SHALLOW POOL, DEEP PRUNE: ACCELERATING VISION-LANGUAGE MODELS VIA HYBRID TOKEN REDUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision Language Models (VLMs) process hundreds or even thousands of visual tokens to perform complex visual understanding and reasoning tasks. However, such long input sequences result in substantial computational overhead during inference. While a range of token reduction approaches have been proposed to accelerate VLMs, they generally encounter early-stage context loss, inadequate salient token retention and suboptimal pruning allocation, leading to significant performance degradation. To address these issues, we propose a novel hybrid token reduction method, termed HTR-VLM, motivated by the depth-dependent variation in compression objectives across VLM layers. Specifically, we introduce the FFN Token Pooling with Residual Approximation (FTP) module, which leverages residual connections to approximate full image-feature updates with pooled tokens, reducing shallow FFN computation while preserving full context. Building on this, Salient Token Retention with Multi-Criteria Scoring (STR) selects and preserve important tokens for full computation by integrating multiple cues. Finally, a Depth-Aware Token Pruning (DATP) mechanism is adopted to enable more precise token reduction in deeper layers. Extensive experiments demonstrate that HTR-VLM substantially reduces computational overhead and maintains higher model performance compared to state-of-the-art token reduction methods. We will release our code upon paper acceptance.

1 INTRODUCTION

In recent years, Vision-Language Models (VLMs) have achieved remarkable progress in multimodal understanding and reasoning Yin et al. (2024); Zhang et al. (2024), demonstrating state-of-the-art performance on diverse tasks such as scene understanding Hudson & Manning (2019), scientific reasoning Lu et al. (2022), and visual text understanding Singh et al. (2019). Representative models like InstructBLIP Dai et al. (2023) and LLaVA Liu et al. (2023; 2024a) have established the mainstream architecture, where visual tokens are concatenated with text and fed into a Large Language Model (LLM) Yin et al. (2024). To capture fine-grained visual information, the number of visual tokens often reaches hundreds or even thousands Zhang et al. (2025), incurring substantial computational overhead that limits deployment in resource-constrained scenarios Jin et al. (2025).

To alleviate this burden, various techniques have been explored, including parameter pruning Wu et al. (2025b); Liang et al. (2025), quantization Li et al. (2025a); Wang et al. (2024), and distillation Cai et al. (2025); Shu et al. (2024). Among them, token reduction has proven particularly effective by adaptively reducing the visual tokens involved in computation. Despite these advances, existing methods still struggle to achieve a favorable efficiency-accuracy trade-off.

We identify three key limitations: 1) *Early-stage context loss.* Most methods discard or merge tokens irreversibly, and to obtain higher savings, reduction shifts to earlier layers Chen et al. (2024); Xing et al. (2025), causing premature context loss that constrains information aggregation in deeper layers. 2) *Insufficient salient token retention.* Fixed-pattern pooling or merging Wu et al. (2025a); Yang et al. (2025); Mao et al. (2025) tends to blur critical information, as certain visual tokens intrinsically crucial for multimodal tasks are not sufficiently preserved in their complete and independent representations. 3) *Suboptimal pruning allocation.* Token pruning in deeper layers is essential for

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.

substantial acceleration, yet most methods lack a principled strategy for determining pruning layers and ratios Ye et al. (2025).

To address these challenges, we propose HTR-VLM, a hybrid token reduction framework. The FFN Token Pooling with Residual Approximation (FTP) module leverages pooled tokens to approximate FFN updates and up-samples them to full image features via residual connections, reducing computation while preserving shallow-layer context. The Salient Token Retention with Multi-Criteria Scoring (STR) module selects critical tokens via similarity and vision-enhanced attention scores, computes their updates directly through the FFN, and restores them to their original positions. Additionally, a Depth-Aware Token Pruning (DATP) mechanism analyzes the sensitivity of token reduction across deep layers and applies multi-layer pruning to achieve better accuracy-efficiency balance.

In summary, our contributions are threefold:

- We propose HTR-VLM, a novel hybrid token reduction framework that accelerates VLMs by performing pooling and pruning operations in shallow and deep layers, respectively.
- We design the FFN Token Pooling with Residual Approximation (FTP), Salient Token Retention with Multi-Criteria Scoring (STR) and Depth-Aware Token Pruning (DATP) modules to mitigate context loss, critical token dis-fidelity and suboptimal pruning allocation.
- We conduct extensive experiments on representative VLMs, demonstrating that HTR-VLM substantially improves inference efficiency while maintaining higher model performance compared to state-of-the-art token reduction methods.

1.1 Vision-Language Models

Benefiting from the success of Large Language Models (LLMs) Yao et al. (2023), Vision-Language Models (VLMs) achieve remarkable multimodal understanding and reasoning capabilities by concatenating visual tokens with textual input Liu et al. (2023). Through supervised fine-tuning and instruction-following inference Dai et al. (2023); Liu et al. (2023), they demonstrate strong generalization across diverse visual question answering tasks Goyal et al. (2017); Liu et al. (2024b); Fu et al. (2023). Architecturally, VLMs typically adopt the standard Transformer Vaswani et al. (2017) structure composed of interleaved Multi-Head Attention (MHA) and Feed-Forward Network (FFN) layers. Increasing image resolution has emerged as a common strategy to further unlock model capacity An et al. (2025); Liu et al. (2024a), but inevitably expands the input sequence to hundreds or even thousands of visual tokens, substantially increasing computational cost Jin et al. (2025). This motivates the need for effective acceleration techniques.

1.2 Acceleration Methods for VLMs

Recent studies have explored VLM acceleration from multiple perspectives. Architecturally, a line of work focuses on training compact models or distilling capacity to smaller architectures Zhou et al. (2024); Cai et al. (2025); Shu et al. (2024). Compression-based approaches reduce model complexity through parameter pruning Wu et al. (2025b); Liang et al. (2025) and quantization Li et al. (2025a); Wang et al. (2024), thereby lowering memory footprint. Token-level techniques directly address the growing sequence length Chen et al. (2024), achieving more significant inference speedups. Additionally, since attention is the primary source of quadratic complexity, dedicated efforts target its optimization Song et al. (2024); Li et al. (2025b), among which FlashAttention Dao et al. (2022) has been widely adopted in modern VLMs. As attention becomes highly optimized, the inference bottleneck gradually shifts toward other modules such as FFNs.

1.3 Token Reduction for VLMs

Token reduction methods reduce the number of visual tokens processed during inference and have proven effective for VLM acceleration. FastV Chen et al. (2024) first identified the inherent sparsity in visual attention, showing that discarding up to half of the visual tokens incurs only minor degradation. Building on this, subsequent methods explore attention-guided token selection Zhang et al. (2025); Ye et al. (2025); Xing et al. (2025), while others incorporate similarity-based metrics to better preserve critical information Yang et al. (2025); Mao et al. (2025). More recently, training-based

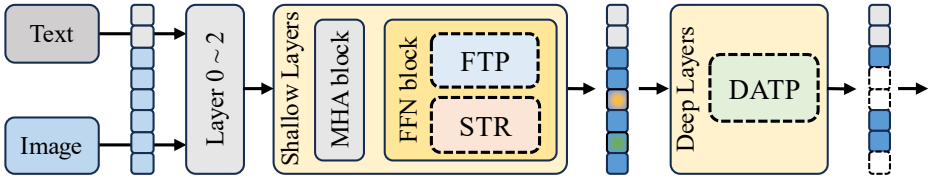

Figure 1: Inference procedure of HTR-VLM. In shallow layers, computation is reduced within FFN blocks via a reversible token pooling mechanism that preserves sequence length, where FTP and STR work complementarily: approximating updates for non-salient tokens while retaining accurate updates for salient ones. In deep layers, DATP performs explicit token pruning, progressively shortening the sequence to further reduce overhead.

approaches Shao et al. (2025); Zeng et al. (2025) have been explored to enhance pruning quality, yet they require additional training overhead. Despite these advances, existing methods typically discard tokens in shallow layers, causing significant loss of contextual information crucial for accuracy. In this work, we investigate context-preserving acceleration, aiming to improve the accuracy-efficiency trade-off.

2 METHOD

2.1 FRAMEWORK OVERVIEW

Most contemporary VLMs adopt an architecture that bridges a visual encoder with an LLM. Given an input image, the visual encoder extracts a sequence of visual features, denoted as $\boldsymbol{V} \in \mathbb{R}^{N_v \times D}$, where N_v represents the number of visual tokens and D is the feature dimension. These raw features are subsequently projected via a lightweight projector to obtain the visual hidden states $\boldsymbol{X}^{v,0} \in \mathbb{R}^{N_v \times D}$, which are compatible with the textual token embedding space. Meanwhile, an input text prompt is tokenized into a textual sequence $\boldsymbol{X}^{t,0} \in \mathbb{R}^{N_t \times D}$ with length N_t. These visual and language tokens are then concatenated and fed into the LLM backbone $\mathcal{F}$ to perform multimodal reasoning.

Here, we focus exclusively on the computation of each layer during the first forward pass. Let $\boldsymbol{X}^{v,l}$ and $\boldsymbol{X}^{t,l}$ denote the visual and textual hidden states at layer l, respectively, and let $\mathcal{F}^l(\cdot)$ represent the computation performed by layer l. The layer-wise forward pass can then be expressed as:

$$[\boldsymbol{X}^{v,l+1}; \boldsymbol{X}^{t,l+1}] = \mathcal{F}^l([\boldsymbol{X}^{v,l}; \boldsymbol{X}^{t,l}]), \tag{1}$$

where $[\cdot;\ \cdot]$ denotes concatenation along the sequence dimension.

Building upon this, as N_v typically reaches hundreds, substantially exceeding N_l, we focus on reducing visual token sequence $\boldsymbol{X}^{v,l}$ across layers to improve inference efficiency. As illustrated in fig. 1, our proposed HTR-VLM adopts a hybrid token reduction scheme across different depths of the LLM. In order to address the aforementioned challenges, the FTP and STR modules are jointly employed to reduce the computation in shallow FFNs while preserving the full token sequence, thereby maintaining contextual completeness and positional consistency. In addition, the DATP module further performs token pruning in deep layers, enabling more aggressive acceleration.

2.2 FFN TOKEN POOLING WITH RESIDUAL APPROXIMATION

As shown in prior studies Wu et al. (2026; 2025a), multi-head attention (MHA) in shallow Transformer Vaswani et al. (2017) layers is crucial for aggregating cross-token and cross-modal information, requiring full contextual integrity. Accordingly, we reduce the number of tokens involved in feed-forward network (FFN) computation while keeping attention operations unchanged.

Formally, given an input sequence $\boldsymbol{X}$, the computation in an FFN block can be expressed as:

$$\boldsymbol{X}' = \boldsymbol{X} + f(\boldsymbol{X}) = \boldsymbol{X} + (\sigma(\boldsymbol{X}\boldsymbol{W}_{\text{gate}}) \odot (\boldsymbol{X}\boldsymbol{W}_{\text{up}}))\boldsymbol{W}_{\text{down}}, \tag{2}$$

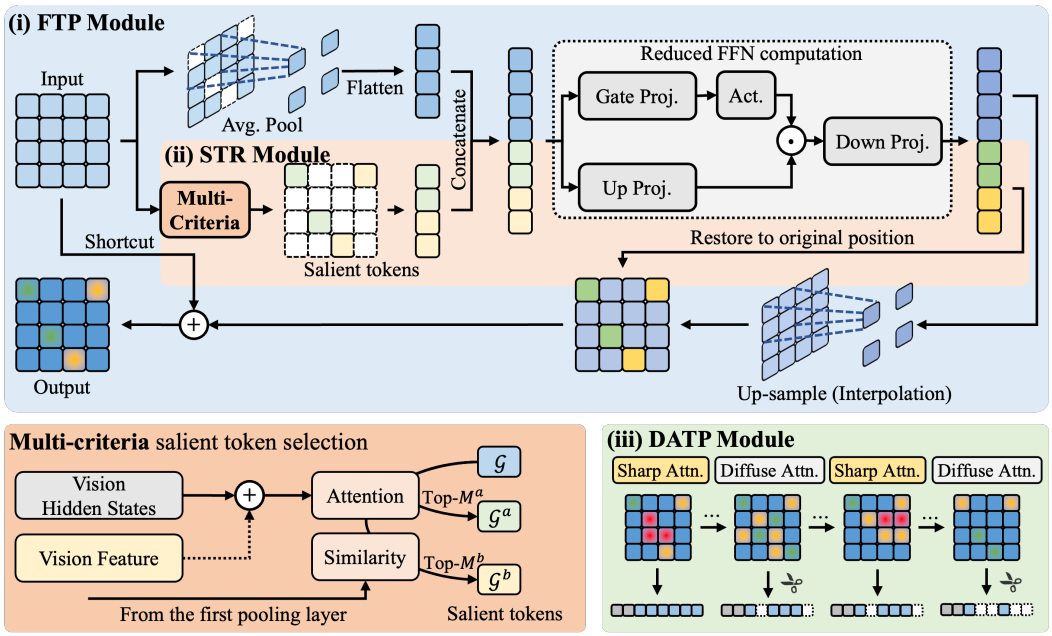

Figure 2: Overview of HTR-VLM. (i) FTP pools visual tokens within fixed spatial windows into representative tokens, whose FFN updates approximate local feature evolution and are up-sampled back to all tokens via the residual connection. (ii) STR identifies critical tokens via attention- and similarity-based criteria, retains full FFN computation for them, and reintegrates their accurate updates into the pooled-and-upsampled representations. (iii) DATP performs depth-aware pruning allocation in deep layers, guided by sensitivity analysis.

where $f(\cdot)$ represents the FFN operation, σ is the nonlinear activation, and $\boldsymbol{W}_{\text{gate}}, \boldsymbol{W}_{\text{up}}, \boldsymbol{W}_{\text{down}}$ are the gate-, up- and down- projection matrices, respectively. For simplicity, the layer index is omitted in the notation.

Intuitively, the residual shortcut preserves the dominant information flow across layers, while $f(\cdot)$ provides an incremental feature refinement Wu et al. (2026). Meanwhile, visual representations exhibit strong spatial locality, where neighboring tokens often encode redundant or highly correlated information. These observations motivate us to approximate the FFN computations using a reduced set of representative tokens, while leveraging the residual connection to maintain global context and feature continuity.

Specifically, without affecting the correctness of FFN computation, we first reshape the visual hidden state sequence $\boldsymbol{X}^v$ back to its original two-dimensional image space, denoted as $\boldsymbol{H} \in \mathbb{R}^{H \times W \times D}$, where $H \times W = N^v$ correspond to the spatial height and width. The spatial grid is then partitioned into non-overlapping local regions of size $c \times c$, within which an average pooling operation $\text{AvgPool2D}(\cdot)$ is applied to obtain a compact set of representative tokens:

$$\hat{\boldsymbol{H}} = \text{AvgPool2D}(\boldsymbol{H}, c) \in \mathbb{R}^{\frac{H}{c} \times \frac{W}{c} \times D}. \tag{3}$$

Then, the pooled representative tokens are fed into $f(\cdot)$ to approximate the feature updates of the original dense token set. The resulting outputs are subsequently up-sampled and added back to their corresponding spatial locations via the residual pathway, as formulated in eq. (4), thereby preserving the original sequence length:

$$\Delta = \text{UpSample}(f(\hat{\boldsymbol{H}})) \in \mathbb{R}^{H \times W \times D}, \ \boldsymbol{H}' = \boldsymbol{H} + \Delta, \tag{4}$$

where $\text{UpSample}(\cdot)$ denotes the up-sampling operation, implement-ed here via bilinear interpolation. $\boldsymbol{H}'$ represents the output of the FFN block with reduced computational cost.

By computing FFN updates on pooled tokens and leveraging residual approximation, FTP substantially reduces the FFN computation cost in shallow layers, while preserving spatial structure, contextual completeness, and compatibility with subsequent Transformer layers.

2.3 SALIENT TOKEN RETENTION WITH MULTI-CRITERIA SCORING

Although FTP enables substantial computational reduction, pooling inevitably discards fine-grained visual details. Since a small subset of critical tokens plays a decisive role in providing task-targeted information Chen et al. (2024); Mao et al. (2025), we selectively retain such tokens for full computation rather than approximating their updates via pooled representations. Specifically, we consider two complementary sources of critical tokens.

The first source corresponds to visually salient tokens that capture key regions for multimodal reasoning. To identify such tokens, we adopt attention scores as an importance criterion. Formally, given a visual token sequence $\boldsymbol{X}^v$, we compute its key projection $\boldsymbol{K} = \boldsymbol{X}^v\boldsymbol{W}_K$, along with the query projection of the last text token $\boldsymbol{q} = \boldsymbol{X}^t_{N_t}\boldsymbol{W}_Q$. The importance score of the i-th token is then obtained by first obtaining the softmax-normalized attention weights and subsequently averaging across all attention heads, *i.e.*:

$$S^a_i = \frac{1}{h}\sum_{j=1}^{h} \mathrm{Softmax}(\frac{\boldsymbol{q}\boldsymbol{K}^T}{\sqrt{d}})_i, \tag{5}$$

where h and d are the number of attention heads and the dimensionality of each head, respectively.

We further note that attention scores in certain layers can be biased, particularly when shallow-layer features are remapped into the new LLM latent space and have not yet fully converged. To mitigate this, we incorporate the original visual features as a supplementary signal. Specifically, for the specific layer l, the key projections in eq. (5) are computed as

$$\boldsymbol{K} = (\boldsymbol{X}^{v,l} + \boldsymbol{V})\boldsymbol{W}_K, \tag{6}$$

thereby improving the fidelity of salient token selection.

The second source is feature diversity: sufficiently diverse tokens serve as informative anchors for stabilizing contextual representation Alvar et al. (2025), potentially encoding regions underrepresented in earlier layers but salient in later processing. To formalize this, we define a similarity-based importance score. Given the hidden states of all visual tokens $\boldsymbol{X}^v$, the importance of the i-th token is computed as the aggregated cosine similarity with all other tokens:

$$S^b_i = \sum_{j=1}^{N_v}(1 - \frac{\boldsymbol{X}^v_i \cdot \boldsymbol{X}^v_j}{\|\boldsymbol{X}^v_i\| \times \|\boldsymbol{X}^v_j\|}), \tag{7}$$

where tokens exhibiting lower similarity to others (i.e., greater diversity) are considered more informative and are thus prioritized for full FFN computation.

Finally, based on the two importance criteria introduced above, we sequentially select the top-M^a and top-M^b tokens according to S^a_i and S^b_i, respectively. Let $\mathcal{G}$ denote the set of all token indices, and $\mathcal{G}^a$ and $\mathcal{G}^b$ denote the indices of the selected tokens:

$$\mathcal{G}^a = \mathrm{TopK}(\{S^a_i\}_{i\in\mathcal{G}}, M^a),\ \mathcal{G}^b = \mathrm{TopK}(\{S^b_i\}_{i\in\mathcal{G}/\mathcal{G}^a}, M^b), \tag{8}$$

where $\mathrm{TopK}(\cdot, M)$ returns the indices of the M-largest scores. Ultimately, the updates of these salient tokens are computed and restored to their original positions via a scatter operation:

$$\Delta_{\mathcal{G}^a} = f(\boldsymbol{X}^v_{\mathcal{G}^a}),\ \Delta_{\mathcal{G}^b} = f(\boldsymbol{X}^v_{\mathcal{G}^b}). \tag{9}$$

Building on this, since salient tokens undergo full computation, we accordingly adjust the pooling operation in the FTP module. Concretely, the average pooling in eq. (3) is applied only to non-salient tokens, *i.e.* $\mathcal{G}/(\mathcal{G}^a \bigcup \mathcal{G}^b)$.

Remark: By integrating FTP and STR, the reduced FFN computation in each layer can be expressed as:

$$f([\boldsymbol{X}^t;\ \mathrm{Seq}(\hat{\boldsymbol{H}});\ \boldsymbol{X}^v_{\mathcal{G}^a};\ \boldsymbol{X}^v_{\mathcal{G}^b}]), \tag{10}$$

where $\mathrm{Seq}(\cdot)$ denotes flattening the 2D feature map into a sequence. The outputs are then rearranged via eqs. (2), (4) and (9) to update the full token sequence, ensuring both pooled and critical tokens contribute while preserving spatial and contextual integrity.

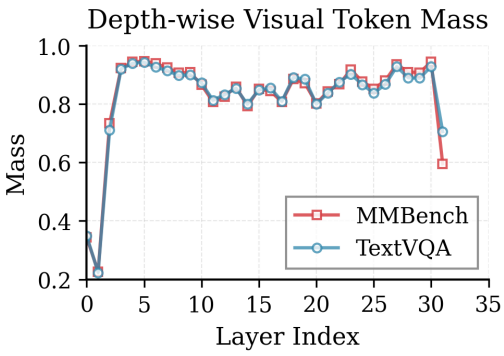

Figure 3: Depth-wise Top-k attention mass in LLaVA.

2.4 Depth-Aware Token Pruning

We select the pruning depth using the concentration of attention over visual tokens. At layer l, let $A^{(l)} \in \mathbb{R}^{B\times H\times T_q\times T_v}$ be the vision attention. We average over queries and normalize to obtain a per-head distribution $\tilde{A}^{(l)} \in \mathbb{R}^{B\times H\times T_v}$. We then compute the Top-k attention mass with $k = \lfloor rT_v \rfloor$:

$$m^{(l)} = \mathbb{E}_{b,h}\Big[\sum_{v\in \text{Top-}k(\tilde{A}^{(l)}_{b,h,:})} \tilde{A}^{(l)}_{b,h,v}\Big]. \tag{11}$$

A higher $m^{(l)}$ indicates sharp attention where the model depends on a few tokens, making pruning risky; a lower $m^{(l)}$ implies diffuse attention with higher redundancy, thus more robust to pruning. Empirically, $m^{(l)}$ over depth exhibits several local minima (valleys). As illustrated in fig. 3, we plot the depth-wise attention mass of LLaVA using all samples from MMBench and TextVQA, and choose a valley as the pruning depth to reduce information loss.

3 Experiments

3.1 Experimental Settings

Model Architecture. We conduct experiments on LLaVA-1.5-7B Liu et al. (2024a), a representative VLM that adopts a ViT-based vision encoder Dosovitskiy (2020) coupled with a Vicuna-7B LLM Chiang et al. (2023), processing a fixed-length sequence of 576 visual tokens for inference.

Datasets and Evaluation Metrics. For training, we follow the standard LLaVA practice Liu et al. (2024a) and use the official supervised fine-tuning (SFT) dataset containing 665K samples. For evaluation, following prior works Zhang et al. (2025); Shao et al. (2025); Xing et al. (2025), we assess methods on seven widely used benchmarks: VQAv2 (VQA^{V2}) Goyal et al. (2017), GQA Hudson & Manning (2019), ScienceQA with image context (SQA^{IMG}) Lu et al. (2022), TextVQA (VQA^{Text}) Singh et al. (2019), MMBench (MMB) Liu et al. (2024b), the sum of perception and cognition scores on MME Fu et al. (2023), and the F1 score on POPE Li et al. (2023). We also report the average score (Avg.) across all benchmarks and the relative performance preservation ratio *w.r.t.* the full model.

Implementation Details. Starting from the pre-trained model (*i.e.*, after training the lightweight projector Liu et al. (2024a)), we perform one epoch of SFT integrated with HTR-VLM to construct the efficient inference model. The FTP and STR modules are applied from layers 3 to 10. Due to the high computational cost of eq. (7), the similarity-based score S_i^b is computed only at layer 3 and reused in subsequent layers. At layer 4, we incorporate eq. (6) to enhance attention-based token selection. The hyperparameters vary with the target number of retained tokens: for 128, 64, and 32 tokens, the scaling factor c is set to 4, 6, and 8, and the memory bank sizes M^a and M^b are set to $0.08|N_v|$, $0.04|N_v|$, and $0.02|N_v|$, respectively. For the DATP module, guided by sensitivity

Table 1: Comparison of different token reduction methods on LLaVA-1.5-7B across varying numbers of retained tokens using 7 representative benchmarks. Best results are highlighted in **bold**.

Method	VQA^{V2}	GQA	SQA^{IMG}	VQA^{Text}	POPE	MME	MMB	Avg.	Rel.
Upper Bound, All 576 Tokens (100%)									
LLaVA-1.5-7B	78.5	61.9	69.5	58.2	85.9	1506.5	64.7	70.6	100.0%
Retain 128 Tokens (↓ 77.8%)									
FastV Chen et al. (2024)	71.0	54.0	69.2	56.4	68.2	1368.9	63.0	64.3	91.1%
PDrop Xing et al. (2025)	74.3	57.1	**70.1**	56.7	77.5	1444.1	62.3	67.2	95.2%
SparseVLM Zhang et al. (2025)	75.1	57.3	69.0	56.3	83.1	1399.3	62.6	67.6	95.8%
PruMerge+ Mao et al. (2025)	75.0	58.2	69.1	54.0	83.1	1408.1	61.8	67.4	95.5%
TRIM Song et al. (2025)	75.4	58.4	68.6	52.2	85.3	1413.4	**63.0**	67.7	95.9%
VisionZip Yang et al. (2025)	75.6	57.6	68.7	**56.9**	83.3	1436.9	62.1	68.0	96.4%
DART Wen et al. (2025)	74.7	57.9	69.1	56.3	80.4	1408.7	60.7	67.1	95.0%
DivPrune Alvar et al. (2025)	76.0	59.4	68.6	55.9	**87.0**	1405.1	61.5	68.4	96.9%
HTR-VLM (Ours)	**77.0**	**61.3**	67.6	55.9	86.1	**1462.4**	62.9	**69.1**	**98.0%**
Retain 64 Tokens (↓ 88.9%)									
FastV Chen et al. (2024)	55.9	46.0	70.1	51.6	35.5	973.5	50.1	51.1	72.4%
PDrop Xing et al. (2025)	56.3	46.1	68.8	49.2	40.8	982.2	48.0	51.2	72.5%
SparseVLM Zhang et al. (2025)	66.9	52.0	69.2	52.1	69.7	1190.4	58.3	61.1	86.6%
PruMerge+ Mao et al. (2025)	71.3	55.4	**69.5**	52.0	75.7	1316.8	59.6	64.2	91.0%
TRIM Song et al. (2025)	72.4	56.6	69.0	49.7	85.9	1350.9	60.9	66.0	93.5%
VisionZip Yang et al. (2025)	72.4	55.1	69.0	**55.5**	77.0	1365.2	60.1	65.3	92.6%
DART Wen et al. (2025)	71.3	54.7	69.3	54.7	73.8	1365.1	59.5	64.5	91.4%
DivPrune Alvar et al. (2025)	74.1	57.5	68.0	54.5	85.5	1334.7	60.1	66.6	94.4%
HTR-VLM (Ours)	**76.7**	**60.8**	68.3	**55.5**	**86.0**	**1453.2**	**64.3**	**69.2**	**98.0%**
Retain 32 Tokens (↓ 94.4%)									
PruMerge+ Mao et al. (2025)	65.6	52.9	67.9	49.2	66.7	1236.6	55.1	59.9	84.9%
TRIM Song et al. (2025)	68.6	54.5	68.1	47.6	84.9	1251.8	57.7	63.4	89.9%
VisionZip Yang et al. (2025)	67.1	51.8	69.1	53.1	69.4	1251.2	57.0	61.4	87.1%
DART Wen et al. (2025)	67.1	52.9	**69.3**	52.2	69.1	1273.3	58.5	61.8	87.6%
DivPrune Alvar et al. (2025)	71.2	54.9	68.6	52.9	81.5	1284.9	57.6	64.4	91.3%
HTR-VLM (Ours)	**76.2**	**60.0**	68.2	**54.8**	**85.6**	**1427.9**	**63.8**	**68.6**	**97.2%**

analysis, token pruning is performed at layers 11 and 20. The pruning counts are: 448/114 tokens removed for the 128-token target, 512/60 for 64 tokens, and 544/30 for 32 tokens.

Table 2: Performance and computational cost under different token on NVIDIA L20X.

Visual Tokens	VQA^{Text} (ms)	MMB (ms)	FLOPs (T)
576	344.49	791.35	7.63
128	114.47	136.00	1.67
64	105.55	116.61	1.17
32	102.92	99.44	0.96

3.2 Main Results

We compare HTR-VLM with state-of-the-art token reduction methods—FastV Chen et al. (2024), PDrop Xing et al. (2025), SparseVLM Zhang et al. (2025), TRIM Song et al. (2025), PruMerge+ Mao et al. (2025), VisionZip Yang et al. (2025), DART Wen et al. (2025), and DivPrune Alvar et al. (2025)—under three retention budgets of 128, 64, and 32 tokens (from 576 input tokens), corresponding to reduction rates of 77.8%, 88.9%, and 94.4%. As shown in table 1, HTR-VLM achieves the best average performance across all compression levels. All baseline results are taken from their respective publications to ensure fair comparison under the unified LLaVA-1.5-7B pipeline.

Table 3: Ablation study of the FTP, STR, and DATP modules on LLaVA-1.5-7B with an average of 64 retained tokens. Results are reported across 7 representative benchmarks, including the overall average score .

FTP	STR	DATP	VQA^{V2}	GQA	SQA^{IMG}	VQA^{Text}	POPE	MME	MMB	Avg.
			55.9	46.0	70.1	51.6	35.5	973.5	50.1	51.1
✓			78.1	61.3	66.7	55.7	86.0	1392.3	61.3	68.4
✓	✓		77.1	61.1	66.8	56.0	86.2	1403.4	63.1	68.6
✓	✓	✓	76.7	60.8	68.3	55.5	86.0	1453.2	64.3	69.2

Table 4: Ablation study on the pooling and up-sampling design choices in the FTP module on LLaVA-1.5-7B with an average of 64 retained tokens. Results are reported across 7 representative benchmarks, including the overall average score. The default setting is underlined.

Operation	Setting	VQA^{V2}	GQA	SQA^{IMG}	VQA^{Text}	POPE	MME	MMB	Avg.
Pool	Max	74.5	60.9	68.5	55.2	86.1	1406.2	64.4	68.6
	Average	76.7	60.8	68.3	55.5	86.0	1453.2	64.3	69.2
Up-sample	Replication	76.5	60.3	68.5	55.4	86.4	1459.8	63.1	69.0
	Interpolation	76.7	60.8	68.3	55.5	86.0	1453.2	64.3	69.2

We further evaluate practical inference latency on an NVIDIA L20X GPU with real-world input/output lengths (table 2). Vanilla LLaVA-1.5-7B requires 791.35 ms and 344.49 ms for average MMBench and TextVQA samples, respectively. Compressing to 128, 64, and 32 tokens reduces MMBench latency to 136.00, 116.61, and 99.44 ms (up to 7.9× speedup). For shorter TextVQA prompts, latency converges to 102.92 ms, primarily bounded by fixed vision encoding and autoregressive decoding overhead. The theoretical prefill FLOPs decrease correspondingly from 7.63T to 1.67T, 1.17T, and 0.96T. Despite latency saturation on short sequences, substantial wall-clock acceleration persists on bandwidth-bound hardware, as our method preserves full spatial interactions in attention while the reduced sequence length mitigates memory bottlenecks during both prefill and decoding.

3.3 Ablation Results

On main modules. We ablate the three main components under the 64-token retention setting: FFN Token Pooling with Residual Approximation (FTP), Salient Token Retention with Multi-Criteria Scoring (STR), and Depth-Aware Token Pruning (DATP), with the baseline constructed upon the FastV-based configuration. As reported in table 3, progressively introducing each module while keeping the compression ratio fixed yields consistent improvement.Specifically, FTP improves average performance by 33.9%, owing to its preservation of global contextual information in shallow layers through pooling-based approximation. STR further improves the average score by 0.3% by retaining critical tokens for full computation, thereby stabilizing task-targeted information flow. Finally, DATP enables a more favorable pruning allocation across deeper layers rather than pruning at a fixed layer, conforming to the observation that information redundancy increases with depth, and yielding an additional 0.9% gain.

On FTP. We further assess the pooling and up-sampling design choices in FTP. As shown in table 4, max pooling introduces more severe information loss by discarding non-maximum responses, while replication-based up-sampling simply copies pooled updates to all tokens within a window, disrupting feature smoothness and amplifying approximation errors. In contrast, combining average pooling with bilinear interpolation provides a more faithful approximation of local feature evolution, as bilinear interpolation serves as a natural approximate inverse to average pooling by smoothly redistributing aggregated information back to dense tokens.

On STR. As shown in table 5, sequentially incorporating attention-based and similarity-based salient token selection, together with retaining full FFN computation for selected tokens, enables more accurate feature updates for critical tokens in shallow layers, leading to accumulated improve-

Table 5: Effect of salient token selection criteria in the STR module on LLaVA-1.5-7B with an average of 64 retained tokens. ATTN and SIM denote retaining full computation for salient tokens selected according to eq. (5) and eq. (7), respectively.

Setting	VQA^{V2}	GQA	SQA^{IMG}	VQA^{Text}	POPE	MME	MMB	Avg.
Pool	76.5	60.7	68.5	55.2	86.0	1441.5	62.6	68.8
Pool + ATTN	76.4	60.5	68.1	54.6	85.7	1443.5	63.0	68.6
Pool + ATTN + SIM	76.7	60.8	68.3	55.5	86.0	1453.2	64.3	69.2

Table 6: Effect of incorporating raw vision features in the STR module on LLaVA-1.5-7B with an average of 128, 64 and 32 retained tokens.

Tokens	Setting	VQA^{V2}	GQA	SQA^{IMG}	VQA^{Text}	POPE	MME	MMB	Avg.
128	Ours	77.0	61.3	67.6	55.9	86.1	1462.4	62.9	69.1
	Ours w/o eq. (6)	76.7	60.2	69.1	56.1	84.7	1391.9	63.1	68.5
64	Ours	76.7	60.8	68.3	55.5	86.0	1453.2	64.3	69.2
	Ours w/o eq. (6)	75.2	59.2	68.4	54.3	83.5	1342.1	63.3	67.3
32	Ours	76.2	60.0	68.2	54.8	85.6	1427.9	63.8	68.6
	Ours w/o eq. (6)	73.6	60.2	67.9	55.9	85.5	1406.2	63.9	68.2

Table 7: Comparison of distinct pruning allocations module on LLaVA-1.5-7B with an average of 64 retained tokens. The default setting is underlined.

Layer	Ratio	VQA^{V2}	GQA	SQA^{IMG}	VQA^{Text}	POPE	MME	MMB	Avg.
(11, 20)	(11.1%, 5.0%)	76.7	60.8	68.3	55.5	86.0	1453.2	64.3	69.2
(11, 17)	(11.1%, 4.0%)	74.4	59.0	68.0	54.3	83.7	1405.8	60.0	67.1
(11, 18)	(11.1%, 4.3%)	73.9	59.1	65.5	52.5	84.0	1336.3	60.6	66.1
(11, 19)	(11.1%, 4.6%)	74.8	59.5	67.3	54.4	83.8	1377.5	61.3	67.1
(11, 21)	(11.1%, 5.5%)	75.0	59.9	67.3	53.3	84.1	1317.1	61.2	66.6

ment in average performance. table 6 further demonstrates the contribution of incorporating original visual features when computing attention-based importance scores (*i.e.*, eq. (6)), which introduces high-quality visual cues to correct potentially ambiguous attention relations in shallow layers, facilitating more reliable salient token selection.

On DATP. As shown in table 7, compared with multiple alternative allocation schemes, our configuration achieves the best overall accuracy, validating the effectiveness of the proposed depth-aware design. These results also provide insights into how pruning allocation in deeper VLM layers impacts performance, and may inspire further exploration of layer-wise pruning strategies.

4 Conclusion

In this paper, we explore token reduction for accelerating VLMs and propose HTR-VLM, a hybrid framework that combines shallow-layer token pooling with deep-layer token pruning to achieve a favorable efficiency–accuracy trade-off. The FFN Token Pooling with Residual Approximation (FTP) module reduces shallow-layer computation while maintaining full context flow via residual approximation. The Salient Token Retention with Multi-Criteria Scoring (STR) module preserves critical tokens through multiple complementary cues. The Depth-Aware Token Pruning (DATP) mechanism provides a principled allocation strategy for deep-layer pruning. Extensive experiments demonstrate that HTR-VLM delivers substantial inference acceleration while maintaining superior performance over state-of-the-art token reduction methods, offering a practical pathway toward efficient deployment of large-scale VLMs in resource-constrained scenarios.

REFERENCES

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

Xiang An, Yin Xie, Kaicheng Yang, Wenkang Zhang, Xiuwei Zhao, Zheng Cheng, Yirui Wang, Songcen Xu, Changrui Chen, Didi Zhu, et al. Llava-onevision-1.5: Fully open framework for democratized multimodal training. *arXiv preprint arXiv:2509.23661*, 2025.

Yuxuan Cai, Jiangning Zhang, Haoyang He, Xinwei He, Ao Tong, Zhenye Gan, Chengjie Wang, Zhucun Xue, Yong Liu, and Xiang Bai. Llava-kd: A framework of distilling multimodal large language models. pp. 239–249, 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. pp. 19–35, 2024.

Wei-Lin Chiang, Zhuohan Li, Ziqing Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. 36:49250–49267, 2023.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. 35:16344–16359, 2022.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. pp. 6904–6913, 2017.

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. pp. 6700–6709, 2019.

Yizhang Jin, Jian Li, Tianjun Gu, Yexin Liu, Bo Zhao, Jinxiang Lai, Zhenye Gan, Yabiao Wang, Chengjie Wang, Xin Tan, et al. Efficient multimodal large language models: A survey. *Visual Intelligence*, 3(1):27, 2025.

Shiyao Li, Yingchun Hu, Xuefei Ning, Xihui Liu, Ke Hong, Xiaotao Jia, Xiuhong Li, Yaqi Yan, Pei Ran, Guohao Dai, et al. Mbq: Modality-balanced quantization for large vision-language models. pp. 4167–4177, 2025a.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne 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, 2023.

Yucheng Li, Huiqiang Jiang, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Amir H Abdi, Dongsheng Li, Jianfeng Gao, Yuqing Yang, et al. Mminference: Accelerating pre-filling for long-context vlms via modality-aware permutation sparse attention. *arXiv preprint arXiv:2504.16083*, 2025b.

Yinan Liang, Ziwei Wang, Xiuwei Xu, Jie Zhou, and Jiwen Lu. Efficientllava: Generalizable auto-pruning for large vision-language models. pp. 9445–9454, 2025.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. pp. 34892–34916, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. pp. 26296–26306, 2024a.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? pp. 216–233. Springer, 2024b.

Pan Lu, Swaroop Mishra, Tanglin 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. 35:2507–2521, 2022.

Junzhu Mao, Yang Shen, Jinyang Guo, Yazhou Yao, Xiansheng Hua, and Hengtao Shen. Prune and merge: Efficient token compression for vision transformer with spatial information preserved. *IEEE Transactions on Multimedia*, 2025.

Zhenwei Shao, Mingyang Wang, Zhou Yu, Wenwen Pan, Yan Yang, Tao Wei, Hongyuan Zhang, Ning Mao, Wei Chen, and Jun Yu. Growing a twig to accelerate large vision-language models. pp. 20064–20074, October 2025.

Fangxun Shu, Yue Liao, Le Zhuo, Chenning Xu, Lei Zhang, Guanghao Zhang, Haonan Shi, Long Chen, Tao Zhong, Wanggui He, et al. Llava-mod: Making llava tiny via moe knowledge distillation. *arXiv preprint arXiv:2408.15881*, 2024.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. pp. 8317–8326, 2019.

Dingjie Song, Wenjun Wang, Shunian Chen, Xidong Wang, Michael X Guan, and Benyou Wang. Less is more: A simple yet effective token reduction method for efficient multi-modal llms. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 7614–7623, 2025.

Lin Song, Yukang Chen, Shuai Yang, Xiaohan Ding, Yixiao Ge, Ying-Cong Chen, and Ying Shan. Low-rank approximation for sparse attention in multi-modal llms. pp. 13763–13773, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. 30, 2017.

Changyuan Wang, Ziwei Wang, Xiuwei Xu, Yansong Tang, Jie Zhou, and Jiwen Lu. Q-vlm: Post-training quantization for large vision-language models. 37:114553–114573, 2024.

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. *arXiv preprint arXiv:2502.11494*, 2025.

Penghao Wu, Lewei Lu, and Ziwei Liu. Streamline without sacrifice–squeeze out computation redundancy in lmm. *arXiv preprint arXiv:2505.15816*, 2025a.

Zimeng Wu, Jiaxin Chen, and Yunhong Wang. Unified knowledge maintenance pruning and progressive recovery with weight recalling for large vision-language models. volume 39, pp. 8550–8558, 2025b.

Zimeng Wu, Donghao Wang, Chaozhe Jin, Jiaxin Chen, and Yunhong Wang. Probe and skip: Self-predictive token skipping for efficient long-context llm inference. *arXiv preprint arXiv:2601.13155*, 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. pp. 14593–14603, 2025.

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. pp. 19792–19802, 2025.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. 36: 11809–11822, 2023.

Weihao Ye, Qiong Wu, Wenhao Lin, and Yiyi Zhou. Fit and prune: Fast and training-free visual token pruning for multi-modal large language models. volume 39, pp. 22128–22136, 2025.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *National Science Review*, 11(12):nwae403, 2024.

Quan-Sheng Zeng, Yunheng Li, Qilong Wang, Peng-Tao Jiang, Zuxuan Wu, Ming-Ming Cheng, and Qibin Hou. A glimpse to compress: Dynamic visual token pruning for large vision-language models. *arXiv preprint arXiv:2508.01548*, 2025.

Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 46(8):5625–5644, 2024.

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. 2025.

Baichuan Zhou, Ying Hu, Xi Weng, Junlong Jia, Jie Luo, Xien Liu, Ji Wu, and Lei Huang. Tinyllava: A framework of small-scale large multimodal models. *arXiv preprint arXiv:2402.14289*, 2024.