PACK, DON'T DROP: BREAKING THE SELECTION BOTTLENECK IN VISUAL TOKEN PRUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual token pruning reduces the inference cost of multimodal large language models (MLLMs), yet under strict token budgets, selection alone inevitably discards complementary visual evidence. Existing aggregation methods reuse information from pruned tokens within compressed sequences, providing a complementary direction beyond selection. Under a fixed token budget, effective information reuse therefore requires both appropriate routing of pruned content and suitable representations at the retained positions. To this end, we propose **PackAdapt**, which enhances a given set of retained positions without changing the final token budget. Its Information Folding module learns task-relevant routing from pruned tokens to retained tokens on top of semantic similarity, while its Representation Adaptation module further transforms the folded features through a lightweight residual mapping. With the base MLLM frozen, the selector, routing module, and representation adapter are jointly trained with the language-modeling objective, so supervision from the final prediction reaches all three components. Extensive experiments across multiple multimodal backbones, image-understanding benchmarks, and video-understanding benchmarks demonstrate that PackAdapt consistently improves compressed-model performance under different token budgets, with particularly strong gains under aggressive visual compression.

1 INTRODUCTION

Multimodal large language models (MLLMs) encode images into visual tokens that are jointly processed with text by the language model (Alayrac et al., 2022; Li et al., 2023b; Liu et al., 2023). As input resolution increases and multi-image and video scenarios become more common, the number of visual tokens grows rapidly, leading to substantial computation, memory, and KV-cache overhead (Bai et al., 2025; Li et al., 2024a).Visual token compression has therefore become an important approach to improving MLLM inference efficiency, with the central goal of preserving task-relevant visual information under restrictive token budgets.

Most existing methods formulate visual token compression as a selection problem. FastV, SparseVLM, DART, and DivPrune estimate token importance based on attention, cross-modal relevance, or feature diversity (Chen et al., 2024; Zhang et al., 2025b; Wen et al., 2025; Alvar et al., 2025). However, such predefined criteria remain proxies for the final task objective. Recent methods further make token selection learnable(Rao et al., 2021). For example, VisionSelector (Zhu et al., 2026) combines a lightweight scoring network with differentiable Top-K selection to learn retained positions directly from downstream language-modeling supervision, while IDPruner (Tan et al., 2026) further incorporates both importance and diversity. As illustrated in Fig. 1(a)(b), these methods improve the quality of selection, but still share the same basic operation: selected tokens are retained, while all remaining tokens are discarded.

This creates an inherent limitation of selection-based compression. A selector can determine which positions deserve the limited token budget, but it cannot reuse complementary information contained in positions that are not selected (Ryoo et al., 2021; Bolya et al., 2023). A token may not deserve to occupy an entire retained position while still containing useful local text, attributes, or spatial relations. As the retention ratio decreases, more visual positions must be removed, making such information loss increasingly difficult to avoid through better selection alone. This motivates a

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

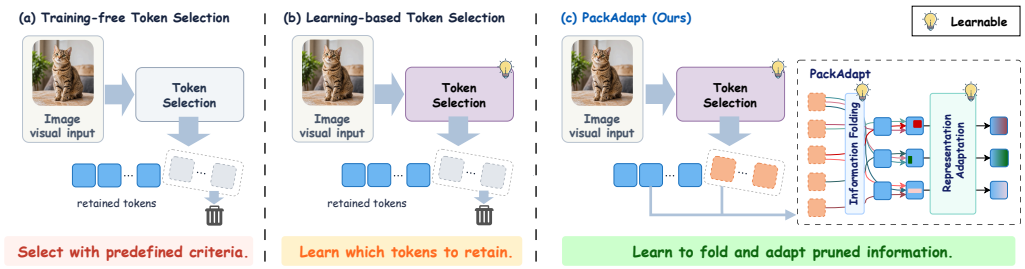

Figure 1: Comparison of three visual token compression paradigms. (a) Selection only directly discards information from pruned tokens. (b) Learnable selection improves which tokens are retained under downstream supervision, but still discards unselected information. (c) PackAdapt reuses pruned information and further adapts the enriched retained representations.

further question: **can useful information from pruned tokens be preserved without increasing the final token count?**

One existing strategy aggregates pruned tokens into a small number of contextual tokens and concatenates them with the primary retained tokens (Yang et al., 2025). Under a fixed budget of K tokens, however, these aggregated tokens also consume final positions, leaving fewer positions for individually retained important tokens. An alternative is to directly merge pruned information into existing retained tokens (Shang et al., 2025; Jiang et al., 2025; Liang et al., 2022). This reuses discarded information without increasing sequence length, allowing each retained token to preserve its original semantics while carrying complementary evidence (Chen et al., 2026).

However, placing information into a retained token does not necessarily mean that the information has been effectively encoded (Kim et al., 2024). If compression stops at predefined weighted aggregation, the resulting representation remains largely determined by a static combination of the input features. Such direct aggregation lacks task-driven guidance on exactly which pruned information should be routed to which retained position. In particular, when the base MLLM is frozen, the aggregated representation itself cannot be further adjusted toward the final prediction objective (Li et al., 2023b; Gao et al., 2023). In other words, simple aggregation puts information into retained tokens, but does not explicitly learn how to optimally allocate this complementary content or how the resulting dense mixed information should be represented.

To address these limitations, we propose **PackAdapt**, as illustrated in Fig. 1(c). PackAdapt aims to preserve more useful information within a fixed token budget by jointly optimizing information reuse and representation refinement. Specifically, **Information Folding** transfers complementary information from pruned tokens into the retained positions selected by the selector. Instead of relying solely on semantic similarity, it introduces a learnable correction term to better align the routing process with the downstream objective. After folding, **Representation Adaptation** refines the resulting representations with a lightweight residual mapping and scale anchoring, enabling the enriched tokens to be better interpreted by the frozen MLLM. The selector, Information Folding, and Representation Adaptation are jointly trained with a unified language-modeling objective, while the parameters of the base MLLM remain unchanged.

PackAdapt introduces no additional visual tokens and therefore preserves the same language-model input length and KV-cache size as direct pruning at the same retention ratio. Experiments across multiple multimodal backbones and image- and video-understanding benchmarks show consistent improvements under different token budgets, with larger gains under more aggressive compression. On Qwen2.5-VL-7B, PackAdapt retains 95.64% of the original model performance with only 20% of the visual tokens, while reducing prefill latency from 843,ms to 344,ms at a 10% retention ratio.

Our main contributions are as follows:

- We identify a limitation of optimizing token selection alone: complementary information in pruned positions remains unused. We therefore consider how retained representations can carry more useful information under a fixed token budget.

- We propose **PackAdapt**, which combines task-aware **Information Folding** for reusing pruned information with lightweight **Representation Adaptation** for refining the resulting dense representations. Both modules are jointly trained with the selector while keeping the base MLLM frozen and the final visual token count unchanged.
- We validate PackAdapt across multiple multimodal backbones, as well as image and video understanding benchmarks. PackAdapt consistently improves compressed-model performance across different retention ratios, while preserving the inference-efficiency benefits of visual token pruning.

2 Related Work

2.1 Training-free Token Pruning Methods

Training-free methods reduce visual redundancy without additional training, typically using attention, cross-modal relevance, or feature geometry to determine which tokens to retain. FastV (Chen et al., 2024) exploits attention scores, SparseVLM (Zhang et al., 2025b) incorporates text-guided relevance, while DART (Wen et al., 2025) and DivPrune (Alvar et al., 2025) emphasize redundancy reduction and feature diversity. Beyond direct pruning, VisionZip (Yang et al., 2025) preserves pruned information through contextual token aggregation, while LLaVA-PruMerge (Shang et al., 2025) and VISA (Jiang et al., 2025) merge or propagate removed content into retained representations. These methods provide efficient plug-and-play compression, but their token selection and information transfer are determined by predefined criteria rather than downstream optimization.

2.2 Learning-based Token Selection Methods

Learning-based methods directly optimize visual token reduction with task supervision. Dynamic-LLaVA (Huang et al., 2025) learns predictors for dynamic vision-language context sparsification, while ATP-LLaVA (Ye et al., 2025) introduces learnable pruning thresholds for instance- and layer-adaptive token reduction. VisionSelector (Zhu et al., 2026) further combines a lightweight scorer with differentiable Top-K selection to learn retained positions under language-modeling supervision, and IDPruner (Tan et al., 2026) balances learned importance with semantic diversity. These approaches primarily improve which tokens are retained. PackAdapt builds on this paradigm but additionally learns how information from pruned tokens is folded into retained positions and how the resulting representations are adapted for downstream prediction.

3 Method

3.1 Problem Setup and Overview

Given a visual input I and a textual instruction q, an MLLM converts I into a sequence of visual tokens $X = [x_1, \ldots, x_N]^\top \in \mathbb{R}^{N \times d}$ through its visual encoder and modality interface, where N denotes the number of visual tokens and d is the feature dimension at the language-model interface. Given a retention ratio $b \in (0, 1]$, visual token compression reduces the sequence to $K = \max(1, \lfloor bN \rfloor)$ tokens while preserving the model's response quality for instruction q. We perform compression after the modality interface and before the visual tokens enter the language model. For Qwen2.5-VL, this position corresponds to the output of the patch merger.

We reuse VisionSelector (Zhu et al., 2026) as the token selection mechanism. Its importance scorer f_ϕ assigns a score to each visual token, and hard Top-K selection determines the retained set $\mathcal{S} \subseteq \{1, \ldots, N\}$ with $|\mathcal{S}| = K$. The remaining positions form the pruned set $\mathcal{D} = \{1, \ldots, N\} \setminus \mathcal{S}$. We further define $h_i = 1$ for $i \in \mathcal{S}$ and $h_i = 0$ otherwise.

Direct pruning simply forwards $X_\mathcal{S}$ to the language model. PackAdapt instead enhances these retained positions using information from the pruned tokens while keeping the final token budget unchanged. Information Folding transfers complementary content from $\mathcal{D}$ to $\mathcal{S}$, after which Representation Adaptation updates the folded features before they enter the frozen language model. The final output is denoted by $Y_\mathcal{S} \in \mathbb{R}^{K \times d}$.

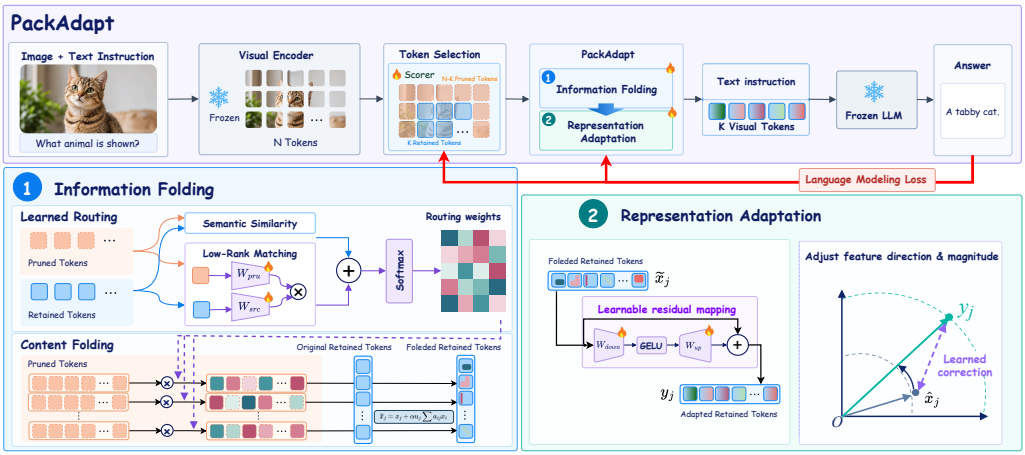

Figure 2: **Overview of PackAdapt.** Given selector-determined retained positions, Information Folding routes complementary content from pruned tokens into the retained set, after which Representation Adaptation further refines the folded features before they enter the frozen language model. PackAdapt enhances the retained representations without increasing the final visual token budget.

3.2 Information Folding

Token selection determines which positions are retained, but complementary evidence may remain in the pruned tokens. Information Folding transfers this information into the fixed retained set without introducing additional visual positions. It combines semantic similarity with a learnable task-aware correction to determine routing weights, followed by residual aggregation at the retained positions.

Task-aware routing. We use cosine similarity between the original visual features as a semantic routing prior. Let $z_i = x_i / \max(\|x_i\|_2, \epsilon)$, where $\epsilon > 0$ is a small constant for numerical stability. For each pruned token $i \in \mathcal{D}$ and retained token $j \in \mathcal{S}$, the routing score is

$$\ell_{ij} = z_i^\top z_j + \frac{\gamma}{\sqrt{r_f}} \left(W_{\mathrm{src}} z_i\right)^\top \left(W_{\mathrm{dst}} z_j\right), \qquad i \in \mathcal{D},\ j \in \mathcal{S}. \tag{1}$$

Here, $W_{\mathrm{src}}, W_{\mathrm{dst}} \in \mathbb{R}^{r_f \times d}$ are low-rank projections for the source and destination roles, respectively, $r_f \ll d$ is the routing rank, and γ controls the strength of the learnable correction. The cosine term preserves semantic affinity in the original feature space, while the low-rank term allows downstream supervision to correct routing relations that cannot be captured by similarity alone. Separate source and destination projections further model their asymmetric roles in information transfer.

Information allocation. For each pruned token, we obtain routing weights by applying a softmax operation over the retained set:

$$a_{ij} = \mathrm{softmax}_{j \in \mathcal{S}}(\ell_{ij}/\tau), \qquad i \in \mathcal{D}, \tag{2}$$

where $\tau > 0$ is a learnable temperature. A lower temperature yields more concentrated routing, whereas a higher temperature distributes information across more retained positions. Normalizing over $\mathcal{S}$ constrains each pruned token to redistribute its information only within the fixed retained set.

Residual information folding. Each retained token collects complementary information from the pruned set and incorporates it through a residual update:

$$\widetilde{x}_j = x_j + \alpha \sum_{i \in \mathcal{D}} a_{ij} x_i, \qquad j \in \mathcal{S}, \tag{3}$$

where α controls the folding strength and a_{ij} denotes the routing weight from pruned token i to retained token j. The normalized features z_i and z_j determine the routing weights, whereas the original interface features x_i carry the transferred content. Thus, the original retained token remains the representational basis while complementary information is incorporated through residual aggregation.

3.3 Representation Adaptation

Information Folding enriches the retained representations through residual aggregation, which can change their magnitude and direction by construction. Rather than treating the folded features as the final representations, we further allow them to be transformed under downstream supervision before entering the frozen language model. We implement this with scale anchoring followed by a lightweight learnable residual mapping.

For each $j \in \mathcal{S}$, we first anchor the magnitude of the folded representation to that of the original retained token, and then apply a bottleneck residual mapping:

$$\widehat{x}_j = \widetilde{x}_j \frac{\|x_j\|_2}{\max(\|\widetilde{x}_j\|_2, \epsilon)}, \qquad y_j = \widehat{x}_j + W_{\text{up}}\,\text{GELU}\left(W_{\text{down}}\widehat{x}_j + b_{\text{down}}\right). \tag{4}$$

Here, $W_{\text{down}} \in \mathbb{R}^{r_a \times d}$, $W_{\text{up}} \in \mathbb{R}^{d \times r_a}$, $b_{\text{down}} \in \mathbb{R}^{r_a}$, and $r_a \ll d$ is the adapter rank. Scale anchoring keeps the magnitude referenced to the original retained token while preserving the directional information introduced by folding. The residual adapter then learns task-dependent adjustments to both the direction and scale of the folded representation.

For compactness, we denote the composition above by $y_j = \mathcal{A}_\psi(\widetilde{x}_j; x_j)$, where x_j provides the scale reference and ψ denotes the parameters of Representation Adaptation. The composition preserves the output shape $K \times d$, so PackAdapt introduces no additional visual positions.

3.4 Joint Optimization

We jointly train the selector, Information Folding, and Representation Adaptation while keeping the base multimodal model frozen. Since hard Top-K selection is non-differentiable, we follow VisionSelector and combine hard selection for the forward routing structure with continuous selection weights for gradient propagation.

Differentiable selection and folding. The selector scores are converted through differentiable Top-K into continuous weights $m_i \in [0, 1]$ satisfying $\sum_{i=1}^{N} m_i = K$. In parallel, hard Top-K selection on the same scores determines $\mathcal{S}$ and $\mathcal{D}$. The hard sets define which positions act as information sources and destinations, whereas the continuous weights provide a differentiable path to the selector.

During training, all N visual positions remain in the sequence as a differentiable surrogate. For each retained position, we weight the contribution of pruned token i by $1 - m_i$:

$$v_j^{\text{train}} = x_j + \alpha \sum_{i \in \mathcal{D}} a_{ij}(1 - m_i)x_i, \qquad j \in \mathcal{S}. \tag{5}$$

For $j \in \mathcal{D}$, we simply set $v_j^{\text{train}} = x_j$. Thus, tokens with lower selection weights contribute more strongly as information sources, coupling token selection with the folding process.

We apply the same Representation Adaptation to the hard-retained positions, setting $\bar{v}_j^{\text{train}} = \mathcal{A}_\psi(v_j^{\text{train}}; x_j)$ for $j \in \mathcal{S}$ and $\bar{v}_j^{\text{train}} = x_j$ for $j \in \mathcal{D}$. The resulting features are modulated by the continuous selection weights as $o_j^{\text{train}} = m_j \bar{v}_j^{\text{train}}$, and the masked visual sequence is denoted by $O_{\phi,\psi}^{\text{train}} = [o_1^{\text{train}}, \ldots, o_N^{\text{train}}]^\top$. This surrogate preserves a gradient path through token selection while allowing downstream supervision to reach both the routing and adaptation modules.

At inference time, differentiable masking is removed. Hard Top-K determines the retained set, Information Folding transfers information from $\mathcal{D}$ to $\mathcal{S}$, and Representation Adaptation produces the final representations $Y_{\mathcal{S}}$. The pruned positions are then physically removed, so only the K adapted retained tokens are passed to the language model.

Unified language-modeling supervision. Let Θ denote the frozen base-model parameters, ϕ the selector parameters, and ψ the parameters of Information Folding and Representation Adaptation. Given instruction q and target response $w = (w_1, \ldots, w_T)$, we optimize

$$\mathcal{L}(\phi, \psi) = -\sum_{t=1}^{T} \log p_\Theta\left(w_t \mid q, O_{\phi,\psi}^{\text{train}}, w_{<t}\right) + \lambda\,\text{BCE}\left(m, \text{sg}(h)\right). \tag{6}$$

Here, $\mathrm{sg}(\cdot)$ denotes stop-gradient. The language-modeling term provides the primary task supervision and propagates downstream prediction errors through the selector, information routing, and representation adaptation. Therefore, the routing module receives task supervision in addition to the visual-similarity prior, with gradients propagated through the adapted representations. The BCE term serves as a selection-consistency regularizer that aligns the continuous selection weights with the corresponding hard Top-K decisions.

Throughout training, the base multimodal model remains frozen; only the selector, routing projections, learnable temperature, and representation adapter are optimized. We initialize W_{dst} and W_{up} to zero so that the learnable routing correction and residual adapter are initially inactive, and gradually increase α from zero to its target value to introduce folded information progressively.

4 EXPERIMENTS

We evaluate PackAdapt under different visual token budgets using Qwen2.5-VL-7B-Instruct (Bai et al., 2025) and LLaVA-OneVision-1.5-8B (An et al., 2025). We further examine its key components through ablation studies, hyperparameter sensitivity analysis, and efficiency evaluation on Qwen2.5-VL-7B-Instruct at a 10% retention ratio. We describe the visual token budget using the retention ratio $b = K/N$, corresponding to a compression ratio of $1 - b$.

4.1 EXPERIMENTAL SETTINGS

Benchmarks and metrics. We evaluate PackAdapt on eight image-understanding benchmarks. OCRBench (Liu et al., 2024), DocVQA (Mathew et al., 2021), and TextVQA (Singh et al., 2019) are treated as OCR-related benchmarks, while ChartQA (Masry et al., 2022), ScienceQA (Lu et al., 2022), AI2D (Kembhavi et al., 2016), MMMU (Yue et al., 2024), and POPE (Li et al., 2023c) cover general visual understanding and reasoning. For OCRBench, DocVQA, ChartQA, TextVQA, ScienceQA, AI2D, MMMU, and POPE, we report Score (out of 1,000), ANLS, Relaxed Accuracy, Exact Match, Exact Match, Exact Match, Accuracy, and F1, respectively. For video evaluation, we use SEED-Bench Video (Li et al., 2023a) and MVBench (Li et al., 2024b), and report the corresponding accuracy scores following their official evaluation protocols.

To summarize performance across benchmarks with different metric scales, we normalize each score by the corresponding Origin score of the same backbone and compute their arithmetic mean, denoted by Avg. Here, Origin denotes the uncompressed model evaluated with the complete visual input. The image and video results are averaged over eight and two metrics, respectively. Origin is therefore normalized to 100%, while the Avg of a compressed model measures the average relative performance retained after visual compression.

Baselines. We compare PackAdapt with FastV (Chen et al., 2024), VisionZip (Yang et al., 2025), DART (Wen et al., 2025), DivPrune (Alvar et al., 2025), VisionSelector (Zhu et al., 2026), and IDPruner (Tan et al., 2026), covering importance-based selection, diversity-aware pruning, and aggregation-based compression strategies. We report the uncompressed backbone with the complete visual input as Origin. VisionSelector provides the scoring network and selection mechanism reused by PackAdapt; comparison against it therefore isolates the benefit of enhancing retained representations beyond token selection. For both Qwen2.5-VL-7B-Instruct and LLaVA-OneVision-1.5-8B, we report results at retention ratios of 20% and 10%.

Implementation details. Qwen2.5-VL and LLaVA-OneVision use dynamic-resolution inputs, with pixel budgets ranging from 200,704 to 1,605,632; video experiments uniformly sample up to 32 frames. Following VisionSelector (Zhu et al., 2026), we train on approximately 144K examples from Cambrian-737K (Tong et al., 2024) while keeping the base MLLM frozen and optimizing only the selector and PackAdapt modules. We use AdamW (Loshchilov & Hutter, 2019) with a learning rate of 5×10^{-5} for one epoch. PackAdapt contains only 15.6M trainable parameters and completes training in approximately 50 minutes on four NVIDIA H20 GPUs, resulting in a lightweight and cost-efficient training footprint. The adapter rank r_a, routing rank r_f, and folding strength α are set to 128, 256, and 0.6, respectively. All reported results are averaged over three runs with different random seeds. Additional training and implementation details are provided in Appendix A.1.

4.2 MAIN RESULTS

Image-understanding performance. Tables 1 and 2 report image-understanding results on Qwen2.5-VL-7B-Instruct and LLaVA-OneVision-1.5-8B, respectively. We evaluate both moderate compression at a 20% retention ratio and more aggressive compression at a 10% retention ratio. Within each retention-ratio block, boldface denotes the best result and underlining denotes the second-best result among the compared compression methods.

On Qwen2.5-VL, PackAdapt achieves the highest normalized average under both token budgets, outperforming the strongest competing baseline by 1.48 and 3.26 percentage points at 20% and 10% retention, respectively. The larger margin under the tighter budget highlights the benefit of preserving and adapting pruned information when only a small number of visual tokens can be retained.

A similar trend is observed on LLaVA-OneVision, where PackAdapt achieves the highest Avg-8 at both retention ratios. Its margin over the strongest competing baseline increases from 0.71 percentage points at 20% retention to 1.61 points at 10%, further showing that PackAdapt is particularly effective under aggressive compression.

Table 1: Image-understanding results on Qwen2.5-VL-7B.

Method	**OCRBench** Score	**DocVQA** ANLS	**ChartQA** Relaxed	**TextVQA** EM	**ScienceQA** EM	**AI2D** EM	**MMMU** Acc	**POPE** F1	**Avg**-8
Origin (100%)									
Qwen2.5-VL-7B	838	94.33	83.40	82.84	87.26	93.59	50.78	86.19	100.00%
Retain 20% Tokens (80% Compression Ratio)									
FastV	597	75.99	60.48	78.01	82.75	82.35	49.00	76.12	85.76%
VisionZip	591	74.75	62.04	72.03	84.68	82.32	47.00	83.23	85.65%
DART	537	50.03	47.16	67.53	83.34	71.04	47.00	80.16	76.52%
DivPrune	648	73.81	57.88	73.86	84.33	82.29	46.33	83.55	85.86%
VisionSelector	**758**	88.03	68.92	**80.49**	85.18	90.15	47.56	82.88	93.42%
IDPruner	741	87.95	71.20	80.19	84.98	**90.64**	**49.44**	84.35	94.16%
PackAdapt (Ours)	748	**89.74**	**79.24**	**80.49**	**85.78**	90.25	48.44	**84.84**	**95.64%**
Retain 10% Tokens (90% Compression Ratio)									
FastV	440	58.64	44.64	70.83	81.95	75.74	45.78	65.99	74.41%
VisionZip	404	48.29	42.84	55.94	82.60	73.09	45.44	78.46	71.45%
DART	415	33.13	34.00	53.97	81.85	67.10	46.44	71.91	66.37%
DivPrune	477	54.04	39.80	64.65	82.15	72.80	46.00	79.27	74.31%
VisionSelector	627	72.01	61.04	76.23	82.65	84.47	47.33	77.47	85.55%
IDPruner	616	74.08	60.48	74.92	83.64	84.39	46.56	81.43	85.90%
PackAdapt (Ours)	**643**	**76.79**	**68.88**	**77.23**	**85.03**	**84.84**	**47.78**	**83.75**	**89.16%**

Table 2: Image-understanding results on LLaVA-OneVision-1.5-8B.

Method	**OCRBench** Score	**DocVQA** ANLS	**ChartQA** Relaxed	**TextVQA** EM	**ScienceQA** EM	**AI2D** EM	**MMMU** Acc	**POPE** F1	**Avg**-8
Origin (100%)									
LLaVA-OneVision-1.5-8B	830	97.87	86.72	79.51	98.56	94.01	56.44	88.46	100.00%
Retain 20% Tokens (80% Compression Ratio)									
FastV	489	76.64	59.08	67.33	87.95	78.89	51.33	67.08	78.75%
VisionZip	441	70.71	67.56	66.28	91.87	79.18	51.44	84.61	81.36%
DART	574	64.65	67.28	72.77	**95.44**	79.57	54.44	85.08	84.80%
DivPrune	556	75.90	58.64	72.73	92.46	84.26	52.56	**87.17**	84.84%
VisionSelector	675	84.86	73.36	75.99	94.40	**89.12**	**55.11**	78.75	90.68%
IDPruner	652	78.20	70.88	75.97	94.50	87.66	54.67	83.01	89.45%
PackAdapt (Ours)	**682**	**85.13**	**73.64**	**76.37**	94.74	88.50	54.44	83.38	**91.39%**
Retain 10% Tokens (90% Compression Ratio)									
FastV	374	54.43	38.08	56.47	81.51	73.15	49.56	62.89	66.88%
VisionZip	256	39.67	43.12	47.17	88.70	71.83	51.11	79.01	65.84%
DART	392	40.61	47.32	63.56	89.34	73.51	50.67	75.53	70.90%
DivPrune	419	56.48	38.24	65.48	88.20	76.85	51.33	**83.92**	73.96%
VisionSelector	521	65.07	**61.00**	70.67	90.38	82.67	51.33	75.46	80.55%
IDPruner	528	57.45	55.32	71.07	90.18	80.21	51.78	80.42	79.37%
PackAdapt (Ours)	**544**	**66.35**	58.72	**72.01**	**91.22**	**83.52**	**52.11**	81.32	**82.16%**

Video-understanding performance. We further evaluate PackAdapt on multi-frame inputs using Qwen2.5-VL-7B-Instruct at a 10% retention ratio. All methods use the same frame-sampling configuration and are evaluated on SEED-Bench Video and MVBench. As shown in Table 3, PackAdapt achieves the strongest overall performance among the compared compression methods and retains 97.38% of the Origin performance on average. This result indicates that Information Folding and Representation Adaptation remain effective when the visual sequence expands from a single image to multiple video frames.

Table 3: Video results on Qwen2.5-VL-7B at a 10% retention ratio.

Method	SEED-Bench Acc	MVBench Acc	Avg (%)
Origin	61.22	67.85	100.00
FastV	57.39	62.25	92.75
VisionZip	60.23	63.42	95.93
DivPrune	58.82	63.65	94.94
DART	60.47	60.92	94.28
VisionSelector	58.72	63.33	94.63
IDPruner	60.55	63.90	96.54
PackAdapt	**60.77**	**64.80**	**97.38**

4.3 Ablation Studies

We conduct ablation studies on Qwen2.5-VL-7B-Instruct at a 10% retention ratio. All configurations use the same selector, token budget, training data, and evaluation protocol, while varying Information Folding, learnable routing correction, and Representation Adaptation. Results are reported on five representative image benchmarks in Figure 3.

We compare four configurations. **Selection Only** directly forwards the selected tokens and is equivalent to VisionSelector. **Folding Only** introduces Information Folding without Representation Adaptation. **Cosine Routing + Adaptation** retains Representation Adaptation but removes the learnable routing correction, allocating information solely according to cosine similarity. **Full PackAdapt** includes all three components.

Effects of Information Folding and Representation Adaptation. Compared with Selection Only, Folding Only improves several benchmarks but decreases ChartQA, indicating that Information Folding itself has task-dependent effects in our setting. Full PackAdapt, which further applies Representation Adaptation to the folded features, improves all five benchmarks over Folding Only. This result shows that learning a post-folding transformation provides additional benefits for downstream prediction.

Effect of learnable routing. The comparison between Cosine Routing + Adaptation and Full PackAdapt isolates the contribution of the learnable routing term. Full PackAdapt performs better across all five benchmarks, indicating that augmenting cosine similarity with task-aware routing is beneficial within our framework. Downstream supervision can therefore further guide how information from pruned tokens is allocated among retained positions.

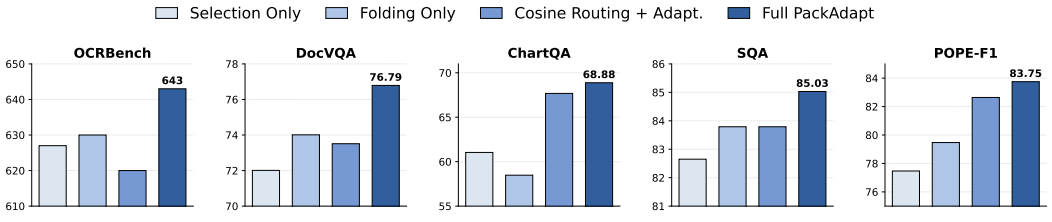

Figure 3: Module ablation on Qwen2.5-VL-7B at a 10% retention ratio.

4.4 Hyperparameter Sensitivity

We study the adapter rank r_a, routing rank r_f, and folding strength α on Qwen2.5-VL-7B-Instruct at a 10% retention ratio, varying one hyperparameter at a time. The default configuration is $r_a = 128$, $r_f = 256$, and $\alpha = 0.6$. Figure 4 reports performance changes relative to the corresponding default setting, to assess whether performance is robust to these design choices.

Adapter rank. For $r_a \in 64, 128, 256$, performance remains stable across the three settings, and increasing the rank beyond 128 brings little additional benefit. We therefore keep $r_a = 128$, which provides sufficient capacity without unnecessarily enlarging the adapter.

Routing rank. Changing r_f from 64 to 256 has only a small effect on overall performance. We use $r_f = 256$ in the main experiments and observe no strong sensitivity to the routing rank over this range, indicating that the routing module does not require careful rank tuning.

Folding strength. Folding strength shows stronger task dependence. A smaller value ($\alpha = 0.3$) favors text-intensive benchmarks such as OCRBench and DocVQA, whereas a larger value ($\alpha = 1.0$) benefits some reasoning-oriented tasks but degrades several OCR-related benchmarks. We therefore use $\alpha = 0.6$, which provides the most balanced overall performance.

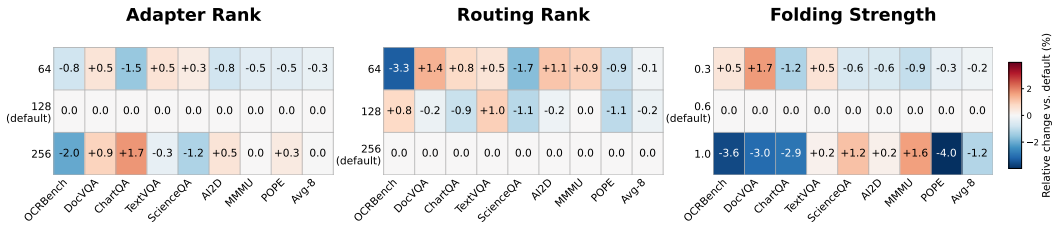

Figure 4: Hyperparameter sensitivity on Qwen2.5-VL-7B-Instruct at a 10% retention ratio. Each cell reports the relative performance change (%) from the corresponding default setting.

4.5 Efficiency Analysis

We evaluate inference efficiency on MVBench using Qwen2.5-VL-7B-Instruct under a long visual sequence setting. The batch size is fixed to 1, and all videos are decoded with up to 32 uniformly sampled frames. All methods are evaluated under the same environment. We report **Prefill Latency**, which measures the time required to process the multimodal input and construct the KV cache for generation, and **E2E Latency**, which measures the total inference time from receiving the input to completing response generation.

Table 4: Inference efficiency on MVBench with Qwen2.5-VL-7B-Instruct.

Method	Prefill (ms)	E2E (ms)
Baseline	843	1020
FastV	386	568
DivPrune	414	576
IDPruner	377	554
PackAdapt	**344**	**524**

As shown in Table 4, PackAdapt achieves the lowest latency among the evaluated pruning methods despite introducing Information Folding and Representation Adaptation before the language model. Compared with the unpruned baseline, PackAdapt reduces Prefill latency from 843 ms to 344 ms and E2E latency from 1020 ms to 524 ms, corresponding to reductions of approximately 59% and 49%, respectively. These results show that the additional lightweight modules introduce limited overhead and preserve the substantial efficiency gains obtained by shortening the visual sequence processed by the language model.

5 Conclusion

We presented PackAdapt, which augments selector-retained tokens with task-aware Information Folding and lightweight Representation Adaptation while keeping the base MLLM frozen and the token budget unchanged. The selector, routing module, and adapter are trained jointly with the language-modeling objective, allowing pruned information to be reused before only the retained tokens enter the language model. Across multiple backbones and image and video benchmarks, PackAdapt consistently improves compressed-model performance, especially at lower retention ratios, while adding little inference overhead. These results support learning both pruned-information routing and retained-feature adaptation under a fixed visual token budget.

Ethics Statement

We confirm that this work complies with the ICLR Code of Ethics and, to the best of our knowledge, does not raise any foreseeable ethical concerns.

Reproducibility Statement

We provide the methodological and experimental details required to reproduce our results throughout the paper and appendix. The architecture, information routing, representation adaptation, and joint training objective of PackAdapt are specified in Section 3. Experimental settings, including the evaluated backbones, benchmarks, metrics, baselines, training data, and main hyperparameters, are described in the experimental section, with additional implementation details provided in Appendix A.1. All reported results are averaged over three runs with different random seeds. Upon acceptance, we will publicly release the source code, training and evaluation configurations, and scripts needed to reproduce the experiments.

LLM Usage

In the preparation of this work, the authors used LLMs primarily as a tool for polishing and refining the English language expression. The LLM was employed to assist in rephrasing certain sentences to improve clarity, fluency, and academic tone. It is important to note that all core ideas, theoretical developments, experimental design, results analysis, and conclusions are solely those of the authors. The LLM did not contribute to the intellectual substance of the research, nor was it used for generating any scientific content or ideation. The final manuscript has been thoroughly reviewed and approved by all authors.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Catalin Ionescu, Arthur Mensch, Jean-Baptiste Lespiau, Olivier Merle, You He, Tom Morrone, et al. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 23716–23736, 2022.

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. DivPrune: Diversity-based Visual Token Pruning for Large Multimodal Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9392–9401, June 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Alvar_DivPrune_Diversity-based_Visual_Token_Pruning_for_Large_Multimodal_Models_CVPR_2025_paper.html.

Xiang An, Yin Xie, Kaicheng Yang, Wenkang Zhang, Xiuwei Zhao, Zheng Cheng, Yirui Wang, Songcen Xu, Changrui Chen, Didi Zhu, Chunsheng Wu, Huajie Tan, Chunyuan Li, Jing Yang, Jie Yu, Xiyao Wang, Bin Qin, Yumeng Wang, Zizhen Yan, Ziyong Feng, Ziwei Liu, Bo Li, and Jiankang Deng. LLaVA-OneVision-1.5: Fully Open Framework for Democratized Multimodal Training, 2025. URL https://arxiv.org/abs/2509.23661.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL Technical Report, 2025. URL https://arxiv.org/abs/2502.13923.

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

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An Image is Worth 1/2 Tokens After Layer 2: Plug-and-Play Inference Acceleration for Large Vision-Language Models, 2024. URL https://arxiv.org/abs/2403.06764.

Yu Chen, Xiaohong Li, Xiaole Wang, Jianjin Zhang, Jun Sun, and Yafeng Deng. Craft: Compression via recursive adaptive fusion of video tokens for vision-language models, 2026.

Peng Gao, Shijie Geng, Renrui Zhang, Teli Fang, Ren Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. In *International Journal of Computer Vision*, 2023.

Wenxuan Huang, Zijie Zhai, Yunhang Shen, Shaosheng Cao, Fei Zhao, Xiangfeng Xu, Zheyu Ye, and Shaohui Lin. Dynamic-llava: Efficient multimodal large language models via dynamic vision-language context sparsification. In *International Conference on Learning Representations*, 2025.

Pengfei Jiang, Hanjun Li, Linglan Zhao, Fei Chao, Ke Yan, Shouhong Ding, and Rongrong Ji. VISA: Group-wise Visual Token Selection and Aggregation via Graph Summarization for Efficient MLLMs Inference, 2025. URL https://arxiv.org/abs/2508.17857.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A Diagram Is Worth A Dozen Images, 2016. URL https://arxiv.org/abs/1603.07396.

Minchul Kim, Shangqian Gao, Yen-Chang Hsu, Yilin Shen, and Hongxia Jin. Token fusion: Bridging the gap between token pruning and token merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy Visual Task Transfer, 2024a. URL https://arxiv.org/abs/2408.03326.

Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. SEED-Bench: Benchmarking Multimodal LLMs with Generative Comprehension, 2023a. URL https://arxiv.org/abs/2307.16125.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*, pp. 19730–19742. PMLR, 2023b.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. MVBench: A Comprehensive Multi-modal Video Understanding Benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22195–22206, 2024b. URL https://openaccess.thecvf.com/content/CVPR2024/html/Li_MVBench_A_Comprehensive_Multi-modal_Video_Understanding_Benchmark_CVPR_2024_paper.html.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating Object Hallucination in Large Vision-Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 292–305. Association for Computational Linguistics, 2023c. doi: 10.18653/v1/2023.emnlp-main.20. URL https://aclanthology.org/2023.emnlp-main.20/.

You Liang, Cheng Ge, Zhi Tong, Yiran Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. In *International Conference on Learning Representations*, 2022.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *Computer Vision – ECCV 2014*, pp. 740–755. Springer International Publishing, 2014. doi: 10.1007/978-3-319-10602-1_48. URL http://dx.doi.org/10.1007/978-3-319-10602-1_48.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual Instruction Tuning, 2023. URL https://arxiv.org/abs/2304.08485.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xucheng Yin, Cheng lin Liu, Lianwen Jin, and Xiang Bai. OCRBench: On the Hidden Mystery of OCR in Large Multimodal Models, 2024. URL https://arxiv.org/abs/2305.07895.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to Explain: Multimodal Reasoning via Thought Chains for Science Question Answering, 2022. URL https://arxiv.org/abs/2209.09513.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A Benchmark for Question Answering about Charts with Visual and Logical Reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2263–2279. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.findings-acl.177. URL https://aclanthology.org/2022.findings-acl.177/.

Minesh Mathew, Dimosthenis Karatzas, and C.V. Jawahar. DocVQA: A Dataset for VQA on Document Images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2200–2209, 2021. URL https://openaccess.thecvf.com/content/WACV2021/html/Mathew_DocVQA_A_Dataset_for_VQA_on_Document_Images_WACV_2021_paper.html.

Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. OCR-VQA: Visual Question Answering by Reading Text in Images. In *ICDAR*, 2019. URL https://ocr-vqa.github.io/.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, Cho-Jui Hsieh, and Dacheng Tao. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems*, volume 34, pp. 13937–13949, 2021.

Michael S Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: What can 8 learned tokens do for images and videos? *arXiv preprint arXiv:2106.11297*, 2021.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. LLaVA-PruMerge: Adaptive Token Reduction for Efficient Large Multimodal Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22857–22867, October 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Shang_LLaVA-PruMerge_Adaptive_Token_Reduction_for_Efficient_Large_Multimodal_Models_ICCV_2025_paper.html.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards VQA Models That Can Read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Singh_Towards_VQA_Models_That_Can_Read_CVPR_2019_paper.html.

Yifan Tan, Yifu Sun, Shirui Huang, Hong Liu, Guanghua Yu, Jianchen Zhu, and Yangdong Deng. IDPruner: Harmonizing Importance and Diversity in Visual Token Pruning for MLLMs, 2026. URL https://arxiv.org/abs/2602.13315.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Ziteng Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A Fully Open, Vision-Centric Exploration of Multimodal LLMs, 2024. URL https://arxiv.org/abs/2406.16860.

Zichen Wen, Yifeng Gao, Shaobo Wang, Junyuan Zhang, Qintong Zhang, Weijia Li, Conghui He, and Linfeng Zhang. Stop Looking for Important Tokens in Multimodal Language Models: Duplication Matters More, 2025. URL https://arxiv.org/abs/2502.11494.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19792–19802, June 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Yang_VisionZip_Longer_is_Better_but_Not_Necessary_in_Vision_Language_CVPR_2025_paper.html.

Xubing Ye, Yukang Gan, Yixiao Ge, Xiao-Ping Zhang, and Yansong Tang. Atp-llava: Adaptive token pruning for large vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24972–24982, 2025.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. MMMU: A Massive Multi-discipline Multimodal Understanding and Reasoning Benchmark for Expert AGI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9556–9567, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Yue_MMMU_A_Massive_Multi-discipline_Multimodal_Understanding_and_Reasoning_Benchmark_for_CVPR_2024_paper.html.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. LMMs-Eval: Reality Check on the Evaluation of Large Multimodal Models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 881–916. Association for Computational Linguistics, 2025a. doi: 10.18653/v1/2025.findings-naacl.51. URL https://aclanthology.org/2025.findings-naacl.51/.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. SparseVLM: Visual Token Sparsification for Efficient Vision-Language Model Inference, 2025b. URL https://arxiv.org/abs/2410.04417.

Jiaying Zhu, Yurui Zhu, Xin Lu, Wenrui Yan, Dong Li, Kunlin Liu, Xueyang Fu, and Zheng-Jun Zha. VisionSelector: End-to-End Learnable Visual Token Compression for Efficient Multimodal LLMs, 2026. URL https://arxiv.org/abs/2510.16598.

A Appendix

A.1 Additional Implementation Details

Input configuration. Both Qwen2.5-VL-7B-Instruct and LLaVA-OneVision-1.5-8B use dynamic-resolution image inputs. During evaluation, the minimum and maximum pixel budgets are set to 200,704 (256×28^2) and 1,605,632 (2048×28^2), respectively. The vision encoders use an original spatial patch size of 14×14, followed by 2×2 spatial merging, such that each visual token corresponds to a 28×28 pixel region. For video experiments, we use Qwen2.5-VL-7B-Instruct and uniformly sample up to 32 frames from each video.

Training data and optimization. Following VisionSelector (Zhu et al., 2026), we train on approximately 144K examples from Cambrian-737K (Tong et al., 2024), including ChartQA (Masry et al., 2022), OCRVQA (Mishra et al., 2019), and a randomly sampled 10% subset of COCO (Lin et al., 2014). The base multimodal model remains frozen throughout training, and only the selector, Information Folding, and Representation Adaptation modules are optimized. We use AdamW (Loshchilov & Hutter, 2019) with an initial learning rate of 5×10^{-5}, zero weight decay, a cosine learning-rate schedule, and a warmup ratio of 0.03. All models are trained for one epoch.

Training is conducted on four H20 GPUs using bf16 precision. The global batch size is 256, with a per-GPU batch size of 16 and four gradient accumulation steps. Evaluation is performed with LMMs-Eval (Zhang et al., 2025a) on one H20 GPU.

PackAdapt hyperparameters. Unless otherwise specified, the adapter rank and routing rank are set to $r_a = 128$ and $r_f = 256$, respectively, and the folding strength is set to $\alpha = 0.6$. The routing temperature τ is initialized to 0.01 and constrained to remain positive through exponential reparameterization, with its value clipped to $[10^{-4}, 1]$. The scaling coefficient of the learnable routing correction is fixed at $\gamma = 100$. Following VisionSelector (Zhu et al., 2026), the selection-consistency weight λ is linearly increased from 0.1 to 2.0 over the full training process.

A.2 BENCHMARK DETAILS

We provide additional details on the image- and video-understanding benchmarks used in our experiments. The selected benchmarks cover a broad range of capabilities, including visual text recognition, document and chart understanding, diagram reasoning, knowledge-intensive multimodal reasoning, object hallucination, and temporal video understanding.

OCRBench. OCRBench (Liu et al., 2024) is a comprehensive benchmark for evaluating the OCR-related capabilities of multimodal models. It covers diverse text-centric tasks, including text recognition, scene-text visual question answering, document-oriented VQA, key information extraction, and handwritten mathematical expression recognition. We report the standard OCRBench score, with a maximum score of 1,000.

DocVQA. DocVQA (Mathew et al., 2021) evaluates visual question answering over document images. Answering its questions requires models to recognize textual content while also exploiting document layout and structural information. We use Average Normalized Levenshtein Similarity (ANLS) as the evaluation metric.

TextVQA. TextVQA (Singh et al., 2019) focuses on question answering that requires reading and reasoning over text appearing in natural images. It therefore evaluates whether visual information associated with scene text is preserved after token compression. We report Exact Match (EM) following our evaluation protocol.

ChartQA. ChartQA (Masry et al., 2022) evaluates question answering over charts, with questions involving visual interpretation as well as logical and numerical reasoning. Compared with conventional scene understanding, successful prediction often requires preserving fine-grained chart elements and their relationships. We report Relaxed Accuracy.

ScienceQA. ScienceQA (Lu et al., 2022) is a multimodal science question-answering benchmark covering a broad range of science topics. Its questions may combine textual context with visual information and require knowledge-based reasoning to determine the correct answer. We report Exact Match (EM).

AI2D. AI2D (Kembhavi et al., 2016) evaluates understanding and reasoning over illustrative scientific diagrams. The benchmark contains diagrams with objects, textual elements, and structural relationships, requiring models to jointly interpret visual components and their interactions. We report Exact Match (EM).

MMMU. MMMU (Yue et al., 2024) is a knowledge-intensive multimodal benchmark constructed from college-level materials across a wide range of disciplines. It contains heterogeneous visual inputs, including diagrams, charts, tables, maps, and other domain-specific representations, and emphasizes multimodal perception combined with expert-level reasoning. We report Accuracy.

POPE. POPE (Li et al., 2023c) evaluates object hallucination in large vision-language models through binary questions about whether particular objects are present in an image. It provides a complementary perspective to conventional question answering by measuring whether the model remains grounded in the available visual evidence. We report the F1 score.

SEED-Bench Video. SEED-Bench (Li et al., 2023a) is a multiple-choice benchmark for evaluating generative multimodal comprehension across image and video inputs. We use its video evaluation setting to assess whether token compression preserves information required for understanding multi-frame visual content and temporal events. Performance is measured by Accuracy.

MVBench. MVBench (Li et al., 2024b) is a comprehensive benchmark designed specifically for multimodal video understanding. It consists of 20 temporally oriented tasks spanning perception and reasoning, with many questions requiring information from multiple frames rather than a single representative image. We report Accuracy following the official evaluation protocol.

A.3 Baseline Details

We compare PackAdapt with representative visual token compression methods covering attention-based importance estimation, redundancy- and diversity-aware selection, aggregation-based compression, and learnable token selection. For all compressed methods, the final visual token budget is matched to the corresponding retention ratio used by PackAdapt.

FastV. FastV (Chen et al., 2024) is a training-free visual token pruning method based on the observation that attention over visual tokens becomes highly sparse in multimodal language models. It estimates token importance using attention signals from early layers and removes low-importance visual tokens from subsequent computation, thereby reducing the sequence length processed by later language-model layers.

VisionZip. VisionZip (Yang et al., 2025) is a training-free visual token compression method that exploits redundancy in vision-encoder representations. It first identifies dominant tokens according to visual attention and then compresses the remaining non-dominant tokens into a smaller set of contextual tokens through similarity-based token merging. The resulting sequence therefore combines individually retained dominant tokens with aggregated contextual representations.

DART. DART (Wen et al., 2025) approaches visual token reduction from the perspective of duplication rather than conventional importance estimation. It selects a small set of pivot tokens and measures the redundancy of other visual tokens relative to these pivots, retaining tokens with lower duplication. This yields a training-free pruning strategy that favors complementary rather than repetitive visual information.

DivPrune. DivPrune (Alvar et al., 2025) formulates visual token pruning as a Max-Min Diversity Problem. Instead of assigning independent importance scores to individual tokens, it selects a subset that maximizes feature diversity among the retained representations. This training-free strategy is designed to reduce redundancy and improve the coverage of the original visual sequence under a limited token budget.

VisionSelector. VisionSelector (Zhu et al., 2026) formulates visual token compression as an end-to-end learnable selection problem. It employs a lightweight importance scorer together with differentiable Top-K selection, allowing the selection mechanism to be optimized using downstream supervision while the underlying multimodal model remains frozen. PackAdapt directly reuses VisionSelector as its token selection mechanism; therefore, the comparison with VisionSelector isolates the effect of Information Folding and Representation Adaptation beyond learned token selection.

IDPruner. IDPruner (Tan et al., 2026) jointly considers token importance and semantic diversity during visual token selection. It uses Maximal Marginal Relevance to balance these two criteria, discouraging the retained subset from containing highly redundant tokens while preserving visually informative content. The method performs one-shot pruning without relying on attention maps, making it compatible with efficient attention implementations.

A.4 Supplementary Experiments on LLaVA-1.5-7B

We further evaluate PackAdapt on LLaVA-1.5-7B (Liu et al., 2023), extending the experiments to a backbone with a fixed visual sequence of 576 tokens. All compressed methods retain 64 visual tokens, corresponding to an 11.11% retention ratio and an 88.89% compression ratio. Table 5 reports results on the same eight image-understanding benchmarks used in the main experiments, with Avg-8 computed using the same per-benchmark normalization against the uncompressed model.

PackAdapt achieves the highest Avg-8 of 94.37%, improving over VisionSelector by 4.04 percentage points and over the strongest competing baseline, IDPruner, by 2.88 percentage points. Among the compressed methods, PackAdapt obtains the best results on OCRBench, DocVQA, ChartQA, and TextVQA. Compared with VisionSelector, it improves all eight benchmark scores, including gains of 1.39 points on DocVQA and 2.64 points on ChartQA. These results provide further evidence

Table 5: Image-understanding results on LLaVA-1.5-7B with 64 retained visual tokens. Avg-8 is the mean performance normalized by the uncompressed model on each benchmark. Bold marks the best result among compressed methods.

Method	**OCRBench** Score	**DocVQA** ANLS	**ChartQA** Relaxed	**TextVQA** EM	**ScienceQA** EM	**AI2D** EM	**MMMU** Acc	**POPE** F1	**Avg**-8
Origin (576 Visual Tokens)									
LLaVA-1.5-7B	198	24.09	18.12	47.78	66.19	52.78	36.00	85.86	100.00%
Retain 64 Tokens (88.89% Compression Ratio)									
FastV	40	9.74	12.24	17.69	63.71	49.42	34.00	48.00	63.18%
VisionZip	105	15.75	15.76	42.75	66.44	51.20	**36.20**	77.00	85.31%
VisionSelector	170	17.29	15.96	42.98	65.84	50.65	35.44	79.94	90.33%
DivPrune	157	15.69	15.12	40.83	66.44	50.00	35.64	**85.60**	88.39%
DART	124	12.35	15.20	30.98	**66.83**	**51.55**	34.81	73.90	80.50%
IDPruner	173	17.65	16.32	43.21	66.68	50.55	35.11	83.06	91.49%
PackAdapt (Ours)	**175**	**18.68**	**18.60**	**43.45**	66.63	51.36	36.11	83.44	**94.37%**

that enhancing retained representations with information from pruned tokens is beneficial under aggressive compression, extending the findings to a backbone with a fixed visual token count.