Structured Redundancy Modeling for Efficient Visual Token Pruning in High-Resolution MLLMs

Juwon Song[1], Woohyeong Kim[2], and Kyeongbo Kong[2⋆]

[1] LG Electronics, Seoul, Republic of Korea
wn5649@sogang.ac.kr
[2] Pusan National University, Busan, Republic of Korea
{zovw8179, kbkong}@pusan.ac.kr
https://github.com/cvsp-lab/SFPruner

Abstract. Recent high-resolution Multimodal Large Language Models (MLLMs) generate thousands of visual tokens per input, leading to a visual token explosion that introduces severe latency bottlenecks. While token pruning mitigates this issue, state-of-the-art subset-optimization methods typically rely on iterative subset construction to jointly capture visual diversity and instruction relevance. As visual token counts scale, this sequential dependency introduces significant selection overhead, severely limiting the translation of theoretical FLOPs reductions into actual wall-clock speedups. To address this limitation, we propose Single-Forward Pruner (SFPruner), a structural reformulation of visual token pruning that embeds redundancy control directly into the scoring space, bypassing the need for iterative combinatorial optimization. Our non-iterative framework achieves redundancy-aware importance selection in a single forward pass through two complementary mechanisms. First, to attenuate redundancy at the covariance level, we introduce a semantics-guided ridge leverage scheme. By integrating instruction relevance and visual saliency, this mechanism suppresses dominant covariance directions and mitigates representation bias. Second, ranking-based directional masking resolves residual overlap through asymmetric similarity competition, where higher-scoring tokens explicitly suppress redundant lower-scoring alternatives via parallel tensor operations. Extensive evaluations demonstrate that our approach maintains stable selection costs, reducing the token selection process by up to 110 ms (from 112.4 ms to just 2.5 ms at 512 tokens in Qwen2.5-VL). This structural efficiency successfully translates theoretical token reductions into tangible inference speedups while preserving highly competitive performance against state-of-the-art techniques under aggressive compression.

1 Introduction

Recent multimodal large language models (MLLMs) [1,2,3] extend beyond text processing to perform fine-grained visual reasoning over high-resolution images

⋆ Corresponding author.

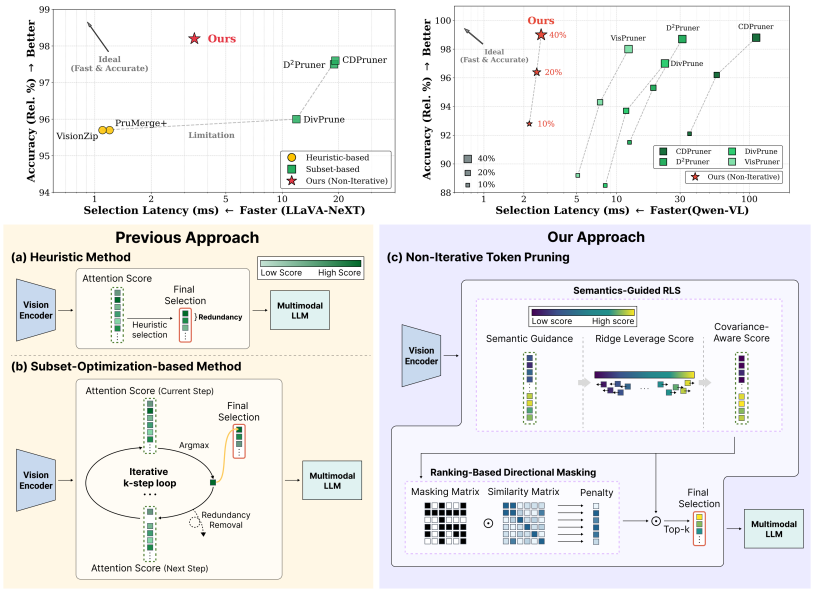

Fig. 1: **Conceptual comparison of token selection frameworks. (a)** Heuristic (Top-K) relies on isolated scores, lacking structural redundancy control. **(b)** Traditional sequential search mitigates redundancy but introduces iterative dependencies. **(c)** Our framework enables efficient single-pass selection via SG-RLS and directional masking, eliminating iteration overhead.

and long video sequences. Architectures like LLaVA-NeXT [4] and Qwen-VL [2] generate thousands of visual tokens per input using dynamic-resolution mechanisms. However, since self-attention scales quadratically with token length, this leads to severe latency and memory bottlenecks, limiting practical deployment.

Vision token pruning addresses the challenge of computational efficiency by selecting informative tokens and removing redundant ones prior to costly attention operations [5,6,7]. Existing approaches can be broadly categorized into two groups, each exhibiting a distinct trade-off between speed and performance, as illustrated in the top panels of Fig. 1. Recent empirical evidence further supports this dichotomy: attention-based pruning tends to be effective for simple images with concentrated visual evidence, whereas diversity-oriented pruning is more beneficial for complex images with distributed visual cues [8]. Heuristic methods (Fig. 1a) employ ranking or filtering strategies based on attention or similarity signals [9,10,11]. As shown in the top-left panel, these methods offer high speed but fail to fully eliminate spatial redundancy and often experience substantial performance degradation under aggressive pruning. In contrast, subset-optimization methods (Fig. 1b) explicitly model both importance and diversity to better preserve accuracy, using formulations such as DPP, diversity maximization, or graph-based MWIS [12,13,14]. However, these approaches depend on iterative selection procedures, resulting in computational overhead that increases significantly with the number of tokens. This limitation is evident in

the top-right panel of Fig. 1, where the latency of iterative subset-optimization methods rises sharply as the token retention ratio increases.

In this work, we propose Single-Forward Pruner (SFPruner), a structural approach to token pruning (**Fig. 1c**). Instead of iterative subset construction, we embed redundancy control directly into the scoring process. Our SFPruner decomposes redundancy handling into two mechanisms in a single forward pass. First, we apply **covariance-level redundancy attenuation** via a semantics-guided ridge leverage scheme [15,16], integrating instruction relevance and visual saliency to modulate the structural scores. This step suppresses dominant features and regularizes scores. Second, we use **ranking-based directional masking**, where higher scores suppress redundant lower ones through asymmetric similarity competition. This resolves token overlap without iterative updates, enabling direct top-k selection.

By controlling redundancy at both covariance and pairwise levels, our approach enables efficient, redundancy-aware selection in a single forward pass. Leverage-based reweighting mitigates dominance bias, and directional masking enforces structured competition. As token counts grow, our non-iterative design avoids the sequential bottlenecks of previous methods. As demonstrated in Fig. 1, our method maintains constant, minimal selection costs, reducing latency from 112.4 ms to just 2.5 ms when selecting 512 tokens in Qwen2.5 [2]. This ensures that FLOP reductions translate directly into real-world speedup without sacrificing performance, even under aggressive pruning.

Our main contributions are:

- **A structural reformulation of visual token pruning:** We show that redundancy-aware selection can be achieved in feature space without combinatorial optimization, by embedding redundancy control into the scoring process.
- **Covariance-level redundancy attenuation:** We introduce a semantics-guided leverage reweighting mechanism that integrates instruction relevance and visual saliency, suppressing dominant covariance directions before selection.
- **Ranking-based directional masking:** We propose an asymmetric masking strategy, where higher-scoring tokens suppress redundant lower-scoring tokens in a single tensor operation, removing the need for iterative subset construction.
- **Scalable efficiency for high-resolution inputs:** Our method maintains stable selection costs. Reducing selection latency from 112.4 ms to 2.5 ms translates theoretical FLOP reductions into practical acceleration, preserving highly competitive reasoning capability (retaining up to 99.0% relative performance).

2 Related Works

2.1 Vision Token Pruning Methods

Pruning techniques that reduce visual token redundancy are widely adopted to enhance the inference efficiency of Multimodal Large Language Models (MLLMs), largely because they require no additional training. Recent approaches broadly fall into two categories: heuristic methods, which rely on direct scoring signals, and subset-optimization-based methods, which explicitly model relationships between tokens.

Heuristic Selection and Merging: Numerous attention-based methods select tokens relying strictly on local attention weights, often ignoring semantic redundancy [17,10,18,6,7]. Attempts to extract global features [19] or dynamically adjust token counts [20] tend to favor dominant objects, risking the loss of fine-grained cues. Merging techniques such as ToMe [21] and VisionZip [11] reduce redundancy through token merging, aggregation, or recognition-oriented importance scoring. However, these vision-only compression strategies are primarily designed for generic visual redundancy reduction and may obscure fine-grained visual details critical for instruction-conditioned reasoning in high-resolution MLLMs. In contrast, MLLM token pruning requires preserving query-relevant visual evidence while suppressing redundant or task-irrelevant tokens.

Subset-Optimization-Based Token Selection: To overcome these limitations, recent works formulate token selection as combinatorial optimization—e.g., maximizing importance and diversity via DPP [12], MMDP [13], or graph-based MWIS formulations [14]. As these problems are NP-hard, **sequential greedy search** is universally adopted for polynomial-time approximation. This requires an iterative k-step loop to update states after each selection. While effective for redundancy-aware selection, this sequential paradigm introduces a severe latency bottleneck that scales linearly with sequence length in high-resolution settings.

2.2 Ridge Leverage Score in Neural Architectures

Based on Randomized Numerical Linear Algebra (RandNLA) [22], the Ridge Leverage Score (RLS) is a statistical metric that balances dominant patterns and sparse details by softening linear dependencies within data through ridge regularization (λ) [15,16]. Notably, RLS provides the mathematical advantage of simultaneously quantifying the global diversity and feature-space uniqueness of all data points via matrix inversion, thereby circumventing the need for iterative state updates. Due to these structural properties, RLS has become widely adopted for dimensionality reduction and computational optimization in deep learning architectures. In Natural Language Processing (NLP), it has been used to compress the effective dimensions of transformer blocks, as demonstrated in MoDeGPT [23]. Furthermore, a recent attention approximation study [24] introduced RLS sampling to quantum algorithmic environments, reducing the standard $\mathcal{O}(N^2)$ attention complexity to sublinear time. As such, RLS has emerged as a core technique for extracting principal components without compromising the information spectrum.

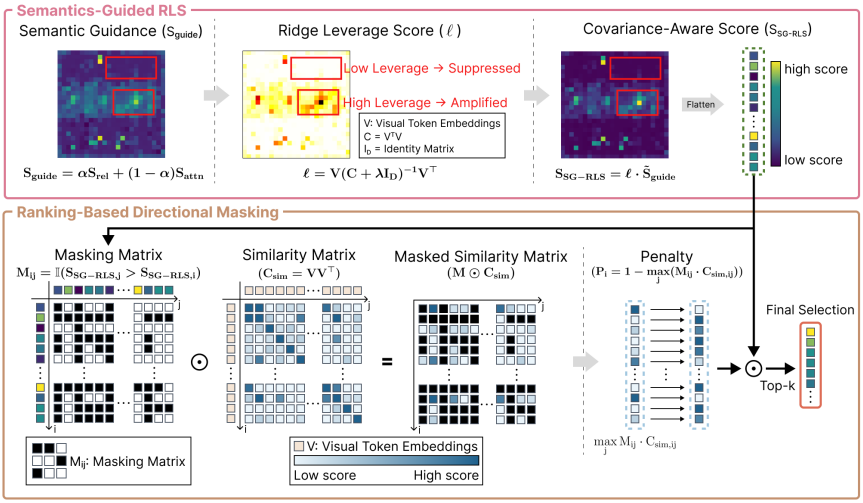

Fig. 2: **Pipeline of the non-iterative token pruning framework. (Top) Semantics-Guided RLS (SG-RLS)**: The semantic guidance (S_{guide}) is modulated by the leverage score map (ℓ) to suppress redundant background features, yielding a covariance-aware score ($S_{\text{SG-RLS}}$). **(Bottom) Ranking-Based Directional Masking**: A ranking mask (M) is applied to the pairwise similarity matrix. Tokens receive a directional suppression penalty (P_i) based on their maximum similarity to higher-priority competitors, extracting the final pruned subset through single-pass tensor operations.

3 Proposed Method: Single-Forward Pruner (SFPruner)

In this section, we present SFPruner, a non-iterative token pruning framework that bypasses sequential subset construction by embedding redundancy modeling directly into the scoring process. Our method consists of three key steps: (1) establishing base token importance via instruction-aware semantic guidance (§3.1); (2) attenuating global covariance-level redundancy using a semantics-guided ridge leverage formulation (§3.2); and (3) resolving residual local overlap via ranking-based directional masking, where higher-scoring tokens asymmetrically suppress correlated lower-scoring alternatives (§3.3).

3.1 Semantic Guidance: Textual Relevance and Visual Saliency

We first estimate a base importance score for each visual token, referred to as the semantic guidance. This guidance integrates instruction relevance and intrinsic visual saliency, providing an instruction-aware base score that is subsequently modulated by covariance-level redundancy modeling.

Let $V \in \mathbb{R}^{N \times D}$ denote the matrix of L_2-normalized visual token embeddings, where each row $v_i \in \mathbb{R}^{1 \times D}$ corresponds to a visual token. Let $q \in \mathbb{R}^{1 \times D}$ denote the projected text embedding.

Textual relevance. We measure instruction relevance using cosine similarity between each visual token and the text embedding, followed by temperature-scaled normalization:

$$S_{\text{rel},i} = \frac{\exp\left(\tau \frac{v_i q^\top}{\|v_i\| \, \|q\|}\right)}{\sum_{j=1}^{N} \exp\left(\tau \frac{v_j q^\top}{\|v_j\| \, \|q\|}\right)}, \tag{1}$$

where τ controls the sharpness of the relevance distribution.

Visual saliency. To capture intrinsic visual importance independent of the instruction, we extract a saliency score from the vision encoder self-attention. Specifically, we use the attention weight from the global token (e.g., CLS) to each visual token:

$$S_{\text{attn},i} = \text{Attention}(\text{CLS} \to v_i). \tag{2}$$

In the absence of a canonical CLS token (e.g., Qwen2.5 [2]), we construct a proxy for global saliency by aggregating the self-attention weights across all spatial tokens.

Semantic guidance fusion. We combine textual relevance and visual saliency through a linear fusion:

$$S_{\text{guide},i} = \alpha S_{\text{rel},i} + (1-\alpha) S_{\text{attn},i}, \quad \alpha \in [0,1]. \tag{3}$$

For architectures lacking a dedicated text encoder (e.g., Qwen2.5 [2]), we rely solely on the visual saliency as the semantic guidance. To ensure numerical stability and comparable scaling across inputs, we apply Min-Max normalization:

$$\tilde{S}_{\text{guide},i} = \frac{S_{\text{guide},i} - \min_j S_{\text{guide},j}}{\max_j S_{\text{guide},j} - \min_j S_{\text{guide},j} + \epsilon}. \tag{4}$$

The normalized semantic guidance $\tilde{S}_{\text{guide}}$ serves as the base importance score for the covariance-level redundancy attenuation described in Sec. 3.2.

3.2 Covariance-Level Attenuation: Semantics-Guided Ridge Leverage Score

Sequential subset selection methods suppress redundancy by progressively updating the residual feature space after each token is selected. While effective, such stepwise dependency introduces non-trivial execution overhead. We instead perform redundancy attenuation globally through a ridge-regularized leverage formulation, eliminating the need for iterative updates.

We define the feature covariance matrix

$$C = V^\top V \in \mathbb{R}^{D \times D}. \tag{5}$$

The ridge leverage score of token i is defined as

$$\ell_i = v_i \left(C + \lambda I_D\right)^{-1} v_i^\top, \tag{6}$$

where λ is a small regularization constant.

The inverse covariance $(C + \lambda I_D)^{-1}$ attenuates dominant eigendirections of the feature space, reducing the influence of globally redundant patterns. Tokens aligned with high-energy subspaces receive lower leverage scores, while structurally distinctive tokens obtain higher scores.

To incorporate task-awareness, we modulate the structural score with the semantic guidance introduced in Section 3.1. The final covariance-level redundancy score is defined as

$$S_{\text{SG-RLS},i} = \ell_i \cdot \tilde{S}_{\text{guide},i}. \tag{7}$$

The multiplicative formulation acts as a soft-AND gate, requiring both structural uniqueness and semantic relevance for a token to receive a high score. Consequently, tokens that are structurally distinctive but task-irrelevant, or semantically relevant but highly redundant, are naturally down-weighted. This formulation separates geometric uniqueness from semantic importance: ridge leverage captures global structural redundancy, while the semantic guidance suppresses task-irrelevant outliers.

Importantly, the entire computation consists of parallel matrix operations and a single matrix inversion. While the covariance matrix C is intrinsically $D \times D$, we leverage linear algebraic duality via the Woodbury matrix identity to establish an equivalence between the feature covariance and the $N \times N$ Gram matrix. This duality allows us to flexibly perform the inversion on the smaller dimension between the sequence length N and the feature dimension D. Consequently, we can circumvent the computationally prohibitive $N \times N$ inversion in high-resolution regimes ($N > D$) while completely removing iterative dependencies. Detailed derivations for this identity are provided in the supplementary material.

3.3 Ranking-Based Directional Masking

While SG-RLS mitigates redundancy at the covariance level, subset-based optimization methods further impose pairwise exclusion: once a token is selected, highly similar alternatives are suppressed. We incorporate this pairwise redundancy control without introducing iterative updates.

Let $C_{\text{sim}} = VV^\top \in \mathbb{R}^{N \times N}$ denote the cosine similarity matrix between tokens. Higher values indicate stronger semantic overlap.

We define a directional masking mechanism based on the covariance-level scores $S_{\text{SG-RLS}}$ from Section 3.2. For each token i, suppression is applied only from tokens with strictly higher scores. Formally, we construct a masking matrix

$$M_{ij} = \mathbb{I}(S_{\text{SG-RLS},j} > S_{\text{SG-RLS},i}), \tag{8}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function. This mask enforces asymmetric competition: higher-ranked tokens may suppress lower-ranked ones, but not vice versa.

The directional suppression penalty for token i is then

$$P_i = 1 - \max_j \left(M_{ij} \cdot C_{\text{sim},ij} \right). \tag{9}$$

By construction, $M_{ii} = 0$, ensuring that the maximum is non-negative. This prevents negatively correlated tokens from increasing the score, i.e., $P_i \leq 1$. If no higher-scoring token exhibits strong similarity, P_i remains close to 1; otherwise, the penalty increases proportionally to the maximum overlap. The final redundancy-aware score is computed as

$$S_{\text{final},i} = S_{\text{SG-RLS},i} \cdot P_i. \tag{10}$$

This formulation enforces structured pairwise suppression through fully parallel tensor operations. All computations consist of matrix multiplications, element-wise comparisons, and row-wise reductions, eliminating iterative subset updates while maintaining directional redundancy control. The final subset is obtained via a single Top-K operation.

4 Experiments

4.1 Experimental Setup

We evaluate the effectiveness and structural efficiency of the proposed non-iterative token pruning framework across multiple Multimodal Large Language Model (MLLM) architectures and diverse vision-language benchmarks.

- **Models.** To demonstrate generality, we evaluate three representative MLLM families: LLaVA-NeXT-7B [4], optimized for high-resolution image processing; Qwen2.5-VL-7B [25], a recent high-capacity vision-language model; and LLaVA-Video-7B [26], which supports multi-frame video inference. These models cover both image-centric and temporal multimodal settings.
- **Benchmarks.** For image understanding, we evaluate on VQAv2 [27], GQA [28], TextVQA [29], MME [30], and ScienceQA (SQA) [31]. For video reasoning, we use VideoMME [32], MLVU [33], and LongVideoBench [34]. These benchmarks collectively test fine-grained reasoning, instruction alignment, and large-scale visual token processing.
- **Evaluation Metrics.** We report absolute task accuracy and relative performance retention (Rel. %) with respect to the corresponding unpruned vanilla models. To evaluate computational behavior, we measure the wall-clock Selection Latency (ms), which isolates the execution time of the token selection module prior to the LLM forward pass. The reported latency includes the complete pruning pipeline (semantic guidance, SG-RLS scoring, directional masking, and Top-K selection), while excluding the vision encoder and LLM forward pass. All latency measurements are performed on a single NVIDIA RTX 4090 GPU with synchronized CUDA timing to ensure consistent profiling.

Detailed parameter configurations (α and λ) and additional sensitivity analyses are provided in the supplementary material.

Method	VQA^{v2}	GQA	SQA^{IMG}	TextVQA	POPE	MME	MMB	MMB^{CN}	MMVet	Rel. (%)	Time (ms) ↓
Vanilla 5 × 576 *Tokens*											
LLaVA-NeXT-7B	81.3	62.5	67.5	60.3	86.8	1511.8	65.8	57.3	40.0	100.0	0.00
Retain 5 × 128 *Tokens*											
SparseVLM (ICML'25) [7]	79.2	61.2	67.6	59.7	85.3	1456.8	65.9	**58.6**	36.1	97.9	19.0
PruMerge+ (ICCV'25) [10]	78.2	60.8	67.8	54.9	85.3	1480.2	64.6	57.3	32.7	98.3	1.2
VisionZip (CVPR'25) [11]	79.1	61.2	**68.1**	**59.9**	86.0	1493.4	65.8	58.1	38.9	99.5	1.1
DivPrune (CVPR'25) [13]	79.3	61.9	67.8	57.0	86.9	1469.7	65.8	57.3	38.0	98.5	33.8
CDPruner (NeurIPS'25) [12]	79.9	62.6	67.9	58.4	**87.3**	1474.2	66.3	57.5	**41.9**	99.9	35.2
SFPruner (Ours)	**80.1**	**62.7**	67.6	59.0	87.1	**1480.3**	**67.1**	57.6	41.0	**100.0**	**3.5**
Retain 5 × 64 *Tokens*											
SparseVLM (ICML'25) [7]	74.6	57.9	67.2	56.5	76.9	1386.1	63.1	**56.7**	32.8	93.3	18.2
PruMerge+ (ICCV'25) [10]	75.3	58.8	**68.1**	54.0	79.5	1444.2	63.0	55.6	31.4	95.7	1.2
VisionZip (CVPR'25) [11]	76.2	58.9	67.5	**58.8**	82.3	1397.1	63.3	55.6	35.8	95.7	1.1
DivPrune (CVPR'25) [13]	77.2	61.1	67.7	56.2	84.7	1423.3	63.9	55.7	34.8	96.0	11.9
CDPruner (NeurIPS'25) [12]	78.4	**61.6**	67.8	57.4	**87.2**	1453.1	**65.5**	55.7	**37.9**	97.6	19.2
SFPruner (Ours)	**78.5**	61.5	67.8	58.3	86.7	**1507.6**	65.1	56.6	**37.9**	**98.2**	**3.4**
Retain 5 × 32 *Tokens*											
PruMerge (ICCV'25) [10]	70.5	56.2	66.9	50.3	71.1	1289.6	58.0	48.9	29.3	87.7	1.1
VisionZip (CVPR'25) [11]	71.4	55.2	**67.9**	55.0	74.9	1327.8	58.6	50.4	32.3	90.0	1.1
DivPrune (CVPR'25) [13]	75.0	59.3	67.1	54.1	80.0	1356.6	62.9	53.7	32.0	92.9	7.8
CDPruner (NeurIPS'25) [12]	**76.7**	**60.8**	67.5	55.4	**86.8**	1425.1	64.2	53.8	36.2	95.5	8.9
SFPruner (Ours)	76.6	60.3	**67.9**	**57.0**	86.6	**1435.3**	**64.3**	**55.0**	**37.2**	**96.4**	**3.4**

Table 1: Performance comparison on LLaVA-NeXT-7B at different token retention budgets. Relative performance (**Rel.**) is normalized to the unpruned vanilla model. Light blue rows indicate heuristic-based methods, and light green rows indicate optimization-based methods.

4.2 Performance on LLaVA-NeXT-7B: High-Resolution Multi-Patch Setting

We evaluate SFPruner on LLaVA-NeXT-7B [4] under high-resolution settings. Using the dynamic AnyRes strategy, each image is divided into up to five patches, yielding 2,880 visual tokens (5×576). To analyze performance under constrained budgets, we progressively reduce the 576 tokens per patch to 128, 64, and 32 tokens, corresponding to total retention budgets of 640, 320, and 160 tokens. Table 1 reports task accuracy, relative performance retention (Rel.%), and selection latency.

Selection Efficiency. Subset-optimization methods such as DivPrune and CDPruner rely on sequential search, resulting in noticeable selection overhead (e.g., 33.8 ms and 35.2 ms at 640-token retention). In contrast, SFPruner performs covariance-level attenuation and directional masking via parallel tensor operations, achieving a stable selection latency of 3.5 ms across retention budgets. Importantly, the selection latency remains nearly constant as the number of retained tokens decreases, indicating the absence of iterative dependency.

Performance Retention. At 640-token retention (approximately 22% of the original tokens), our method achieves 100.0% relative performance, matching or slightly exceeding the unpruned baseline within evaluation variance. At 320-token retention (11% of tokens), performance remains at 98.2%, comparable to optimization-based methods while requiring substantially lower selection time. Under more aggressive compression (160 tokens, 5.5% retention), our method retains 96.4% performance, remaining competitive with CDPruner (95.5%) while

Method	TextVQA	AI2D	HBench	MME	MMB	MMB^{CN}	POPE	MMStar	Rel. (%)	Time (ms) ↓
Vanilla 100% Tokens										
Qwen2.5-VL-7B	85.3	80.8	64.3	2316.0	79.7	81.8	86.4	56.7	100.0	-
Retain 40% Tokens (Avg. 512 Tokens)										
PruMerge (ICCV'25) [10]	82.1	79.8	62.6	2287.0	78.4	80.0	86.0	55.6	98.1	**1.2**
VisPruner (ICCV'25) [9]	82.6	79.9	61.7	2297.5	78.7	79.9	85.4	55.3	98.0	12.3
VisionZip (CVPR'25) [11]	82.6	79.5	61.2	2306.5	78.3	**80.5**	**86.1**	55.2	98.0	1.1
DivPrune (CVPR'25) [13]	82.3	78.6	61.4	2230.6	78.5	80.0	84.5	54.7	97.0	23.1
CDPruner (NeurIPS'25) [12]	83.7	79.6	63.4	2302.2	**79.2**	80.3	**86.1**	**55.9**	98.8	112.4
VisionSelector [35]	**84.5**	79.5	**63.8**	2315.8	77.4	79.9	85.8	53.3	98.1	1.2
D^{2}Pruner (AAAI'26) [36]	83.2	**82.9**	**63.7**	2305.3	78.2	79.6	85.3	54.9	98.7	31.2
SFPruner (Ours)	84.0	79.9	63.1	**2319.3**	79.0	79.7	**86.5**	**55.9**	**99.0**	2.5
Retain 20% Tokens (Avg. 256 Tokens)										
PruMerge (ICCV'25) [10]	74.7	77.2	59.1	2280.1	77.6	77.7	84.5	52.8	94.6	**1.1**
VisPruner (ICCV'25) [9]	75.3	77.4	59.0	2213.9	77.5	78.3	84.5	52.2	94.3	7.5
VisionZip (CVPR'25) [11]	75.9	77.6	59.1	2276.5	77.4	78.5	84.5	53.5	95.1	**1.1**
DivPrune (CVPR'25) [13]	76.5	76.6	58.1	2203.7	76.9	77.5	83.6	51.8	93.7	11.8
CDPruner (NeurIPS'25) [12]	79.1	78.4	**61.1**	2264.5	77.9	**79.5**	84.9	53.4	96.2	56.7
VisionSelector [35]	**83.4**	78.1	60.7	2283.4	76.2	77.1	84.6	51.2	95.7	1.1
D^{2}Pruner (AAAI'26) [36]	78.6	**79.4**	60.2	2280.5	76.1	78.6	83.0	52.1	95.3	18.9
SFPruner (Ours)	79.8	78.3	60.2	**2295.7**	**78.1**	79.4	**85.7**	**53.7**	**96.5**	2.5

Table 2: **Main Results on Qwen2.5-VL-7B.** Performance comparison across various token pruning methods at different retention ratios. Relative performance (**Rel. (%)**) is normalized to the unpruned baseline. Light blue rows indicate heuristic-based methods, and light green rows represent optimization-based methods.

maintaining significantly lower selection latency. Compared to heuristic approaches, which exhibit larger performance drops under high compression, the proposed framework maintains stable reasoning accuracy. At the same time, it achieves efficiency close to lightweight heuristics, without relying on sequential subset construction.

4.3 Performance on Qwen2.5-VL: High-Resolution Single Sequence

We further evaluate the proposed method on Qwen2.5-VL-7B to examine scalability under denser visual sequences. Unlike LLaVA-NeXT, which processes high-resolution images by dividing them into independent patches, Qwen2.5-VL encodes the entire image as a single continuous token sequence. This architectural setting increases the number of tokens competing within a single attention window and therefore presents a more demanding redundancy modeling scenario.

As shown in Table 2, optimization-based methods incur noticeable selection overhead under this dense single-sequence regime. At 40% token retention (512 tokens), CDPruner requires 112.4 ms, DivPrune requires 23.1 ms, and D^{2}Pruner requires 31.2 ms for subset selection. At 20% retention (256 tokens), their latency decreases proportionally to the reduced target token count (e.g., CDPruner: 56.7 ms; DivPrune: 11.8 ms; D^{2}Pruner: 18.9 ms), reflecting the iterative nature of their selection process. In contrast, our method maintains an identical selection time of 2.5 ms for both 40% and 20% retention settings. This retention-invariant behavior arises from the absence of iterative subset construction, since

Method	Video-MME			MLVU	LongVideoBench			MMVBench Rel. (%)		Time
	Short	Medium	Long	Avg.	Val	Perception	Relation			(ms) ↓
Vanilla 100% Tokens										
LLaVA-Video-7B	67.1	62.8	47.6	60.3	61.1	65.0	53.8	58.8	100.0	-
Retain 512 Tokens per Frame										
VisPruner (ICCV'25) [9]	64.0	50.8	44.0	57.5	53.5	52.5	53.5	56.5	91.0	12.3
VisionZip (CVPR'25) [11]	65.0	51.5	44.5	58.5	54.2	53.0	54.0	58.1	92.4	**1.1**
DivPrune (CVPR'25) [13]	66.6	**53.0**	45.0	60.1	55.6	**55.2**	55.5	58.1	94.5	131.0
CDPruner (NeurIPS'25) [12]	64.6	52.0	45.7	60.0	54.9	54.9	54.7	57.1	93.5	92.0
D^2Pruner (AAAI'26) [36]	65.5	52.5	46.0	59.5	55.0	54.5	54.8	57.5	93.8	31.2
SFPruner (Ours)	**67.4**	52.6	**46.0**	**60.2**	**55.7**	**55.2**	**55.6**	**58.3**	**94.9**	3.1
Retain 256 Tokens per Frame										
VisPruner (ICCV'25) [9]	60.5	47.0	41.5	54.0	50.5	49.5	49.5	53.0	85.3	7.5
VisionZip (CVPR'25) [11]	61.5	48.0	42.0	55.0	51.0	50.5	50.0	54.5	86.8	**1.1**
DivPrune (CVPR'25) [13]	64.1	51.2	45.3	**58.3**	**54.1**	**56.3**	52.6	54.9	91.9	64.0
CDPruner (NeurIPS'25) [12]	61.1	49.6	46.0	57.2	52.4	52.9	51.9	52.8	89.4	44.0
D^2Pruner (AAAI'26) [36]	62.1	49.9	45.4	57.5	52.8	52.0	52.2	55.2	90.0	18.9
SFPruner (Ours)	**64.1**	**51.8**	**47.0**	58.2	53.1	53.2	**53.0**	**56.7**	**92.1**	3.1

Table 3: **Main Results on Video MLLM Benchmarks.** Performance comparison using the LLaVA-Video-7B model across multiple frames. Relative performance (**Rel. (%)**) is normalized to the unpruned baseline. Light blue rows indicate heuristic-based methods, and light green rows represent optimization-based methods.

redundancy modeling is resolved through parallel covariance-level attenuation and ranking-based directional masking.

Despite the reduced computational overhead, the proposed framework preserves competitive reasoning performance. At 40% retention, our method achieves 99.0% relative performance, comparable to CDPruner (98.8%) and D^2Pruner (98.7%). Furthermore, compared with the recently proposed *learned* VisionSelector, SFPruner consistently achieves higher relative performance at both 40% (99.0% vs. 98.1%) and 20% (96.5% vs. 95.7%) retention, despite requiring no additional training. Under more aggressive compression (20% retention), performance remains at 96.5%, again comparable to optimization-based methods while requiring substantially lower selection latency. These results demonstrate that the proposed non-iterative formulation remains stable under long continuous token sequences, achieving redundancy-aware pruning without the latency growth associated with sequential subset construction.

4.4 Performance on Video Sequences: Multi-Frame Processing

We further evaluate scalability under multi-frame settings using LLaVA-Video-7B. Video inference introduces substantially larger token counts due to temporal stacking, making redundancy modeling more challenging than in single-image settings. Table 3 reports performance and selection latency under 512 and 256 tokens per frame.

Sequential optimization-based methods incur noticeable selection overhead in this regime. At 512 tokens per frame, DivPrune requires 131.0 ms, CDPruner

requires 92.0 ms, and D^{2}Pruner requires 31.2 ms. In contrast, our method maintains a stable selection latency of 3.1 ms, comparable to lightweight heuristic approaches. Moreover, the latency remains unchanged when reducing the retention budget from 512 to 256 tokens per frame, indicating retention-invariant pruning overhead under batch processing.

Despite the reduced computational cost, the proposed framework preserves competitive spatio-temporal reasoning accuracy. At 512 tokens per frame, our method achieves 94.9% relative performance, the highest among compared methods. Under stronger compression (256 tokens per frame), performance remains at 92.1%, comparable to optimization-based baselines while maintaining significantly lower selection latency. These results demonstrate that the proposed non-iterative formulation remains stable under temporal batching, achieving redundancy-aware pruning without the latency growth associated with sequential subset construction.

4.5 Extreme Scalability: High-Resolution Stress Test

To further evaluate structural scalability under extreme conditions, we conduct a stress test by scaling up the input resolution to an ultra-high level on Qwen2.5-VL using the entire MME dataset [30], producing a continuous sequence of 9,216 visual tokens ($D = 3584$). We apply a 30% retention budget (approximately 2,770 tokens) and profile peak GPU memory, isolated pruning time, end-to-end prefill latency, and total inference time. The results are summarized in Table 4. At this scale, sequential subset-optimization methods incur substantial pruning overhead. For example, CDPruner requires 576 ms for subset selection, while DivPrune requires 458 ms. In contrast, the proposed method completes pruning in 28 ms. Importantly, the impact of pruning overhead becomes more pronounced as token count increases. In this 9,216-token regime, optimization-based methods exhibit limited end-to-end speedup due to selection overhead dominating attention savings. By comparison, our non-iterative formulation reduces prefill latency from 1813 ms (vanilla) to 1114 ms, and total inference time for the dataset from 6101 sec to 3601 sec, demonstrating that token reduction translates into practical acceleration when pruning overhead is bounded.

Peak memory usage remains comparable across methods. Our approach consumes 18,432 MB, similar to other optimization-based baselines and lower than the unpruned full sequence (19,749 MB). This confirms that covariance-level redundancy attenuation does not introduce additional memory burden even in ultra-high-resolution settings.

Method	FLOPs (T) ↓	Peak Memory (MB) ↓	Pruning (ms) ↓	Prefill (ms) ↓	Infer Time (sec) ↓
Vanilla 100% Tokens					
Full Seq	148	19,749	-	1813	6101
Retain 30% Tokens					
DivPrune (CVPR'25)	44.4	18,512	458	1547	5387
CDPruner (NeurIPS'25)	44.4	18,736	576	1658	5787
SFPruner (Ours)	44.4	**18,432**	**28**	**1114**	**3601**

Table 4: **Hardware Profiling under Extreme Scale.** Qwen2.5-VL ($N = 9,216$).

4.6 Micro-Architectural Profiling and Code-Level Analysis

To comprehend the limited speedups of existing methods at high resolutions, we analyzed pruning overheads. As illustrated in Fig. 3, the total latency is expressed as $(T_{\text{total}} = T_{\text{ideal inference}}(K) + T_{\text{pruning}}(N))$.

We conducted benchmarks across various token counts while maintaining previous settings. By replicating the extreme scaling scenario with $(N = 9,216)$ tokens and a 30% retention budget, we isolated the execution of the pruning module. Averaging over 10 independent trials with explicit `torch.cuda.synchronize()` boundaries, the pure selection latency consistently converged to 27.7 ms. This consistency between isolated function-level profiling and macroscopic measurements (e.g., Table 4) demonstrates similar outcomes across different token counts. Methods like DivPrune and CDPruner employ sequential autoregressive loops repeated (K) times, resulting in inefficient GPU utilization. With (N=16,384), pruning overheads reach 2.6 seconds and 1.5 seconds, negating any inference speedup.

Our approach addresses this by eliminating (K)-dependent loops. As noted in Sec. 3.2, we utilize the Woodbury matrix identity when (N ¿ D) to circumvent the costly $(\mathcal{O}(N^3))$ inversion of the $(N \times N)$ similarity matrix. This reduces the problem to a $(D \times D)$ feature covariance matrix, solved using Cholesky decomposition for speed and numerical stability, reducing computational complexity to $(\mathcal{O}(ND^2))$. As shown in Fig. 3, this optimization achieves substantial latency reduction without increasing memory usage; peak memory remains comparable to existing baselines. Consequently, our method executes pruning on 16,384 tokens in just 50.0 ms, translating theoretical FLOP reductions into real-world acceleration. We rigorously validate our latency claims via detailed, function-level micro-benchmarks. Our parallel tensor formulation effectively circumvents the sequential bottlenecks encountered in previous subset-optimization techniques, such as iterative search loops. This approach enables stable execution without incurring additional kernel-launch overheads. Further experimental details can be found in the supplementary material.

4.7 Ablation Study

We conduct ablation experiments on LLaVA-NeXT-7B, retaining 320 tokens per patch, to evaluate the contributions of key components in SFPruner. Table 5 summarizes task-specific accuracy, relative performance retention (Rel.%), and selection latency.

Directional Masking vs. Sequential Selection. To assess the effect of pairwise redundancy control, we compare our ranking-based directional masking with a naive Top-K baseline and a conventional sequential selection strategy, all employing the baseline semantic guidance (SG). As shown in Table 5, the naive Top-K approach yields the lowest retention (96.9%) despite its low latency (0.7 ms) due to unresolved spatial redundancy. Sequential selection achieves the highest retention among SG-based methods (98.0%) by explicitly updating token

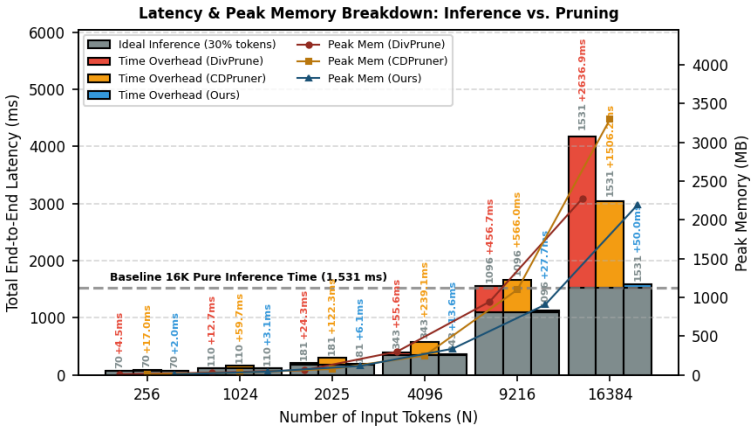

Fig. 3: **End-to-end latency and peak memory analysis across varying numbers of input tokens (N).** While existing token pruning methods (DivPrune, CDPruner) incur severe time overheads as sequence length increases, ours maintains exceptionally low overhead even at a high resolution of $N = 16384$. It closely approaches the pure inference time of the 16K baseline (1,531 ms), delivering practical computational acceleration. Bar charts indicate total latency, and line plots represent peak memory.

states step-by-step, but suffers from substantial latency (18.6 ms) due to iterative computation. In contrast, directional masking operates as a parallel process, achieving a comparable retention (97.9%) while drastically reducing latency to 0.8 ms. These results indicate that directional masking provides an efficient and effective means of mitigating local overlaps without incurring iterative bottlenecks.

Effect of Covariance-Level Attenuation. To isolate the contribution of covariance-level redundancy modeling, we replace the semantic guidance (SG) with the proposed Semantics-Guided Ridge Leverage Score (SG-RLS) while retaining the same Top-K selection strategy. As shown in Table 5, SG-RLS consistently improves relative performance from 96.9% to 97.6%, demonstrating that covariance-level attenuation alone effectively suppresses globally redundant tokens. When combined with directional masking, the relative performance further increases to 98.3% with consistent gains across tasks (e.g., GQA: 61.2 → 61.5, POPE: 86.4 → 86.7). Although covariance matrix operations introduce a modest overhead (+1.5 ms), the total latency (2.3 ms) remains substantially lower than that of sequential search (18.6 ms). These results indicate that covariance-level attenuation and directional masking are complementary, addressing global and local redundancy, respectively.

Scoring Metric	Selection Strategy	GQA	TextVQA	POPE	Rel. (%)	Time (ms) ↓
SG (S_{guide})	Top-K Selection	60.3	57.8	85.5	96.9	**0.7**
SG-RLS (Ours)	Top-K Selection	60.5	58.1	86.1	97.6	2.2
SG (S_{guide})	Sequential Search	61.2	58.2	86.3	98.0	18.6
SG (S_{guide})	Directional Masking	61.2	**58.3**	86.4	97.9	0.8
SG-RLS (Ours)	**Directional Masking**	**61.5**	**58.3**	**86.7**	**98.3**	2.3

Table 5: **Ablation on Scoring Metrics and Selection Strategies.** Evaluated on LLaVA-NeXT-7B retaining 320 tokens per image. The synergistic combination of SG-RLS and ranking-based directional masking achieves the optimal balance.

5 Conclusion

This paper revisited the computational behavior of token pruning in high-resolution Multimodal Large Language Models (MLLMs). We observed that while subset-optimization-based methods provide strong redundancy control, their iterative selection procedures introduce non-negligible overhead in dense visual regimes. Motivated by this observation, we proposed SFPruner, a non-iterative token pruning framework that performs redundancy-aware importance selection through structured tensor operations. SFPruner decomposes redundancy modeling into two complementary components. Semantics-Guided Ridge Leverage (SG-RLS) attenuates covariance-level redundancy by rebalancing dominant feature directions, while ranking-based directional masking resolves residual pairwise overlap without sequential subset construction. This formulation enables structured redundancy control while avoiding iterative dependency in the pruning process. Extensive experiments across diverse MLLM architectures, including multi-patch high-resolution (LLaVA-NeXT), single-sequence dense encoding (Qwen2.5-VL), and multi-frame video processing (LLaVA-Video), demonstrate that SFPruner consistently converts token reduction into practical latency improvement. Across evaluated settings, it maintains competitive reasoning performance while significantly reducing selection overhead. Overall, the results suggest that redundancy-aware token pruning can be realized without iterative subset construction, providing a scalable alternative for high-resolution and long-context multimodal inference.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. RS-2024-00456152). This work was also supported by LG Electronics. Computational resources were provided by "the Advanced GPU Utilization Support Program" funded by the Government of the Republic of Korea (Ministry of Science and ICT) and the Cluster Server for Computational Science at Pusan National University.

References

1. Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.
2. Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.
3. Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.
4. Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llavanext: Improved reasoning, ocr, and world knowledge, 2024.
5. Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, et al. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *arXiv preprint arXiv:2410.17247*, 2024.
6. Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pages 19–35. Springer, 2024.
7. Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. In *International Conference on Machine Learning*, 2025.
8. Changwoo Baek, Jouwon Song, Sohyeon Kim, and Kyeongbo Kong. Agilepruner: An empirical study of attention and diversity for adaptive visual token pruning in large vision-language models. In *The Fourteenth International Conference on Learning Representations*, 2026.
9. Qizhe Zhang, Aosong Cheng, Ming Lu, Renrui Zhang, Zhiyong Zhuo, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. Beyond text-visual attention: Exploiting visual cues for effective token pruning in vlms. *arXiv preprint arXiv:2412.01818*, 2024.
10. Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In *ICCV*, 2025.
11. 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 Computer Vision and Pattern Recognition Conference*, pages 19792–19802, 2025.
12. Qizhe Zhang, Mengzhen Liu, Lichen Li, Ming Lu, Yuan Zhang, Junwen Pan, Qi She, and Shanghang Zhang. Beyond attention or similarity: Maximizing conditional diversity for token pruning in mllms. *arXiv preprint arXiv:2506.10967*, 2025.
13. Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9392–9401, 2025.
14. Jouwon Song, Sohyeon Kim, and Kyeongbo Kong. Uncertainty-guided graph formulation via mwis for token pruning in lvlms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pages 9510–9519, June 2026.

15. Cameron Musco and Christopher Musco. Recursive sampling for the nystrom method. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
16. Shannon McCurdy. Ridge regression and provable deterministic ridge leverage score sampling. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.
17. 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.
18. Qizhe Zhang, Aosong Cheng, Ming Lu, Zhiyong Zhuo, Minqi Wang, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. [cls] attention is all you need for training-free visual token pruning: Make vlm inference faster. *arXiv e-prints*, pages arXiv–2412, 2024.
19. Kazi Hasan Ibn Arif, JinYi Yoon, Dimitrios S Nikolopoulos, Hans Vandierendonck, Deepu John, and Bo Ji. Hired: Attention-guided token dropping for efficient inference of high-resolution vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 1773–1781, 2025.
20. 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*, pages 24972–24982, 2025.
21. Daniel Bolya, Jianfeng Fu, Xiyang Dai, and Matt Feiszli. Token merging: Your ViT but faster. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4553–4563, 2023.
22. Michael W. Mahoney. Randomized algorithms for matrices and data. *Found. Trends Mach. Learn.*, 3(2):123–224, February 2011.
23. Chi-Heng Lin, Shangqian Gao, James Seale Smith, Abhishek Patel, Shikhar Tuli, Yilin Shen, Hongxia Jin, and Yen-Chang Hsu. ModeGPT: Modular decomposition for large language model compression. In *The Thirteenth International Conference on Learning Representations*, 2025.
24. Zhao Song, Jianfei Xue, Jiahao Zhang, and Lichen Zhang. Sublinear time quantum algorithm for attention approximation. In *The Fourteenth International Conference on Learning Representations*, 2026.
25. Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.
26. Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data, 2024.
27. 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. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017.
28. Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019.

29. 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*, pages 8317–8326, 2019.
30. 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.
31. 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. *Advances in Neural Information Processing Systems*, 35:2507–2521, 2022.
32. Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *CVPR*, 2025.
33. Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. Mlvu: A comprehensive benchmark for multi-task long video understanding. *arXiv preprint arXiv:2406.04264*, 2024.
34. Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding, 2024.
35. 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, 2025.
36. Evelyn Zhang, Fufu Yu, Aoqi Wu, Zichen Wen, Ke Yan, Shouhong Ding, Biqing Qi, and Linfeng Zhang. D^2pruner: Debiased importance and structural diversity for mllm token pruning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.