Per method, the visual token retention ratio R_v and the audio counterpart R_a are adjusted to satisfy the given overall ratio R. **Bold** and underline denote the best and second-best per column. Methods sorted in ascending order by their mean performance.

Method	R (R_v-R_a)	TFLOPs↓	World Sense	Daily Omni	AVUT	OmniVideo Bench	Video MME	LVOmni Bench	Mean
Full tokens	100 (100-100)	111.0	46.7	64.0	63.6	34.1	65.3	33.3	51.2 100.0%
FastV-om	35 (30-65)	38.3	43.3	58.7	59.2	34.4	65.6	34.3	49.3 96.3%
FastV ECCV'24	35 (24-100)	38.3	43.7	59.6	59.1	34.7	65.2	34.8	49.5 96.7%
Random	35 (30-65)	37.4	44.6	59.7	61.1	34.1	65.0	34.9	49.9 97.5%
DyCoke CVPR'25	35 (24-100)	37.7	44.4	59.9	60.3	34.6	66.0	34.7	50.0 97.7%
FastVID NIPS'25	35 (24-100)	37.1	44.6	59.8	60.4	34.6	65.4	35.8	50.1 97.9%
VisionZip CVPR'25	35 (24-100)	37.4	44.5	60.6	60.7	33.6	66.2	35.3	50.2 98.0%
DivPrune CVPR'25	35 (24-100)	37.4	44.8	60.7	61.0	33.9	66.1	35.7	50.4 98.4%
OmniZip CVPR'26	35 (30-65)	38.2	45.2	60.4	61.3	34.5	66.3	34.7	50.4 98.4%
DASH ECCV'26	35 (30-65)	37.8	44.7	59.9	61.3	34.8	66.5	35.4	50.4 98.4%
DivPrune-om	35 (30-65)	37.4	45.4	59.7	61.7	34.7	66.3	35.4	50.5 98.6%
VisionZip-om	35 (30-65)	37.4	45.2	60.2	62.3	34.6	66.3	35.4	50.7 99.0%
SEATS	35 (30-65)	36.5	**46.3**	**62.1**	**62.7**	**35.0**	**66.6**	**36.0**	**51.4** 100.4%
FastV	25 (12-100)	28.3	42.4	56.5	54.2	33.4	62.8	33.8	47.2 92.2%
FastV-om	25 (20-55)	28.3	41.6	56.1	57.6	33.0	63.4	32.7	47.4 92.6%
OmniZip	25 (25-25)	23.9	42.7	55.5	57.7	32.2	66.0	34.0	48.0 93.8%
Random	25 (20-55)	27.6	43.0	57.3	59.1	33.5	64.2	34.6	48.6 94.9%
FastVID	25 (12-100)	27.0	43.4	58.5	57.8	33.8	64.3	34.8	48.8 95.3%
VisionZip	25 (12-100)	27.6	42.4	59.7	58.4	34.4	64.1	34.5	48.9 95.5%
VisionZip-om	25 (20-55)	27.6	44.4	58.3	59.1	33.7	65.4	34.7	49.3 96.3%
DivPrune	25 (12-100)	27.6	43.2	59.7	58.4	34.3	65.3	35.9	49.5 96.7%
DASH	25 (20-55)	27.6	44.3	58.7	59.6	34.3	66.2	35.5	49.8 97.3%
DivPrune-om	25 (20-55)	27.6	44.5	59.2	60.4	**34.5**	66.2	35.4	50.0 97.7%
SEATS	25 (20-55)	26.4	**45.1**	**60.9**	**62.5**	34.4	**66.3**	**36.0**	**50.9** 99.4%
FastV-om	15 (10-45)	24.3	38.4	51.8	51.0	31.2	58.8	31.2	43.7 85.4%
Random	15 (10-45)	18.0	40.4	55.7	57.2	32.7	63.4	32.8	47.0 91.8%
VisionZip-om	15 (10-45)	17.6	43.0	57.1	56.5	33.6	63.1	33.8	47.9 93.6%
DASH	15 (10-45)	18.0	42.9	56.2	57.3	33.7	64.2	34.1	48.1 93.9%
DivPrune-om	15 (10-45)	17.6	42.4	57.6	57.7	33.0	64.4	**34.4**	48.3 94.3%
SEATS	15 (10-45)	17.3	**43.9**	**58.7**	**60.2**	**34.2**	**65.9**	34.2	**49.5** 96.7%
Random	10 (6-35)	12.8	39.2	52.0	54.0	31.6	60.4	32.9	45.0 87.9%
VisionZip-om	10 (6-35)	12.8	41.8	52.8	53.8	33.4	60.7	34.4	46.2 90.2%
DivPrune-om	10 (6-35)	12.8	41.3	55.1	55.9	**33.5**	63.4	34.8	47.3 92.4%
SEATS	10 (6-35)	12.0	**43.4**	**57.2**	**58.5**	33.4	**63.8**	**35.4**	**48.6** 94.9%

Table 3: **Efficiency analysis on Qwen2.5-Omni-7B**. GPU usage, the time spans for token selection and prefill, and time-to-first-token (TTFT) are all measured on WorldSense using an A800 GPU.

Method	R	TFLOPs↓	GPU Peak Mem. (GB)↓	Token selection (sec.)↓		Prefill (sec.)↓	TTFT (sec.)↓	Mean↑ Score
				Pre-LLM	*Inner-LLM*			
Full tokens	100	111.0 (1.0×)	22.83	–	–	0.937 (1.0×)	1.471 (1.0×)	51.2 100.0%
FastV-om	35	38.3 (2.9×)	19.48	–	0.010	0.363 (2.6×)	0.901 (1.6×)	49.3 96.3%
VisionZip-om	35	37.4 (3.0×)	19.63	0.125	–	0.349 (2.7×)	1.066 (1.4×)	50.7 99.0%
DivPrune-om	35	37.4 (3.0×)	19.63	0.025	–	0.349 (2.7×)	0.952 (1.5×)	50.5 98.6%
OmniZip	35	38.2 (2.9×)	19.69	0.095	–	0.360 (2.6×)	1.088 (1.4×)	50.4 98.4%
DASH	35	37.8 (2.9×)	19.70	0.202	–	0.350 (2.7×)	1.102 (1.3×)	50.4 98.4%
SEATS	35	36.5 (3.0×)	18.70	0.034	0.083	0.424 (2.2×)	1.011 (1.5×)	**51.4** 100.4%
SEATS	25	26.4 (4.2×)	18.32	0.028	0.076	0.333 (2.8×)	0.916 (1.6×)	50.9 99.4%
SEATS	10	12.0 (**9.3**×)	**17.68**	0.020	0.059	0.192 (**4.9**×)	0.766 (**1.9**×)	48.6 94.9%

of SEATS remains well controlled across retention ratios. At 35% retention, SEATS achieves a 2.2× prefill speedup (1.5× in TTFT), while attaining the highest mean score (51.4 *vs*. 51.2 for full tokens). At 10% retention, the prefill speedup further increases to 4.9× (1.9× in TTFT) with GPU peak memory lowered to 17.68 GB, demonstrating a favorable efficiency-performance trade-off.

Results on Qwen3-Omni-30B. Qwen3-Omni-30B employs an MoE architecture with a deeper LLM and a DeepStack mechanism that injects multi-scale visual features into the early LLM layers.

Table 4: **Ablation studies**. Om-LLM: Qwen2.5-Omni-7B. Overall TRR (R): 35%.

#	Setup	WorldSense	Daily-Omni	AVUT	OmniVideoBench	Video-MME	LVOmniBench	Mean
0	Full-setup	**46.3**	**62.1**	**62.7**	**35.0**	**66.6**	**36.0**	**51.4**
Stage I: Pre-LLM Token Selection:								
1	`winDivPrune` → Random	45.5	61.4	61.9	34.4	66.4	34.8	50.7
2	`winDivPrune` → DivPrune-om	45.9	61.6	62.7	34.8	66.2	35.3	51.1
3	`winDivPrune` → VisionZip-om	45.3	61.8	62.6	34.3	66.3	35.3	51.0
4	`winDivPrune` → OmniZip	45.5	61.8	62.4	34.9	66.4	33.6	50.8
Stage II: Inner-LLM Token Selection:								
TRR Decay Schedule:								
5	Block-wise Decay → Single Decay	45.7	61.5	62.3	34.5	65.8	35.2	50.8
6	Linear Step → Equal Step	45.8	61.7	62.4	34.8	66.4	35.6	51.1
Token Budget Allocation:								
7	Global Token Selection	46.3	61.6	62.0	33.9	66.3	35.1	50.9
8	Modality Decouple	45.4	62.1	62.3	34.1	66.3	34.9	50.9
9	*w/o* Inter-window	46.2	61.5	62.4	34.3	66.4	35.4	51.0
10	*w/o* Intra-window	46.1	61.7	62.3	34.5	66.2	35.8	51.1

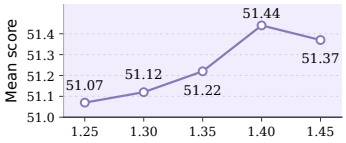


Figure 5: **Evaluating** λ.

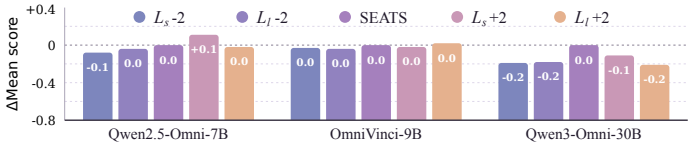


Figure 6: **Robustness to layer boundaries** across three om-LLMs.

Accordingly, SEATS synchronizes the pruning of these injected features. As reported in Tab. 5, our method achieves the best average performance across all retention ratios, preserving 99.8% of the full-token performance at 35% retention and 94.7% at the more aggressive 10% TRR. The results confirm the effectiveness of SEATS on a more complex MoE architecture.

Results on OmniVinci-9B. As shown in Tab. 6, SEATS consistently achieves the best average performance under different TRRs, indicating strong generalization across diverse om-LLMs.

5.3 Ablation Studies

Ablation studies are conducted on Qwen2.5-Omni with R=35%. See Tab. 4 and Figures 5 and 6.

`WinDivPrune` for pre-LLM token selection. Removing either attention weighting or diversity degrades performance (Setup-2 and Setup-3), confirming that the two are complementary. Replacing `winDivPrune` with random selection (Setup-1) or OmniZip (Setup-4) leads to a larger drop.

TRR decay schedule. Replacing our linear-step block-wise decay schedule with one-shot selection at a single layer (Setup-5) or an equal-step schedule (Setup-6) results in performance loss.

Top-down allocation for token budget. Replacing per-window selection with global ranking (Setup-7) lowers performance, confirming window-local selection better preserves temporal structure. Decoupling modality budgets (Setup-8), removing inter-window (Setup-9), or intra-window re-allocation (Setup-10) each degrades performance, verifying every level contributes.

Choice of λ. Fig. 5 shows that increasing λ generally improves performance, indicating that allocating more tokens to relevance-based inner-LLM compression is more effective than pre-LLM pruning.

Robustness to block boundaries. Fig. 6 reveals that shifting the transition block boundaries by ± 2 layers has minimal impact on the mean score, confirming robust boundary choices.

6 Conclusions

We introduce SEATS, a training-free method for efficient om-LLM inference. Extensive experiments on three omni-modal LLMs (Qwen2.5-Omni-7B, OmniVinci-9B, and Qwen3-Omni-30B) and six audio-visual benchmarks (WorldSense, DailyOmni, AVUT, OmniVideoBench, VideoMME and LVOmniBench) show that SEATS achieves a state-of-the-art efficiency-performance trade-off. SEATS serves as a plug-and-play module applicable to existing om-LLMs, enabling significant reductions in FLOPs, prefill latency, and memory consumption while preserving task performance.

AI USE STATEMENT

In this work, we used large language models (LLMs) to check for grammar errors, polish the fluency of our writing, and assist in organizing and presenting experimental results. No LLM was used for research ideation, experimental design, or forming conclusions. All substantive intellectual contributions are solely made by the authors. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work aims to improve the inference efficiency of omni-modal large language models, addressing a key barrier to deployment and scalability. By reducing computational cost and memory consumption, it broadens access to advanced audio-visual AI in resource-constrained settings. All benchmarks used in our experiments are publicly available. We do not foresee negative societal impacts beyond those inherent to the underlying om-LLMs.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide evaluation benchmark details in Appendix C.1, implementation and configuration of SEATS in Appendix C.2, and baseline reproduction details in Appendix C.3. Our source code and evaluation scripts are included in the supplementary materials.

REFERENCES

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. DivPrune: Diversity-based visual token pruning for large multimodal models. In *CVPR*, pp. 9392–9401, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

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 *ECCV*, 2024.

Jinhong Deng, Wen Li, Joey Tianyi Zhou, and Yang He. Scope: Saliency-coverage oriented token pruning for efficient multimodel LLMs. In *NeurIPS*, 2025.

Yue Ding, Yiyan Ji, Jungang Li, Xuyang Liu, Xinlong Chen, Junfei Wu, Bozhou Li, Bohan Zeng, Yang Shi, Yushuo Guan, et al. OmniSIFT: Modality-asymmetric token compression for efficient omni-modal large language models. *arXiv preprint arXiv:2602.04804*, 2026.

Sixun Dong, Juhua Hu, Mian Zhang, Ming Yin, Yanjie Fu, and Qi Qian. MMTok: Multimodal coverage maximization for efficient inference of VLMs. In *ICLR*, 2026.

Junhao Du, Jialong Xue, Anqi Li, Jincheng Dai, and Guo Lu. Unified spatiotemporal token compression for video-llms at ultra-low retention. In *CVPR*, 2026.

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 *ICLR*, 2026.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In *CVPR*, 2025.

Yuying Ge, Yixiao Ge, Chen Li, Teng Wang, Junfu Pu, Yizhuo Li, Lu Qiu, Jin Ma, Lisheng Duan, Xinyu Zuo, et al. Arc-hunyuan-video-7b: Structured video comprehension of real-world shorts. *arXiv preprint arXiv:2507.20939*, 2025.

Chao Gong, Depeng Wang, Zhipeng Wei, Ya Guo, Huijia Zhu, and Jingjing Chen. Echoingpixels: Aliasing-resistant joint token reduction for audio-visual llms. In *ICML*, 2026.

Jack Hong, Shilin Yan, Jiayin Cai, Xiaolong Jiang, Yao Hu, and Weidi Xie. WorldSense: Evaluating real-world omnimodal understanding for multimodal LLMs. In *ICLR*, 2026.

Xiaohu Huang, Hao Zhou, and Kai Han. PruneVid: Visual token pruning for efficient video large language models. In *Findings of ACL*, pp. 19959–19973, 2025.

Bingzhou Li and Tao Huang. DASH: Dynamic audio-driven semantic chunking for efficient omnimodal token compression. In *ECCV*, 2026.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024a.

Caorui Li, Yu Chen, Yiyan Ji, Jin Xu, Zhenyu Cui, Shihao Li, Yuanxing Zhang, Zhenghao Song, Dingling Zhang, Ying He, et al. Omnivideobench: Towards audio-visual understanding evaluation for omni mllms. In *ICLR*, pp. 138214–138236, 2026.

Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. LLaVA-NeXT-Interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*, 2024b.

Yadong Li, Jun Liu, Tao Zhang, Song Chen, Tianpeng Li, Zehuan Li, Lijun Liu, Lingfeng Ming, Guosheng Dong, Da Pan, et al. Baichuan-omni-1.5 technical report. *arXiv preprint arXiv:2501.15368*, 2025.

Xuyang Liu, Yiyu Wang, Junpeng Ma, and Linfeng Zhang. Video compression commander: Plug-and-play inference acceleration for video large language models. In *EMNLP*, pp. 1910–1924, 2025.

Qwen Team. Qwen3.5-Omni technical report. *arXiv preprint arXiv:2604.15804*, 2026.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In *ICCV*, pp. 22857–22867, 2025.

Kele Shao, Keda TAO, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. HoliTom: Holistic token merging for fast video large language models. In *NeurIPS*, 2025.

Kele Shao, Keda TAO, Kejia Zhang, Sicheng Feng, Mu Cai, Yuzhang Shang, Haoxuan You, Can Qin, Yang Sui, and Huan Wang. A survey of token compression for efficient multimodal large language models. *Transactions on Machine Learning Research*, 2026.

Leqi Shen, Guoqiang Gong, Tao He, Yifeng Zhang, Pengzhang Liu, Sicheng Zhao, et al. Fastvid: Dynamic density pruning for fast video large language models. In *NeurIPS*, pp. 123553–123581, 2025.

Jinsen Su, Yongdong Luo, Yuexiao Ma, Yibo Hu, Meiguang Jin, and Xiawu Zheng. Omniscope: Modality-decoupled token compression for omnimodal large language models. In *ACMMM*, 2026.

Changli Tang, Yixuan Li, Yudong Yang, Jimin Zhuang, Guangzhi Sun, Wei Li, Zejun Ma, and Chao Zhang. video-SALMONN 2: Caption-enhanced audio-visual large language models. *arXiv preprint arXiv:2506.15220*, 2025.

Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. DyCoke: Dynamic compression of tokens for fast video large language models. In *CVPR*, 2025.

Keda Tao, Kele Shao, Bohan Yu, Weiqiang Wang, Jian Liu, and Huan Wang. Omnizip: Audio-guided dynamic token compression for fast omnimodal large language models. In *CVPR*, pp. 17682–17692, 2026a.

Keda Tao, Yuhua Zheng, Jia Xu, Wenjie Du, Kele Shao, Hesong Wang, Xueyi Chen, Xin Jin, Junhan Zhu, Bohan Yu, et al. LVOmniBench: Pioneering long audio-video understanding evaluation for omnimodal LLMs. *arXiv preprint arXiv:2603.19217*, 2026b.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

Zining Wang, Zhihang Yuan, Yingjie Zhai, Wenshuo Li, Han Shu, Ruihao Gong, Jinyang Guo, and Xianglong Liu. Omnifit: Bridging modalities via layer-adaptive token compression for omnimodal large language models. In *ICML*, 2026.

Hao Wu, Yingqi Fan, Jinyang Dai, Junlong Tong, Yunpu Ma, and Xiaoyu Shen. HiDrop: Hierarchical vision token reduction in MLLMs via late injection, concave pyramid pruning, and early exit. In *ICLR*, 2026.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, et al. Conical visual concentration for efficient large vision-language models. In *CVPR*, pp. 14593–14603, 2025.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. Qwen2.5-Omni technical report. *arXiv preprint arXiv:2503.20215*, 2025a.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, et al. Qwen3-Omni technical report. *arXiv preprint arXiv:2509.17765*, 2025b.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. VisionZip: Longer is better but not necessary in vision language models. In *CVPR*, 2025a.

Yudong Yang, Jimin Zhuang, Guangzhi Sun, Changli Tang, Yixuan Li, Peihan Li, Yifan Jiang, Wei Li, Zejun Ma, and Chao Zhang. Audio-centric video understanding benchmark without text shortcut. In *EMNLP*, 2025b.

Linli Yao, Long Xing, Yang Shi, et al. Towards efficient multimodal large language models: A survey on token compression. *TechRxiv*, 2026. doi: 10.36227/techrxiv.176823010.07236701/v1.

Hanrong Ye, Chao-Han Huck Yang, Arushi Goel, Wei Huang, Ligeng Zhu, Yuanhang Su, Sean Lin, An-Chieh Cheng, Zhen Wan, Jinchuan Tian, et al. OmniVinci: Enhancing architecture and data for Omni-Modal understanding LLM. In *ICLR*, 2026.

Tianyu Yu, Zefan Wang, Chongyi Wang, Fuwei Huang, Wenshuo Ma, Zhihui He, Tianchi Cai, Weize Chen, Yuxiang Huang, Yuanqian Zhao, et al. MiniCPM-V 4.5: Cooking efficient MLLMs via architecture, data, and training recipe. *arXiv preprint arXiv:2509.18154*, 2025.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, et al. LMMs-Eval: Reality check on the evaluation of large multimodal models. In *Findings of NAACL*, pp. 881–916, 2025a.

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. In *ICCV*, pp. 20857–20867, 2025b.

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. In *NeurIPS*, 2025c.

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. In *ICML*, 2025d.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Llava-video: Video instruction tuning with synthetic data. *Transactions on Machine Learning Research*, 2025e.

Ziwei Zhou, Rui Wang, Zuxuan Wu, and Yu-Gang Jiang. Daily-omni: Towards audio-visual reasoning with temporal alignment across modalities. *arXiv preprint arXiv:2505.17862*, 2025.

Supplementary Material

CONTENTS

A Generality of the Block-wise Dependency Pattern **14**

B Additional Experimental Results **15**

B.1 Full Results on Qwen3-Omni-30B 15

B.2 Full Results on OmniVinci-9B 15

B.3 Detailed Analysis of λ 17

B.4 Detailed Robustness to Block Boundaries 17

C Additional Experimental Details **18**

C.1 Evaluation Benchmarks 18

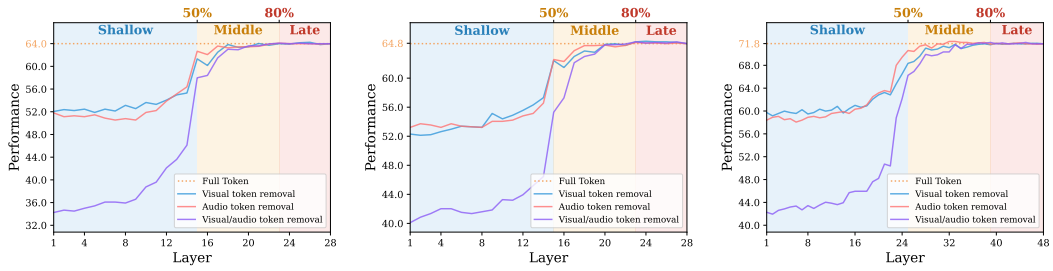
C.2 Reproduction Details of `SEATS` 18

C.3 Reproduction Details of Compared Baselines 20

D Limitations **21**

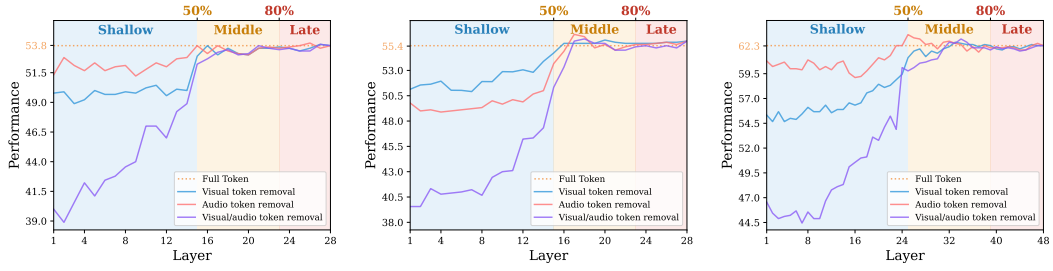
A GENERALITY OF THE BLOCK-WISE DEPENDENCY PATTERN

Sec. 3.2 reports the block-wise dependency pattern on WorldSense. To further demonstrate the generality of this pattern across video durations, we present results on Daily-Omni (short videos under one minute, Fig. 7) and the Video-MME Long subset (videos up to one hour, Fig. 8). Notably, the block-wise pattern of layer importance holds consistently across all three om-LLMs and datasets, confirming its generality beyond specific benchmarks or model architectures.



(a) Qwen2.5-Omni-7B (b) OmniVinci-9B (c) Qwen3-Omni-30B

Figure 7: **Layer-wise token removal on Daily-Omni, where videos are under one minute**.



(a) Qwen2.5-Omni-7B (b) OmniVinci-9B (c) Qwen3-Omni-30B

Figure 8: **Layer-wise token removal on Video-MME Long, where videos are up to one hour**.

Table 5: Comparison of different methods on Qwen3-Omni-30B-A3B-Instruct.

Method	R (R_v-R_a)	TFLOPs↓	World Sense	Daily Omni	AVUT	OmniVideo Bench	Video MME	LVOmni Bench	Mean
Full tokens	100 (100-100)	47.2	54.3	71.8	74.9	40.3	73.1	38.0	58.7 100.0%
Random	35 (32-70)	13.4	52.6	69.5	71.6	39.5	70.7	35.7	56.6 96.4%
FastV	35 (29-100)	13.7	51.5	68.6	71.2	40.0	69.5	37.1	56.3 95.9%
FastV-om	35 (32-70)	13.7	49.7	66.4	69.0	37.8	67.5	35.7	54.4 92.7%
VisionZip	35 (29-100)	13.4	53.0	71.4	72.7	40.3	70.6	36.6	57.4 97.8%
VisionZip-om	35 (32-70)	13.4	53.3	71.1	72.4	39.1	71.6	37.5	57.5 98.0%
DivPrune	35 (29-100)	13.4	53.4	69.8	73.2	39.6	70.7	36.6	57.2 97.4%
DivPrune-om	35 (32-70)	13.4	53.2	70.4	72.9	39.5	71.5	36.6	57.4 97.8%
DyCoke	35 (29-100)	13.4	52.9	68.6	72.0	40.1	70.3	36.6	56.8 96.8%
OmniZip	35 (32-70)	13.4	53.8	70.9	73.0	40.1	71.6	36.8	57.7 98.3%
DASH	35 (32-70)	13.4	53.2	70.2	73.0	39.2	71.7	37.4	57.4 97.8%
SEATS	35 (32-70)	13.3	**53.9**	**72.4**	**73.8**	**40.4**	**72.1**	**38.8**	**58.6** 99.8%
Random	25 (22-60)	9.4	51.3	68.1	70.3	39.0	70.4	35.6	55.8 95.1%
FastV	25 (18-100)	9.8	49.5	67.0	68.8	39.7	66.1	35.7	54.5 92.8%
FastV-om	25 (22-60)	9.8	47.2	62.9	65.3	38.1	65.7	33.7	52.2 88.9%
VisionZip	25 (18-100)	9.4	51.9	68.0	70.4	39.0	68.8	36.7	55.8 95.1%
VisionZip-om	25 (22-60)	9.4	51.8	69.8	72.0	39.8	71.2	36.4	56.8 96.8%
DivPrune	25 (18-100)	9.4	52.1	69.5	72.2	39.6	70.2	36.8	56.7 96.6%
DivPrune-om	25 (22-60)	9.4	52.2	69.0	71.9	40.0	70.9	36.2	56.7 96.6%
FastVID	25 (18-100)	9.1	51.4	68.3	69.7	40.2	68.0	36.5	55.7 94.9%
OmniZip	25 (25-25)	8.5	50.7	66.0	69.3	39.4	70.1	36.0	55.3 94.2%
DASH	25 (22-60)	9.4	52.4	68.7	71.9	40.3	71.5	36.8	56.9 96.9%
SEATS	25 (22-60)	9.0	**53.2**	**71.2**	**72.7**	**41.3**	**71.9**	**38.0**	**58.1** 99.0%
Random	15 (12-50)	5.8	49.8	64.6	65.7	36.8	67.9	33.9	53.1 90.5%
FastV-om	15 (12-50)	6.3	44.6	58.4	58.3	35.0	61.1	34.4	48.6 82.8%
VisionZip	15 (8-100)	5.8	49.8	65.4	67.4	39.1	65.6	35.9	53.9 91.8%
VisionZip-om	15 (12-50)	5.8	50.2	66.0	66.7	37.6	67.7	36.2	54.1 92.2%
DivPrune	15 (8-100)	5.8	51.0	67.8	69.5	38.0	66.8	35.2	54.7 93.2%
DivPrune-om	15 (12-50)	5.8	50.7	66.3	69.8	37.8	68.6	35.4	54.8 93.4%
DASH	15 (12-50)	5.8	51.0	65.6	69.3	38.8	69.3	35.8	55.0 93.7%
SEATS	15 (12-50)	5.5	**52.1**	**70.0**	**71.3**	**40.7**	**70.9**	**36.3**	**56.9** 96.9%
Random	10 (7-45)	4.0	48.3	61.2	63.4	36.6	65.3	34.2	51.5 87.7%
VisionZip-om	10 (7-45)	4.0	48.9	61.8	64.8	38.3	65.2	35.0	52.3 89.1%
DivPrune-om	10 (7-45)	4.0	49.3	64.0	67.1	38.7	67.2	35.3	53.6 91.3%
SEATS	10 (7-45)	3.9	**51.2**	**67.0**	**68.5**	**40.2**	**69.4**	**37.0**	**55.6** 94.7%

B Additional Experimental Results

B.1 Full Results on Qwen3-Omni-30B

As noted in Sec. 5.2, Qwen3-Omni-30B adopts an MoE architecture with a deeper LLM (L=48), and SEATS synchronizes pruning with its DeepStack feature injection accordingly. Tab. 5 presents the full results. The performance gap between SEATS and the best baseline widens as R decreases, *i.e.* 99.8% *vs.* 98.3% of the full-token performance at R=35% and 94.7% *vs.* 91.3% at R=10%, indicating that our multi-stage design is particularly effective under aggressive compression. The relative ranking of baselines is largely consistent with that on Qwen2.5-Omni, with FastV-om being the weakest and DivPrune-om and DASH the strongest competitors.

B.2 Full Results on OmniVinci-9B

OmniVinci-9B adopts a different backbone architecture and training strategy from the Qwen-Omni series. Tab. 6 presents the full results, complementing Sec. 5.2. SEATS achieves a mean score of 53.0 at R=35%, surpassing the full-token baseline (52.9), and retains 95.3% of the full-token performance at R=10%. Consistent with the trends on Qwen-Omni, SEATS ranks first across all retention ratios, with its advantage growing as R decreases. It is worth noting that the "-om" variants,

Table 6: Comparison of different methods on OmniVinci-9B.

Method	R (R_v-R_a)	TFLOPs↓	World Sense	Daily Omni	AVUT	OmniVideo Bench	Video MME	LVOmni Bench	Mean
Full tokens	100 (100-100)	68.5	49.3	64.7	65.1	34.6	67.1	36.8	52.9 100.0%
Random	35 (32-65)	29.7	47.0	60.7	60.0	33.6	67.3	34.6	50.5 95.5%
FastV	35 (28-100)	29.8	44.9	63.6	59.4	33.5	63.6	35.0	50.0 94.5%
FastV-om	35 (32-65)	29.8	47.2	64.1	61.7	34.3	66.3	35.9	51.6 97.5%
VisionZip	35 (28-100)	28.8	46.3	60.3	58.9	34.3	66.8	37.1	50.6 95.7%
VisionZip-om	35 (32-65)	29.4	47.6	62.8	62.3	34.3	69.0	37.2	52.2 98.7%
DivPrune	35 (28-100)	29.7	47.2	62.8	61.9	34.0	66.3	34.5	51.1 96.6%
DivPrune-om	35 (32-65)	29.4	47.8	62.5	62.3	34.2	68.2	37.3	52.1 98.5%
OmniZip	35 (32-65)	30.0	47.8	61.5	62.1	34.6	69.1	37.3	52.1 98.5%
DASH	35 (32-65)	29.7	47.9	62.7	61.9	34.4	67.9	**37.5**	52.1 98.5%
SEATS	35 (32-65)	28.6	**48.4**	**64.2**	**63.8**	**34.9**	**69.4**	37.4	**53.0** 100.2%
Random	25 (22-55)	23.1	46.5	57.3	57.7	34.1	66.0	34.8	49.4 93.4%
FastV	25 (17-100)	23.3	43.6	61.2	55.5	32.3	61.8	34.7	48.2 91.1%
FastV-om	25 (22-55)	23.3	45.5	61.9	59.2	34.6	64.3	35.6	50.2 94.9%
VisionZip	25 (17-100)	22.3	45.4	56.0	56.3	32.9	66.3	37.3	49.0 92.6%
VisionZip-om	25 (22-55)	22.9	46.9	62.0	60.2	34.4	68.7	37.9	51.7 97.7%
DivPrune	25 (17-100)	23.1	44.8	58.3	59.9	34.2	65.8	36.9	50.0 94.5%
DivPrune-om	25 (22-55)	23.4	47.1	62.2	59.6	33.7	66.9	37.5	51.2 96.8%
OmniZip	25 (25-25)	15.1	44.8	58.1	55.4	33.2	66.6	37.3	49.2 93.0%
DASH	25 (22-55)	23.1	47.3	60.7	60.6	34.4	66.5	**38.2**	51.3 97.0%
SEATS	25 (22-55)	22.0	**47.8**	**63.4**	**62.9**	**34.7**	**69.1**	38.1	**52.7** 99.6%
Random	15 (12-45)	16.7	44.2	54.3	53.0	34.0	64.4	35.0	47.5 89.8%
FastV-om	15 (12-45)	17.0	43.1	58.7	53.3	32.0	61.6	36.2	47.5 89.8%
VisionZip	15 (6-100)	16.4	43.1	53.3	54.3	33.1	64.0	37.6	47.6 90.0%
VisionZip-om	15 (12-45)	16.5	45.3	57.6	58.1	34.5	67.1	37.5	50.0 94.5%
DivPrune	15 (6-100)	16.7	44.6	57.1	57.2	34.4	64.3	37.9	49.3 93.2%
DivPrune-om	15 (12-45)	16.5	45.3	57.8	58.7	34.5	66.0	37.9	50.0 94.5%
DASH	15 (12-45)	16.7	45.7	57.1	56.3	**35.0**	66.2	**38.2**	49.8 94.1%
SEATS	15 (12-45)	16.2	**46.9**	**61.5**	**60.6**	34.7	**68.2**	38.0	**51.7** 97.7%
Random	10 (7-40)	13.6	42.9	50.2	52.7	32.9	63.3	35.3	46.2 87.3%
VisionZip-om	10 (7-40)	13.7	44.6	54.5	55.7	33.8	65.0	37.8	48.6 91.9%
DivPrune-om	10 (7-40)	13.7	44.1	55.8	56.3	33.4	65.4	37.6	48.8 92.2%
SEATS	10 (7-40)	13.0	**46.5**	**59.4**	**58.0**	**34.3**	**66.1**	**38.0**	**50.4** 95.3%

e.g. VisionZip-om and DivPrune-om, consistently outperform their visual-only counterparts on this model, suggesting that joint compression of both modalities is more effective on OmniVinci.

B.3 Detailed Analysis of λ

The scale factor λ controls the token budget split between the pre-LLM and inner-LLM stages. Specifically, the pre-LLM TRR is set to $\lambda \cdot R$ ($\lambda > 1$), so a larger λ retains more tokens at the pre-LLM stage and shifts more compression to the inner-LLM relevance-based selection. Fig. 5 shows the overall trend of λ w.r.t. the mean score on Qwen2.5-Omni at R=35%, and Tab. 7 reports the per-benchmark breakdown. As λ increases from 1.25 to 1.45, the mean score rises from 51.07 to a peak of 51.44 at λ=1.40, before slightly dropping to 51.37 at λ=1.45. We adopt λ=1.40 as the default, as it achieves the highest mean and yields a more balanced distribution across individual benchmarks.

Table 7: Effect of the scale hyperparameter λ.

Config	WorldSense	Daily-Omni	AVUT	OmniVideoBench	Video-MME	LVOmniBench	Mean
$\lambda = 1.25$	45.6	61.4	62.4	34.6	66.9	35.5	51.1
$\lambda = 1.30$	45.5	61.6	62.2	34.6	66.9	35.9	51.1
$\lambda = 1.35$	45.4	61.8	62.4	34.8	66.9	36.0	51.2
$\lambda = 1.40$	46.3	62.1	62.7	35.0	66.6	36.0	51.4
$\lambda = 1.45$	45.9	61.8	62.6	34.9	66.6	36.4	51.4

B.4 Detailed Robustness to Block Boundaries

Fig. 6 shows the change in mean score when shifting the block boundaries L_s or L_l by ± 2 layers. Tab. 8 further provides the per-benchmark breakdown. Across all three om-LLMs, ΔMean stays within 0.21 for every shifted configuration, and individual benchmark scores fluctuate only marginally. This confirms that the performance of SEATS is not sensitive to the exact positions of the transition layers. As such, a simple model-agnostic fixed depth-proportion strategy, *i.e.* placing the boundaries at fixed percentages of the LLM depth, suffices to achieve stable performance across different om-LLMs.

Table 8: Robustness to block boundaries at R=35%, with per-benchmark scores reported.

Config	Transition Layer $L_s - L_{m_1} - L_{m_2} - L_l$	World Sense	Daily Omni	AVUT	OmniVideo Bench	Video MME	LVOmni Bench	Mean	ΔMean
Qwen2.5-Omni-7B									
SEATS	$15-17-20-23$	46.3	62.1	62.7	35.0	66.6	36.0	51.44	–
L_s−2	$13-16-20-23$	45.7	62.6	62.5	34.9	66.7	35.9	51.36	−0.08
L_l−2	$15-17-19-21$	46.1	61.9	63.0	34.9	66.8	35.7	51.40	−0.04
L_s+2	$17-18-20-23$	46.3	62.1	62.9	35.2	66.9	36.0	51.55	+0.11
L_l+2	$15-17-21-25$	46.1	62.0	62.9	35.1	66.5	36.0	51.42	−0.02
OmniVinci-9B									
SEATS	$15-17-20-23$	48.4	64.2	63.8	34.9	69.4	37.4	53.01	–
L_s−2	$13-16-20-23$	48.3	64.4	63.4	35.3	69.1	37.4	52.98	−0.03
L_l−2	$15-17-19-21$	48.5	64.3	63.6	35.4	69.2	36.8	52.97	−0.04
L_s+2	$17-18-20-23$	48.4	64.7	63.7	35.1	69.3	36.9	52.99	−0.02
L_l+2	$15-17-21-25$	48.4	63.8	63.7	35.4	69.4	37.4	53.03	+0.02
Qwen3-Omni-30B									
SEATS	$25-29-34-39$	53.9	72.4	73.8	40.4	72.1	38.8	58.55	–
L_s−2	$23-28-34-39$	53.6	72.0	73.8	40.1	72.2	38.7	58.36	−0.19
L_l−2	$25-29-33-37$	53.7	72.0	73.7	39.9	72.2	38.8	58.37	−0.18
L_s+2	$27-30-34-39$	54.0	72.2	73.6	40.0	72.2	38.7	58.44	−0.11
L_l+2	$25-29-35-41$	53.9	72.0	73.8	40.1	72.1	38.6	58.34	−0.21

C Additional Experimental Details

C.1 Evaluation Benchmarks

The experiments are conducted on six widely used video understanding benchmarks, summarized in Tab. 9. They span diverse temporal scales and differ in their reliance on audio versus visual cues, enabling a thorough assessment of compression robustness. We describe each benchmark below.

Table 9: **Six benchmarks used in our experiments**.

Benchmark	**#Videos**	**#QA Pairs**	**Duration** (sec)
WorldSense (Hong et al., 2026)	1,662	3,172	140.7
Daily-Omni (Zhou et al., 2025)	684	1,197	43.2
AVUT (Yang et al., 2025b)	691	1,734	68.6
OmniVideoBench (Li et al., 2026)	628	1,000	409.0
Video-MME (Fu et al., 2025)	900	2,700	1021.3
LVOmniBench (Tao et al., 2026b)	275	1,014	2069.7

WorldSense (Hong et al., 2026) comprises 1,662 synchronized audio-visual videos spanning 8 domains, with 3,172 expert-annotated multiple-choice questions across 26 cognitive tasks. Its core design principle is tight audio-visual coupling: every question requires jointly integrating visual and audio evidence, and removing either modality leads to a drastic accuracy drop. The average video duration is approximately 141 seconds.

Daily-Omni (Zhou et al., 2025) collects 684 real-world YouTube videos of 30 to 60 seconds with 1,197 multiple-choice questions across six task families. Its distinguishing feature is temporal precision: answering requires pinpointing the correspondence between audio events and visual actions along the timeline, not global semantic matching.

AVUT (Yang et al., 2025b) targets audio-centric video understanding, defining eight task types that span audio content understanding and audio-visual alignment. An answer-permutation filter removes questions answerable from text alone. We use the *AV-Human* partition, which contains 691 expert-annotated videos with 1,734 multiple-choice QA pairs across 18 audio-centric domains, with an average duration of approximately 69 seconds.

OmniVideoBench (Li et al., 2026) comprises 628 videos ranging from several seconds to 30 minutes, with 1,000 manually annotated multiple-choice questions covering 13 task types. Each question is accompanied by a multi-step reasoning chain (5.68 steps on average), explicitly recording the modalities and evidence involved, making it well suited for diagnosing weak links in a model's reasoning pipeline.

Video-MME (Fu et al., 2025) contains 900 videos across 6 domains with durations from 11 seconds to 1 hour, divided into short, medium, and long tiers, yielding 2,700 human-annotated QA pairs. It supports subtitle and audio auxiliary inputs. We report results *without subtitles* to exclude external textual cues and isolate the effect of token compression.

LVOmniBench (Tao et al., 2026b) is, to our knowledge, the only benchmark dedicated to ultra-long audio-visual understanding. It curates 275 videos of 10 to 90 minutes (34.5 minutes on average, 140 hours in total) with 1,014 multiple-choice questions. Its average duration is 6 to 20 times longer than that of existing omni-modal benchmarks, specifically stressing multi-modal information retention and temporal localization over extended sequences.

C.2 Reproduction Details of SEATS

Implementation. For Qwen-Omni series, video frames are uniformly sampled at 2 FPS. Following (Tao et al., 2026a), each time window contains 288 visual tokens, along with 50 audio tokens for Qwen2.5-Omni-7B and 26 for Qwen3-Omni-30B. For Qwen2.5-Omni-7B, the maximum number of input frames is set to 128 for WorldSense, Daily-Omni and AVUT, 256 for OmniVideoBench, and 768 for Video-MME and LVOmniBench. As for Qwen3-Omni-30B, due to its larger memory consumption, the maximum number of input frames is set to 128 for the first three benchmarks

and 196 for the remaining three. For OmniVinci-9B, we follow the official configuration with 128 uniformly sampled frames for WorldSense, Daily-Omni, and AVUT, and 256 for the remaining three long benchmarks, with each frame producing 256 visual tokens and audio tokenized at 25 tokens per second. Unless otherwise specified, our hyperparameter setting is as follows: λ=1.4. For a fair comparison, we evaluate each method with the same R, chosen from $\{35\%, 25\%, 15\%, 10\%\}$. All experiments are conducted on NVIDIA A800 80GB GPUs using LMMs-Eval (Zhang et al., 2025a).

Decay schedule configuration. We detail the configuration of `SEATS` used to reproduce our results. Its inner-LLM block-wise decay schedule (Sec. 4.2.1) is specified by the pre-LLM scale factor λ, the four transition layers that delimit the shallow, middle, and late blocks, and the per-sub-block scale factor δ.

The four transition layers L_s, L_{m_1}, L_{m_2}, and L_l follow a *model-agnostic fixed depth-proportion* strategy, placed at 50%, 60%, 70%, and 80% of the LLM's total depth, respectively, *i.e.* $L_x = \lfloor \mathrm{L} \cdot p \rfloor + 1$ with $p \in \{0.5, 0.6, 0.7, 0.8\}$. The resulting transition layers for different om-LLMs are listed in Tab. 10.

Table 10: **Transition layers under the *model-agnostic depth-proportion* strategy**.

Transition Layer	**Qwen2.5-Omni-7B** (L=28)	**OmniVinci-9B** (L=28)	**Qwen3-Omni-30B** (L=48)
$L_s = \lfloor 0.5 \times \mathrm{L} \rfloor + 1$	15	15	25
$L_{m_1} = \lfloor 0.6 \times \mathrm{L} \rfloor + 1$	17	17	29
$L_{m_2} = \lfloor 0.7 \times \mathrm{L} \rfloor + 1$	20	20	34
$L_l = \lfloor 0.8 \times \mathrm{L} \rfloor + 1$	23	23	39

Given these boundaries, the scale factor δ is determined by the token budget. As described in Sec. 4.2.1, the overall `TRR` R equals the layer-count weighted average of the per-block `TRR`s. Since no token selection occurs in the shallow block (`TRR` $= r_s = \lambda R$) and all non-textual tokens are removed in the late block (`TRR` $= 0$), only the shallow and middle blocks contribute:

$$
\begin{aligned}
R &= \frac{1}{\mathrm{L}} \Big(\underbrace{\sum_{i=1}^{L_s} r_s}_{\text{shallow}} + \underbrace{\sum_{i=L_s+1}^{L_{m_1}} r_{m_1}}_{\text{middle sub-block 1}} + \underbrace{\sum_{i=L_{m_1}+1}^{L_{m_2}} r_{m_2}}_{\text{middle sub-block 2}} + \underbrace{\sum_{i=L_{m_2}+1}^{L_l} r_{m_3}}_{\text{middle sub-block 3}} \Big) \\
&= \frac{1}{\mathrm{L}} \Big(L_s\, r_s + (L_{m_1} - L_s)\, r_{m_1} + (L_{m_2} - L_{m_1})\, r_{m_2} + (L_l - L_{m_2})\, r_{m_3} \Big) \\
&= \frac{1}{\mathrm{L}} \Big(L_l\, \lambda R + \underbrace{\big(L_s + 2\, L_{m_1} + 3\, L_{m_2} - 6\, L_l \big)}_{\text{Constant } C} \cdot \delta \Big).
\end{aligned}
\tag{2}
$$

Solving for δ:

$$
\delta = \frac{(\mathrm{L} - L_l \lambda)\, R}{C}, \quad C = L_s + 2\, L_{m_1} + 3\, L_{m_2} - 6\, L_l. \tag{3}
$$

For the Qwen2.5-Omni-7B configuration (L=28, λ=1.4), we have $C = -29$. With R=0.3, this yields $\delta \approx 0.04345$.

Budget allocation procedure. As described in Sec. 4.2.2, the relevance scores $S_{t,v}$ and $S_{t,a}$ are obtained from the last text token's query-key attention to the visual and audio tokens within each window. Concretely, the per-head softmax-normalized attention weights are averaged across heads to yield per-token scores. These scores are then aggregated per window and normalized via a softmax over windows. In practice, the continuous budgets from Eq. (1) are rounded to integers. Any rounding residual is redistributed proportionally to S_t so that the combined total $\sum_t B_t = r_v N_v + r_a N_a$ is maintained, with each window's budget capped at its current token count. Within the middle block, each modality retains at least one token per window. Since each transition layer selects from the tokens retained by earlier layers, removed tokens are never restored and the capacity constraint is naturally satisfied.

C.3 Reproduction Details of Compared Baselines

Tab. 11 lists the per-modality retention ratios used in our experiments. As stated in Sec. 5.1, each 2-second window contains n_v=288 visual tokens and n_a audio tokens (n_a=50 for Qwen2.5-Omni, n_a=26 for Qwen3-Omni). For OmniVinci-9B, since it samples a fixed number of frames rather than at a fixed frame rate, the visual token count per window varies with video duration. Taking WorldSense (average 140 s, 128 frames) as reference, each window contains approximately n_v=468 visual tokens and n_a=50 audio tokens, so the per-modality ratios are approximate. The overall budget constraint is $R_v \cdot n_v + R_a \cdot n_a = R \cdot (n_v + n_a)$. Since $n_a < n_v$, audio tokens constitute a smaller portion of the total budget, making it natural to preserve them and apply selection only to visual tokens. We therefore evaluate two modes. Under the *Audio-intact* mode, R_a=100% and R_v is solved accordingly ("–" indicates cases where the resulting R_v is impractically low). Under the *Both-selected* mode, the budget is allocated to both modalities proportionally. In what follows, we detail how each baseline is adapted to the om-LLM setting and which mode it operates under:

Table 11: **Per-modality retention ratios (R_v-R_a) for each `TRR` (R).** *Audio-intact*: only visual tokens are selected (R_a=100%). *Both-selected*: token selection applied to both modalities.

TRR (R)	Qwen2.5-Omni-7B		OmniVinci-9B		Qwen3-Omni-30B	
	Audio-intact	*Both-selected*	*Audio-intact*	*Both-selected*	*Audio-intact*	*Both-selected*
35	24-100	30-65	28-100	32-65	29-100	32-70
25	12-100	20-55	17-100	22-55	18-100	22-60
15	–	10-45	6-100	12-45	8-100	12-50
10	–	6-35	–	7-40	–	7-45

• **FastV**[1] **(Chen et al., 2024) (ECCV 2024).** FastV prunes tokens at the K-th LLM layer using cross-modal attention scores, with a pruning ratio r. We follow the official setting with K=2. The original method targets only visual tokens, so we evaluate it under the *Audio-intact* mode. We also extend it to both visual and audio tokens, yielding FastV-om under the *Both-selected* mode. The pruning ratio r is set to match R_v and R_a in Tab. 11 for fair comparison.

• **VisionZip**[2] **(Yang et al., 2025a) (CVPR 2025).** VisionZip selects dominant tokens by encoder attention and merges the rest into contextual tokens. Since the original method operates at the encoder output, conflicting with pooling in Qwen-Omni, we apply compression after pooling instead (Shen et al., 2025). We retain dominant and contextual tokens at a ratio of $(R-0.05)$:0.05 per frame. The original method also targets only visual tokens, so we evaluate it under the *Audio-intact* mode. We further extend it to both modalities with window-level compression for audio tokens, yielding VisionZip-om under the *Both-selected* mode.

• **DivPrune**[3] **(Alvar et al., 2025) (CVPR 2025).** DivPrune retains a maximally diverse subset via greedy Max-Min cosine distance selection. The original method operates on pre-projector embeddings, which leads to noticeable degradation on om-LLMs. We therefore apply selection on post-projector embeddings after pooling instead. Selection is performed per frame for visual tokens. The original method targets only visual tokens, so we evaluate it under the *Audio-intact* mode. We further extend it to both modalities with window-level selection for audio tokens, yielding DivPrune-om under the *Both-selected* mode.

• **DyCoke**[4] **(Tao et al., 2025) (CVPR 2025).** DyCoke operates at both the prefill and decode stages. Following the evaluation protocol of (Tao et al., 2026a), we use only its prefill-stage TTM module, which partitions the video into 4-frame groups, keeps the first frame in each group intact, and merges temporally redundant visual tokens in the remaining frames based on inter-frame similarity. Since one frame is always preserved in each 4-frame window, the minimum video-token retention ratio R_v is 25%. As TTM targets only video tokens, we evaluate it under the *Audio-intact* mode, which limits the lowest feasible R to 35%.

[1] https://github.com/pkunlp-icler/FastV.
[2] https://github.com/dvlab-research/VisionZip, Apache 2.0 License.
[3] https://github.com/vbdi/divprune, CC BY-NC 4.0 License.
[4] https://github.com/KD-TAO/DyCoke, Apache 2.0 License.

• **FastVID**[5] **(Shen et al., 2025) (NeurIPS 2025).** FastVID prunes visual tokens via spatiotemporal DPC-kNN. It dynamically segments video tokens based on transition similarities, selects salient tokens per frame, and merges the remaining ones by spatiotemporal redundancy elimination. We follow the official hyperparameters: minimum segment count c=8, segment threshold τ=0.9, salient token ratio d=0.4, anchor frame step p=4, and merging factor α=0.6. As it targets only video tokens, we evaluate it under the *Audio-intact* mode.

• **OmniZip**[6] **(Tao et al., 2026a) (CVPR 2026).** OmniZip derives per-time-group retention scores from audio saliency to guide dynamic video token pruning, combined with interleaved spatiotemporal compression. The official implementation limits the per-group pruning ratio to 75%, resulting in a minimum R_v of 25%. To avoid constructing the full global audio attention matrix (over 30 GB GPU memory), the latest implementation uses chunked global attention on Qwen2.5-Omni (query chunked, key over the full audio sequence) and window-local attention on Qwen3-Omni. As OmniZip handles both modalities, we evaluate it under the *Both-selected* mode. At R=25%, R_v cannot be reduced below 25%, thus we set R_a=25% and gray out this entry. In addition, since OmniZip's audio branch merges 5% contextual tokens, we count both selected and merged tokens toward R_a to ensure a fair budget comparison.

• **DASH**[7] **(Li & Huang, 2026) (ECCV 2026).** DASH detects semantic boundaries from cosine-similarity drops between adjacent audio embeddings, replacing OmniZip's fixed temporal windows with variable-length segments and scoring tokens via a tri-signal fusion of boundary probability, uniqueness, and encoder attention. As an omni-modal compression method, we likewise evaluate it under the *Both-selected* mode. In the official implementation, projected video segment boundaries are not aligned to complete frame windows, so tail tokens shorter than one frame are discarded, causing the actual R_v to fall slightly below the configured value. We omit R=10% as the resulting R_v is impractically low.

All methods are evaluated using LMMs-Eval[8] (Zhang et al., 2025a) for consistency. The base omni-modal large language models used are Qwen2.5-Omni-7B[9] (Xu et al., 2025a), OmniVinci-9B[10] (Ye et al., 2026) and Qwen3-Omni-30B-A3B-Instruct[11] (Xu et al., 2025b).

D LIMITATIONS

Although our method SEATS allocates token budgets in an input-dependent manner via query relevance scores, the model-agnostic fixed depth-proportion decay schedule remains fixed for the three om-LLMs in our paper. Exploring sample-adaptive compression warrants further investigation. Our current method operates after the encoders and does not reduce the computational cost of visual and audio encoding itself. Moreover, SEATS targets single-turn QA over complete audio-visual inputs. Incremental budget allocation for streaming inputs and cross-turn evidence reuse in multi-turn interactions are directions worthy of future investigation.

[5] https://github.com/LunarShen/FastVID, MIT License.
[6] https://github.com/KD-TAO/OmniZip, Apache 2.0 License.
[7] https://github.com/laychou666/DASH, Apache 2.0 License.
[8] https://github.com/EvolvingLMMs-Lab/lmms-eval, Apache 2.0 License.
[9] https://huggingface.co/Qwen/Qwen2.5-Omni-7B, Apache 2.0 License.
[10] https://huggingface.co/nvidia/omnivinci, Apache 2.0 License.
[11] https://huggingface.co/Qwen/Qwen3-Omni-30B-A3B-Instruct, Apache 2.0 License.