SWIFT: Spatial-Window Integrated Frequency-aware Token Pruning for Efficient MLLMs on Edge Devices

Guanglai Liu[1], Jubo Chen[2], and Xiaosheng Yu[2⋆]

[1] College of Information Science and Engineering, Northeastern University, Shenyang 110819, P. R. China
liugl2@mails.neu.edu.cn
[2] Faculty of Robot Science and Engineering, Northeastern University, Shenyang 110819, P. R. China
jbchen@stumail.neu.edu.cn
yuxiaosheng@mail.neu.edu.cn

Abstract. While Multimodal Large Language Models (MLLMs) excel in visual understanding, the quadratic complexity imposed by dense high-resolution visual tokens poses significant challenges for edge-device deployment. Current training-free pruning strategies predominantly depend on attention weights. However, viewed from the frequency domain, the self-attention mechanism functions as a low-pass filter, causing the discard of essential high-frequency signals (*e.g.*, textures and edges) and resulting in feature over-smoothing. In this paper, we present **S**patial-**W**indow **I**ntegrated **F**requency-aware **T**oken Pruning (SWIFT), a seamless plug-and-play framework designed for efficient token compression. SWIFT leverages two novel components: a Frequency-Aware Indicator (FAI), which identifies fine-grained details by estimating high-frequency residuals through matrix factorization, and a Spatial-Window Integration (SWI) module, which prevents spatial structure collapse and positional bias via localized retention. Extensive evaluations on Qwen2.5VL-3B and LLaVA benchmarks show that under a practical 50% compression ratio, SWIFT achieves a 2.10× speedup with near-lossless performance. Under an extreme 75% compression stress test, SWIFT yields an average performance drop of 5.1% across 8 benchmarks, while delivering a 2.66× speedup and 69.07% KV cache reduction. SWIFT exhibits superior robustness in detail-intensive tasks compared to SOTA methods, providing a potent lightweight solution for on-device MLLM applications. Our code is available at `https://github.com/damo-lgl/SWIFT`.

Keywords: Train-free Pruning · Efficient MLLM · Frequency-Aware Indicator

⋆ Corresponding author. Email: yuxiaosheng@mail.neu.edu.cn

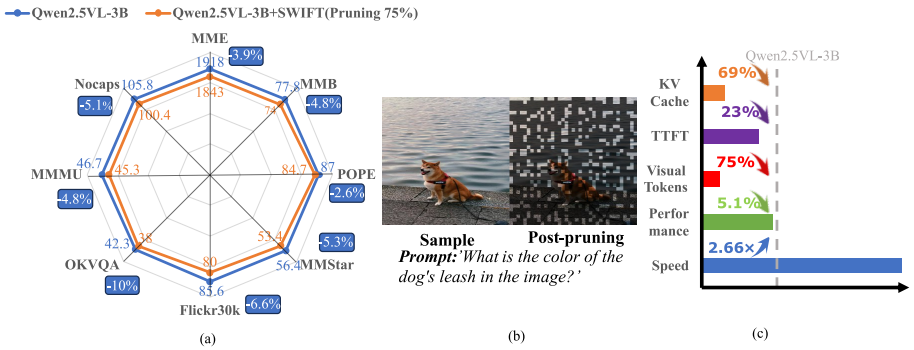

Fig. 1: (a) The radar chart illustrates the performance of SWIFT across 8 datasets when compressing the visual tokens of Qwen2.5VL-3B by 75%. The results indicate that despite a significant reduction in the number of tokens, the overall performance remains comparable to the uncompressed version. **(b)** The visualization depicts the spatial distribution of tokens during the compression process (*black tokens indicate the pruned ones*). For this task, SWIFT not only ensures the spatial uniformity of the retained tokens but also focuses on the critical regions containing the "leash". **(c)** Key performance metrics demonstrate that at a 75% compression rate, SWIFT incurs only a marginal performance drop of 5.1%, while achieving a 2.66× inference speedup, a 23% reduction in TTFT, and a 69.07% decrease in KV cache footprint.

1 Introduction

With the rapid development of Multimodal Large Language Models (MLLMs) [3, 10, 12, 18], the alignment of visual encoders with LLMs has achieved remarkable image-text reasoning capabilities. However, this success comes at immense computational costs. High-resolution images are encoded into thousands of visual tokens, leading to a quadratic ($\mathcal{O}(N^2)$) increase in self-attention complexity and substantial memory consumption. While acceptable for cloud clusters, this overhead poses a critical bottleneck for deploying high-performance MLLMs on resource-constrained edge devices. For instance, despite being optimized for edge computing, models like Qwen2.5VL-3B still struggle with high Latency and Time to First Token (TTFT) when processing long visual sequences due to severe memory bandwidth limitations.

To address the computational redundancy of visual tokens, existing compression strategies are predominantly categorized into training-based and training-free paradigms. Training-based approaches [7, 10, 16] introduce parameterized modules that entail exorbitant retraining costs, failing to meet the lightweight, plug-and-play requirements of edge devices. Conversely, training-free methods [9, 34] primarily rely on the heuristic assumption that "attention weight equals importance". However, this perspective ignores the "attention sinks" induced by Softmax normalization and the low-pass filtering characteristics of self-attention, erroneously discarding high-frequency tokens that carry crucial details. Further-

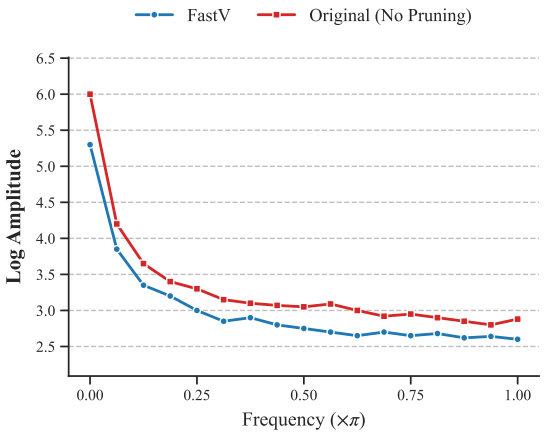

Fig. 2: Spectral energy distribution at the 31st layer. The original model (red) exhibits a pronounced low-frequency bias due to self-attention. Applying FastV (blue) further exacerbates this low-pass filtering effect, severely accelerating the loss of high-frequency information and resulting in over-smoothed features devoid of fine-grained details.

more, unstructured global pruning inherently disrupts the spatial continuity of images, severely degrading performance on fine-grained perception tasks.

In this study, we discard the conventional attention-score-based stereotype and re-examine MLLMs from a signal processing frequency-domain perspective. Through spectral analysis of the attention matrix, we observe that the self-attention mechanism acts as a severe low-pass filter in deep networks. The attention matrix tends to homogenize, suppressing high-frequency alternating current (AC) components (*e.g.*, textures and edges) while dominating with redundant low-frequency direct current (DC) components. This "feature over-smoothing" phenomenon fundamentally explains the failure of existing pruning methods: they tend to retain low-frequency tokens trapped in "attention sinks", accelerating the loss of fine-grained information. Thus, introducing an explicit frequency-aware mechanism is critical for preserving fine-grained perception capabilities.

Motivated by these insights, we propose Spatial-Window Integrated Frequency-aware Token Pruning (SWIFT), a training-free, plug-and-play visual token compression framework. SWIFT consists of two core modules: the Frequency-Aware Indicator (FAI) and the Spatial-Window Integrated (SWI) module. FAI utilizes attention matrix decomposition to strip away uniform low-frequency bases, precisely locating high-frequency tokens to preserve principal edges and textures. Concurrently, SWI introduces spatial regularization by restoring the sequence to a 2D grid and enforcing local window retention, thereby overcoming positional bias and maintaining spatial structure. Extensive evaluations on 8 benchmark datasets using LLaVA-1.5 (7B/13B) and Qwen2.5VL-3B demonstrate SWIFT's superiority. For practical edge deployment, the 50% compression setting delivers near-lossless accuracy with a 2.10× inference speedup. As shown in Figure

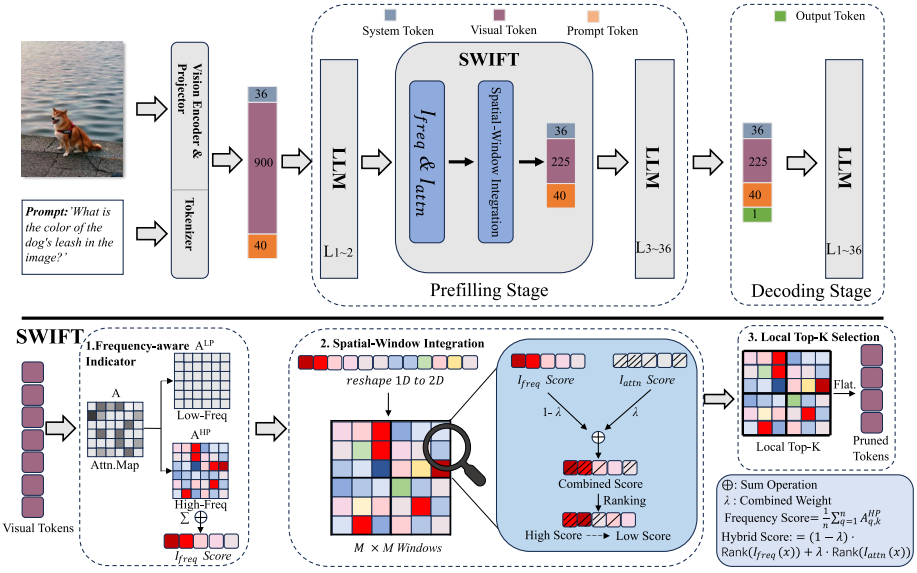

Fig. 3: Overview of the SWIFT framework. Integrated as a plug-and-play module in Qwen2.5VL-3B, SWIFT compresses visual tokens during prefilling. The process involves three key steps: (1) calculating high-frequency contributions via matrix factorization (FAI); (2) remapping 1D tokens to 2D spatial windows to compute hybrid scores fusing frequency and attention cues (SWI); and (3) performing independent local Top-K selection within each window to preserve critical spatial structures before outputting the pruned sequence.

1, under the extreme 75% compression stress test, SWIFT retains only 25% of visual tokens with an average 5.1% performance drop across 8 datasets, while reducing the KV cache footprint by 69.07% and accelerating inference by 2.66×, significantly alleviating edge device memory bottlenecks.

The main contributions of this paper are summarized as follows:

- We reveal the low-pass filtering characteristics of self-attention in deep networks and identify the resulting "feature over-smoothing" problem. We theoretically expose the flaws of the conventional "attention weight equals importance" assumption, providing a novel frequency-domain perspective to mitigate fine-grained information loss under high compression ratios.
- We propose SWIFT, a training-free and plug-and-play visual token compression framework. Its core modules, FAI and SWI, synergistically combine frequency-domain saliency and spatial regularization to effectively preserve critical textures and overcome spatial structure collapse during pruning.
- Extensive experiments demonstrate that SWIFT achieves near-lossless performance under the practical 50% compression setting, delivering a 2.10× inference speedup and 46.05% KV cache reduction. It significantly outper-

forms SOTA methods on detail-sensitive visual tasks, highlighting its immense potential for edge computing deployment.

2 Related Work

As Multimodal Large Language Models (MLLMs) evolve towards high-resolution perception, the surge in the number of visual tokens has led to substantial computational overhead and memory bottlenecks [5, 9, 30]. To alleviate this issue, existing research on visual token compression can be broadly categorized into training-based methods [2, 6, 7, 13, 16, 23] and training-free methods [9, 14, 19, 22, 24, 25, 28, 29, 34].

2.1 Training-Based Visual Token Compression Methods

Training-based methods aim to reduce the sequence length of visual features by introducing parameterized modules during the alignment stage. BLIP-2 [16] and Flamingo [2] proposed Q-Former and Perceiver Resampler, respectively. These methods utilize a set of learnable query vectors to extract semantic summaries from visual features via cross-attention mechanisms, thereby achieving fixed-length outputs. MiniGPT-v2 [7] adopted a more direct spatial pooling strategy, reducing the resolution by concatenating adjacent tokens. Although these methods effectively reduce the number of tokens, they are often accompanied by prohibitive pre-training costs. Furthermore, once the compression strategies are trained, it is difficult to adjust them dynamically according to different task requirements, resulting in a lack of flexibility.

2.2 Training-Free Visual Token Compression Methods

To circumvent the costs of retraining, training-free methods eliminate redundancy by directly manipulating the attention mechanism during the inference stage. FastV [9], based on the attention sparsity hypothesis, directly prunes tokens with lower attention scores in deep networks. SparseVLM [34] introduced a text-guided mechanism, utilizing text tokens as scorers to filter key visual information. Subsequent works, such as MustDrop [19] and VisionZip [29], further explored similarity-based token merging and clustering strategies to preserve more contextual information. While ToMe [4] reduces tokens via similarity merging, it exacerbates the low-pass filtering effect by averaging features, failing to protect high-frequency signals. SWIFT explicitly isolates these details via FAI. Furthermore, although VScan [33] employs spatial scans, its core metric relies on attention scores, which are biased towards low-frequency direct current components. However, the aforementioned methods primarily rely on attention weights as the basis for importance evaluation, making them susceptible to the "Attention Sink" phenomenon. They tend to retain low-frequency background information while neglecting high-frequency details that carry textures and edges. Moreover, global pruning easily disrupts the spatial structural integrity of images.

Table 1: Performance evaluation of the SWIFT method on the LLaVA-1.5-7B model (based on three visual token compression ratios: 50%, 75%, and 90%, covering 8 datasets). The baseline model configuration uses 576 visual tokens. In this study, SWIFT is compared with published methods (such as FastV and SparseVLM). The best performance is highlighted in red, and the second-best in orange. Experimental results demonstrate that SWIFT achieves the best or near-best performance across the vast majority of benchmark datasets, and its performance advantage becomes even more significant under extreme compression conditions.

Method	MMB	MMStar	MME	OKVQA	POPE	MMMU	Nocaps	Flickr30k
LLaVA-7B	Upper Bound, 576 Tokens (100%)							
Vanilla	64.1	33.8	1610	53.4	86.4	36.3	105.6	84.4
	Retain 288 Tokens (↓ 50%)							
Random	61.7	34.0	1596	51.6	84.4	34.7	103.9	82.9
SparseVLM [34]	62.8	34.2	1543	52.6	82.2	36	104.6	84.8
FastV [9]	64.2	33.3	1599	53.0	83.7	35.3	104.4	84.3
SWIFT (Ours)	64.5	34.0	1607	52.8	85.2	35.5	105.4	84.5
	Retain 144 Tokens (↓ 75%)							
Random	58.7	32.6	1548	48.2	80.2	35.1	101.3	76.3
SparseVLM [34]	59.7	33.1	1543	51.3	80.5	35.4	96.3	75.5
FastV [9]	57.9	32.1	1588	50.4	75.3	35.2	99.2	79.1
SWIFT (Ours)	62.6	33	1607	51.9	81.4	34.7	102.7	83.3
	Retain 58 Tokens (↓ 90%)							
Random	53	31.1	1434	41.3	64	34.2	80.7	63
SparseVLM [34]	56.2	31.2	1403	39.5	71.4	35.8	50.9	36.6
FastV [9]	50.6	30.6	1448	44.0	66.3	35.1	74.7	55.1
SWIFT (Ours)	57.4	31.1	1540	45.0	75.3	35.4	89.1	65

To address these limitations, this paper proposes SWIFT. Unlike traditional attention scoring, SWIFT innovatively takes a frequency-domain perspective, utilizing matrix factorization to extract high-frequency residuals to lock onto crucial details. Combined with Spatial-Window Integration, SWIFT achieves efficient compression while effectively overcoming positional bias and preserving fine-grained perception capabilities.

3 Methodology

3.1 Motivation: High-Frequency Information Loss

Multimodal Large Language Models (MLLMs) process a concatenated sequence of system, visual, and user tokens, denoted as $X = [T_s, T_v, T_p]$, through stacked Transformer layers. Within each layer, the self-attention matrix $A = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)$ serves as the core computational unit.

However, from a signal processing perspective, the self-attention mechanism exhibits pronounced low-pass filtering characteristics. As network layers deepen, the attention matrix A gradually converges towards a normalized all-ones matrix ($A \approx \frac{1}{n}\mathbf{1}\mathbf{1}^T$). This operation effectively computes the mean of all tokens, preserving the low-frequency direct current (DC) components (*e.g.*, global background) while heavily suppressing high-frequency components (*e.g.*, object edges and local texture details).

Furthermore, traditional token pruning methods often exacerbate this high-frequency information loss. Recent studies [26, 27] indicate that MLLMs tend to allocate massive attention weights to low-information "attention sinks" rather than semantically rich tokens. Consequently, blindly pruning tokens based on attention weights discards crucial high-frequency details. As shown in Fig. 2, applying attention-based pruning (*e.g.*, FastV [9]) further accelerates the loss of high-frequency components, resulting in over-smoothed feature representations devoid of fine-grained information.

This insight naturally motivates our proposed SWIFT method. To maintain model performance under the limited computational capacities of resource-constrained edge devices, it is imperative to move beyond simple attention weights and explicitly identify tokens that contribute to high-frequency components, thereby counteracting the low-pass filtering effect.

3.2 Overall Pipeline of SWIFT

During the prefilling stage of Multimodal Large Language Models (MLLMs), SWIFT is inserted between the second and third layers of the Large Language Model (LLM). It discards the single importance scoring criterion and instead adopts a "frequency-spatial" dual-stream selection mechanism to transform the original visual token sequence into a more compact representation while minimizing information loss. Figure 3 illustrates the overall pipeline of SWIFT.

3.3 Frequency-Aware Indicator

To identify high-frequency tokens with low computational cost during the inference stage, we circumvent the relatively expensive Fast Fourier Transform (FFT) and employ an approximation method based on attention matrix factorization [15].

We decompose the attention matrix A into a low-frequency component A^{LP} and a high-frequency component A^{HP}:

$$A^{LP} = \frac{1}{n}\mathbf{1}\mathbf{1}^T, \tag{1}$$

$$A^{HP} = A - A^{LP}. \tag{2}$$

Here, A^{LP} represents a completely uniform attention distribution (a pure low-pass filter), whereas A^{HP} is the residual between the original attention matrix

and the uniform matrix. This residual matrix A^{HP} precisely captures the specific interaction information among tokens that deviates from the global average.

For the k-th visual token, its global contribution to the high-frequency components of the feature layer, denoted as $I_{freq}(k)$, is defined as the mean of the k-th column of the high-frequency attention matrix A^{HP} (*i.e.*, the average high-frequency attention received by this token):

$$I_{freq}(k) = \frac{1}{n} \sum_{q=1}^{n} A^{HP}_{q,k}, \tag{3}$$

where n denotes the sequence length, and $A^{HP}_{q,k}$ represents the high-frequency attention residual weight from the q-th query token to the k-th key token.

By definition, $A^{HP} = A - A^{LP}$, where A^{LP} represents the uniformly distributed low-frequency basis. Therefore, $I_{freq}(k)$ essentially measures the extent to which the k-th visual token deviates from a uniform attention distribution.

- **Significant Positive Value ($I_{freq} > 0$):** If a token scores a high positive value, it indicates that it is "over-attended" during attention interactions, demonstrating strong high-frequency saliency. This typically corresponds to semantic subjects, edges, or key textures in the image, classifying it as a high-frequency (HF) token.
- **Redundant Negative Value ($I_{freq} < 0$):** Conversely, if the score is negative, the attention received by the token is below average. This generally corresponds to large areas of smooth background or semantically sparse regions, classifying it as a low-frequency (LF) redundant token.

We adopt a Top-K retention strategy. All visual tokens are sorted in descending order based on $I_{freq}(k)$, and the top r tokens with the highest scores are prioritized for retention (forming the set $\mathcal{N}_{HF}$):

$$\mathcal{N}_{HF} = \arg \max_{S \subset \{1 \dots n\}, |S| = r} \sum_{k \in S} I_{freq}(k). \tag{4}$$

3.4 Spatial-Window Integrated Selection

Although the frequency-domain indicator effectively mitigates the loss of high-frequency information, pure score-based filtering still faces challenges under high compression scenarios. Existing studies [26] point out that attention-based pruning methods exhibit significant positional bias, leading to an uneven distribution of retained tokens. This is detrimental to tasks relying on spatial distribution information: global pruning can cause spatially independent regions with low scores to be mistakenly discarded, forming "spatial blind spots". To address this issue, we design a Spatial-Window Integrated (SWI) selection mechanism based on the principle of spatial uniformity [26]. Through a mandatory local retention strategy, this mechanism ensures spatial coverage of image features, thereby overcoming the risk of spatial collapse induced by global sorting.

Table 2: Performance comparison of various token compression methods on the LLaVA-1.5-13B model. SWIFT demonstrates excellent performance across 8 datasets, and its performance advantage becomes increasingly significant under extreme conditions with higher compression ratios.

Method	MMB	MMStar	MME	OKVQA	POPE	MMMU	Nocaps	Flickr30k
LLaVA-13B	Upper Bound, 576 Tokens (100%)							
Vanilla	68.7	36.2	1716	58.3	86.6	35.1	109.4	85.3
	Retain 288 Tokens (↓ 50%)							
FastV [9]	68.3	35.0	1708	57.9	82.6	35.4	108.4	83.6
SWIFT (Ours)	68.6	35.9	1731	57.6	85.8	35.4	109.6	84.5
	Retain 144 Tokens (↓ 75%)							
FastV [9]	67.0	33.4	1649	55.9	78.3	34.9	103.6	80.8
SWIFT (Ours)	67.8	35.4	1680	56.0	83.2	35.7	107.0	83.4
	Retain 58 Tokens (↓ 90%)							
FastV [9]	62.1	32.8	1562	49.7	70.5	34.0	90.8	60.8
SWIFT (Ours)	64.5	33.7	1582	51.3	78.2	35.6	97.0	70.3

3.5 Hybrid Scoring Formula

To balance semantic importance (reflected by attention scores) and feature diversity (reflected by frequency scores), we follow the approach in FastV for calculating the attention scores of visual tokens, utilizing the attention matrix from the second layer of the LLM. We define the hybrid score as:

$$\text{Score}(x) = (1 - \lambda) \cdot \text{Rank}(I_{freq}(x)) + \lambda \cdot \text{Rank}(I_{attn}(x)), \tag{5}$$

where λ is the balancing coefficient. Through hyperparameter tuning experiments on the MME benchmark (for detailed experimental data, please refer to Sec. 4.4), we observe that the model achieves the optimal trade-off between performance and compression ratio when $\lambda \approx 0.5$. This confirms that semantic information and frequency-domain information are not mutually exclusive in visual representations, but rather highly complementary: S_{attn} locks onto "where to look" (object localization), while S_{freq} ensures "what to see clearly" (detail reconstruction).

4 Experimental Setup

4.1 Evaluation Tasks

To evaluate the effectiveness of our method on image understanding tasks, we conduct experiments on eight widely used benchmark datasets, including MME [11], MMBench (MMB) [20], POPE [17], MMStar [8], Flickr30k [31], OK-VQA [21], MMMU [32], and Nocaps [1]. Meanwhile, we compare our method with

Table 3: Performance comparison of various token compression methods on the Qwen2.5VL-3B model. SWIFT demonstrates excellent performance across 8 datasets, and its performance advantage becomes increasingly significant under extreme conditions with higher compression ratios. Notably, this method performs particularly well on tasks such as comprehensive reasoning and image captioning, proving that the SWIFT method plays a more critical role when applied to small-parameter models suitable for edge devices.

Method	MMB	MMStar	MME	OKVQA	POPE	MMMU	Nocaps	Flickr30k
Qwen2.5VL-3B	Reduct Ratio: 0%							
Vanilla	77.8	56.4	1918	42.3	87.0	47.6	105.8	85.7
	Reduct Ratio: 50%							
FastV [9]	76.3	54.4	1848	40.3	87.3	46.5	103.8	83.4
SWIFT (Ours)	77.3	55.4	1913	41.8	86.8	46.9	104.7	84.3
	Reduct Ratio: 66%							
FastV [9]	74.3	50.6	1868	38.2	86.2	45.9	100.0	81.4
SWIFT (Ours)	75.8	54.9	1885	39.1	85.4	46.2	101.8	82.2
	Reduct Ratio: 75%							
FastV [9]	71.4	48.6	1809	36.1	84.3	46.3	97.0	77.0
SWIFT (Ours)	74.0	53.4	1843	38.0	84.7	45.3	100.4	80.0

existing state-of-the-art (SOTA) approaches (*e.g.*, FastV [9], SparseVLM [34]), which primarily utilize attention scores to compress redundant visual tokens in Multimodal Large Language Models (MLLMs).

4.2 Implementation Details

To further verify the generalization of our method, we test SWIFT on various open-source MLLMs with different architectures. We primarily conduct experiments on LLaVA-1.5-7B and the Qwen2.5VL-3B model, which is suitable for edge device deployment. Experiments on LLaVA-1.5-13B further validate the effectiveness of our method on larger-scale models. All experiments are conducted on a single NVIDIA GeForce RTX 4090 PLUS GPU with 48GB of memory.

4.3 Main Results

Results on LLaVA-1.5-7B. As shown in Tab. 1, SWIFT consistently outperforms existing SOTA methods (FastV [9], SparseVLM [34]) across eight benchmarks. Notably, SWIFT's advantage widens significantly as the compression rate increases, demonstrating exceptional robustness under extreme compression. On fine-grained tasks highly sensitive to high-frequency information (*e.g.*, Nocaps, Flickr30k), SWIFT achieves massive improvements (*e.g.*, +19.3% over FastV at 90% compression on Nocaps), strongly validating the Frequency-Aware Indicator (FAI) in preserving crucial texture details. Furthermore, SWIFT effectively

Table 4: Ablation study results based on the Qwen2.5VL-3B model under different compression configurations. We evaluate the performance changes on the MME, POPE, Nocaps, and Flickr30k datasets after removing the Spatial-Window Integration (SWI), Frequency-Aware Indicator (FAI), and semantic Attention Guidance (AG) components, respectively. In comparison, the SWIFT method achieves the lowest performance degradation across the four datasets and demonstrates stronger stability during the token compression process.

Method	SWI	FAI	AG	MME	POPE	Nocaps	Flickr30k
Qwen2.5VL-3B	-	-	-	1918	87	105.8	85.7
				Reduct Ratio: 50%			
w/o SWI	✗	✓	✓	1836	83.3	103.7	83.2
w/o FAI	✓	✗	✓	1890	83.8	105.1	83.5
w/o AG	✓	✓	✗	1900	84.2	104.5	83.6
SWIFT (Ours)	✓	✓	✓	1913	86.8	104.7	84.3
				Reduct Ratio: 66%			
w/o SWI	✗	✓	✓	1787	74.9	94.7	77.0
w/o FAI	✓	✗	✓	1849	79	100.5	80.5
w/o AG	✓	✓	✗	1841	81.0	100.6	81.4
SWIFT (Ours)	✓	✓	✓	1885	85.4	101.8	82.2
				Reduct Ratio: 75%			
w/o SWI	✗	✓	✓	1708	70.9	68.6	63.3
w/o FAI	✓	✗	✓	1795	77.6	94.8	74.1
w/o AG	✓	✓	✗	1817	73.7	98.1	73.9
SWIFT (Ours)	✓	✓	✓	1843	84.7	100.4	80.0

prevents performance collapse in the POPE hallucination evaluation (+9 vs. FastV), confirming that the Spatial-Window Integrated (SWI) mechanism successfully maintains spatial structural integrity and mitigates hallucination issues caused by local receptive field loss.

Results on LLaVA-1.5-13B. To validate cross-scale generalization, we evaluate LLaVA-1.5-13B (Tab. 2). Under a mild 50% compression, SWIFT tightly matches the Vanilla model on MMB (68.6 vs. 68.7) and even surpasses the full model on MME (1731 vs. 1716), showcasing extraordinary information fidelity. At an extreme 90% compression, where FastV suffers drastic degradation, SWIFT remains highly robust. Specifically, it outperforms FastV by 9.5 and 6.2 points on Flickr30k and Nocaps, respectively, proving that FAI effectively counteracts the feature over-smoothing trend in deeper networks. Additionally, SWIFT maintains a 7.7-point lead on POPE, further verifying SWI's stability in suppressing hallucinations across larger models.

Results on Qwen2.5VL-3B. To assess viability for edge devices, we evaluate the lightweight Qwen2.5VL-3B (Tab. 3). SWIFT demonstrates superior token utilization: under 75% compression (retaining 25% tokens), it surpasses or matches FastV under 66% compression (retaining 34% tokens) on MMStar (53.4 vs. 50.6) and OK-VQA (38.0 vs. 38.2). This highly optimized token efficiency

greatly unleashes edge computational power. Despite small models being inherently sensitive to feature loss, SWIFT scores 74.0 on MMB (+2.6 over FastV), proving that our hybrid sorting strategy preserves critical reasoning semantics. Moreover, leading FastV by 3.4 and 3.0 points on Nocaps and Flickr30k respectively, SWIFT confirms that explicitly retaining high-frequency tokens is vital for mitigating fine-grained detail loss in capacity-constrained models, making it an optimal visual acceleration solution for edge deployment.

4.4 Ablation Study

To quantify the contribution of each component, we progressively ablate Spatial-Window Integration (SWI), the Frequency-aware Indicator (FAI), and Attention-based Guidance (AG) on Qwen2.5VL-3B (Tab. 4).

Effect of Spatial-Window Integration (SWI). SWI is critical for maintaining the model's spatial perception. Removing SWI under 90% compression causes catastrophic performance collapse (*e.g.*, POPE drops drastically from 75.27 to 60.94, Nocaps from 89.13 to 68.6). Without spatial constraints, global sorting entirely discards low-confidence regions, creating visual blind spots that trigger severe object hallucinations. SWI guarantees spatial uniformity, effectively preventing the loss of local semantics.

Effect of Frequency-aware Indicator (FAI). FAI explicitly counteracts deep network feature smoothing by acting as a "detail anchor". When FAI is removed, performance on texture-sensitive tasks degrades notably (Nocaps drops from 89.13 to 84.75; MME from 1456 to 1425). Relying solely on AG in a highly sparse token space leads to the loss of high-frequency edges and text features, verifying FAI's necessity for preserving fine-grained semantics.

Effect of Attention-based Guidance (AG). AG provides the essential semantic orientation. Without AG, despite SWI and FAI preventing spatial collapse and retaining textures, overall performance remains sub-optimal compared to full SWIFT (*e.g.*, POPE reaches only 70.72). Relying exclusively on physical characteristics (position and frequency) creates a misalignment with the LLM's semantic intent.

Necessity of the Weighting Coefficient. We sweep the balancing coefficient $\lambda \in [0, 1]$ on MME across multiple model architectures and input resolutions (Fig. 4). Performance degrades when λ approaches either extreme, and consistently peaks at $\lambda \approx 0.5$ across all settings. This validates the robust 1:1 complementarity between semantic attention and frequency saliency: I_{attn} locates semantic subjects, while I_{freq} preserves fine-grained textures, jointly maximizing information integrity.

4.5 Effectiveness Analysis

To validate SWIFT's capability in counteracting feature over-smoothing (Sec. 3), we visualize the spectral energy distribution of 31st-layer features under 50% compression (Fig. 5).

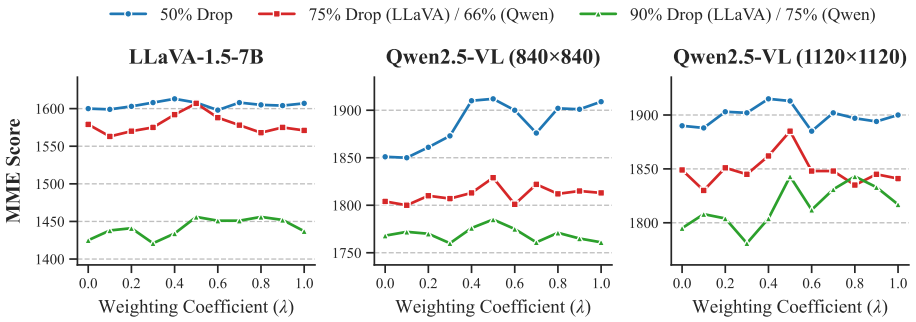

Fig. 4: Impact of weighting coefficient λ on MME performance across different architectures and resolutions. Performance consistently peaks at $\lambda \approx 0.5$ under all compression ratios, verifying the universal complementarity of the hybrid scoring strategy.

As shown, FastV's spectral energy decays sharply in the high-frequency range. By relying solely on attention weights, it predominantly retains low-frequency components, which accelerates the loss of high-frequency textures and exacerbates deep network over-smoothing. Conversely, SWIFT maintains significantly higher energy in mid-to-high frequency bands. The Frequency-Aware Indicator (FAI) successfully corrects the inherent low-pass bias of self-attention, explicitly preserving tokens crucial for high-frequency reconstruction. This optimized spectral response, synergized with spatial-window selection, maximizes the retention of fine-grained semantics (*e.g.*, edges) under high compression. It physically explains SWIFT's superior robustness on detail-sensitive tasks (VQA, Image Captioning), fully validating the theoretical efficacy of our "frequency-spatial" dual mechanism.

4.6 Efficiency

To evaluate SWIFT's deployment potential on resource-constrained edge devices, we conduct comprehensive profiling on an NVIDIA RTX 2080 (10GB) with 840×840 input images, covering both end-to-end performance and module-level overhead (Tab. 5).

Decoding Latency. Compressing visual tokens from 900 to 225 (75% reduction) reduces single-step decoding latency from 7.74 ms to 2.91 ms (62.4% reduction), yielding a 2.66× inference speedup and significantly improving real-time interaction fluency on edge devices.

TTFT and Module Overhead. By shortening the input sequence, SWIFT reduces TTFT from 2.61 s to 2.01 s (23.0% improvement) at 75% compression. We further break down the prefill-stage overhead: the lightweight FAI module consumes only ∼0.5 ms, and the total module overhead remains below 1.8% of the overall TTFT across all compression ratios, with only ∼5 MB active memory and zero additional peak VRAM, verifying its negligible computational cost.

Table 5: Performance and overhead profiling of SWIFT on RTX 2080 with 840×840 input images. The overhead ratio is calculated relative to the total TTFT.

Metric	**Vanilla**	**50%↓**	**66%↓**	**75%↓**
End-to-End Performance				
Visual Token	900	450	306	225
TTFT (s)	2.61	2.22 (↓14.9%)	2.07 (↓20.7%)	2.01 (↓23.0%)
KV Cache	100%	46.05%	64.05%	69.07%
Decoding Latency (ms)	7.74 (1.00×)	3.69 (↓52.3%) (2.10×)	3.19 (↓58.8%) (2.42×)	2.91 (↓62.4%) (2.66×)
Prefill-Stage Module Overhead				
FAI Latency (ms)	–	0.51	0.52	0.45
SWI Latency (ms)	–	34.26	35.14	33.83
Module Total (ms)	–	34.77	35.66	34.28
Overhead Ratio	–	1.57%	1.73%	1.72%
Active Memory (MB)	–	5.46	5.07	5.50
Peak VRAM Overhead	–	+0 MB	+0 MB	+0 MB

KV Cache Memory Footprint. SWIFT reduces the KV Cache footprint by up to 69.07%, which directly mitigates the memory bottleneck for long-context edge deployment and enables longer multi-turn dialogues within limited memory budgets.

These results confirm that SWIFT is a lightweight, low-latency inference solution that well satisfies the stringent requirements of edge computing scenarios.

5 Conclusion

To address the deployment bottlenecks of MLLMs on edge devices, we propose SWIFT, a training-free visual token compression framework. Revealing that conventional attention-based pruning inherently suffers from low-pass filtering and discards crucial high-frequency details, SWIFT synergizes a Frequency-Aware Indicator (FAI) and Spatial-Window Integration (SWI) to strip redundant low-frequency backgrounds while preserving critical textures and spatial integrity. Evaluations on Qwen2.5VL-3B demonstrate that under the practical 50% compression ratio, SWIFT delivers a 2.10× speedup with near-lossless performance. Under the extreme 75% compression stress test, retaining only 25% of tokens yields a 2.66× inference speedup and a 69.07% KV cache reduction with an average 5.1% performance drop across 8 benchmarks. By breaking the "attention equals importance" stereotype, our "frequency-spatial" dual mechanism ensures robust fine-grained perception, offering a novel theoretical perspective and a highly practical solution for efficient edge-based multimodal computing.

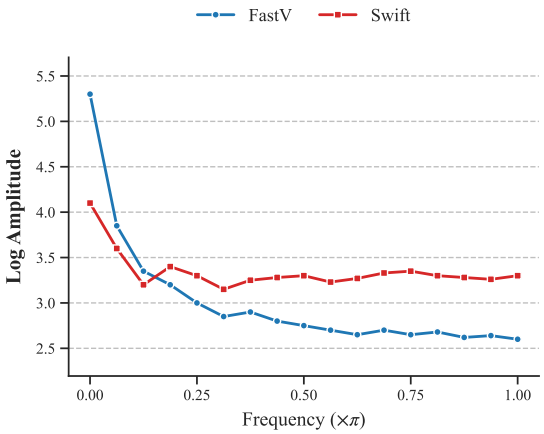

Fig. 5: Spectral comparison of FastV (*blue*) and SWIFT (*red*) at the 31st layer (50% pruning rate). FastV exhibits a severe low-pass filtering effect, leading to significant high-frequency attenuation. Conversely, SWIFT maintains a much smoother spectral response, particularly outperforming FastV in mid-to-high frequency bands ($> 0.25\pi$). This demonstrates SWIFT's effectiveness in preserving fine-grained details and mitigating feature over-smoothing.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 52571332.

References

1. Agrawal, H., Desai, K., Wang, Y., Chen, X., Jain, R., Johnson, M., Batra, D., Parikh, D., Lee, S., Anderson, P.: nocaps: novel object captioning at scale. arXiv preprint arXiv:1812.08658 (2019)
2. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: A visual language model for few-shot learning. arXiv preprint arXiv:2204.14198 (2022)
3. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., Zhong, H., Zhu, Y., Yang, M., Li, Z., Wan, J., Wang, P., Ding, W., Fu, Z., Xu, Y., Ye, J., Zhang, X., Xie, T., Cheng, Z., Zhang, H., Yang, Z., Xu, H., Lin, J.: Qwen2.5-vl technical report. arXiv preprint arXiv:2502.13923 (2025)
4. Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token merging: Your vit but faster. arXiv preprint arXiv:2210.09461 (2022)
5. Cao, J., Ye, P., Li, S., Yu, C., Tang, Y., Lu, J., Chen, T.: Madtp: Multimodal alignment-guided dynamic token pruning for accelerating vision-language transformer. In: CVPR. pp. 15710–15719 (2024)
6. Chen, J., Ye, L., He, J., Wang, Z., Khashabi, D., Yuille, A.L.: Efficient large multimodal models via visual context compression. In: NeurIPS. vol. 37, pp. 73986–74007 (2024)

7. Chen, J., Zhu, D., Shen, X., Li, X., Liu, Z., Zhang, P., Krishnamoorthi, R., Chandra, V., Xiong, Y., Elhoseiny, M.: Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. arXiv preprint arXiv:2310.09478 (2023)
8. Chen, L., Li, J., Dong, X., Zhang, P., Zang, Y., Chen, Z., Duan, H., Wang, J., Qiao, Y., Lin, D., et al.: Are we on the right way for evaluating large vision-language models? arXiv preprint arXiv:2403.20330 (2024)
9. Chen, L., Zhao, H., Liu, T., Bai, S., Lin, J., Zhou, C., Chang, B.: An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. arXiv preprint arXiv:2403.06764 (2024)
10. Chen, Z., Wu, J., Wang, W., Su, W., Chen, G., Xing, S., Zhong, M., Zhang, Q., Zhu, X., Lu, L., et al.: Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In: CVPR (2024)
11. Fu, C., Chen, P., Shen, Y., Qin, Y., Zhang, M., Lin, X., Yang, J., Zheng, X., Li, K., Sun, X., et al.: Mme: A comprehensive evaluation benchmark for multimodal large language models. arXiv preprint arXiv:2306.13394 (2023)
12. Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al.: Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948 (2025)
13. Huang, W., Zhai, Z., Shen, Y., Cao, S., Zhao, F., Xu, X., Ye, Z., Lin, S.: Dynamic-llava: Efficient multimodal large language models via dynamic vision-language context sparsification. arXiv preprint arXiv:2412.00876 (2024)
14. Jiang, Y., Wu, Q., Lin, W., Yu, W., Zhou, Y.: What kind of visual tokens do we need? training-free visual token pruning for multi-modal large language models from the perspective of graph. arXiv preprint arXiv:2501.02268 (2025)
15. Lee, D.J., Hur, J., Choi, J., Yu, J., Kim, J.: Frequency-aware token reduction for efficient vision transformer. arXiv preprint arXiv:2511.21477 (2025)
16. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: ICML. pp. 19730–19742 (2023)
17. Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, W.X., Wen, J.R.: Evaluating object hallucination in large vision-language models. arXiv preprint arXiv:2305.10355 (2023)
18. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. In: NeurIPS. vol. 36, pp. 34892–34916 (2023)
19. Liu, T., Shi, L., Hong, R., Hu, Y., Yin, Q., Zhang, L.: Multi-stage vision token dropping: Towards efficient multimodal large language model. arXiv preprint arXiv:2411.10803 (2024)
20. Liu, Y., Duan, H., Zhang, Y., Li, B., Zhang, S., Zhao, W., Yuan, Y., Wang, J., He, C., Liu, Z., et al.: Mmbench: Is your multi-modal model an all-around player? arXiv preprint arXiv:2307.06281 (2023)
21. Marino, K., Rastegari, M., Farhadi, A., Mottaghi, R.: Ok-vqa: A visual question answering benchmark requiring external knowledge. arXiv preprint arXiv:1906.00067 (2019)
22. Shang, Y., Cai, M., Xu, B., Lee, Y.J., Yan, Y.: Llava-prumerge: Adaptive token reduction for efficient large multimodal models. arXiv preprint arXiv:2403.15388 (2024)
23. Tong, B., Lai, B., Zhou, Y., Luo, G., Shen, Y., Li, K., Sun, X., Ji, R.: Flashsloth: Lightning multimodal large language models via embedded visual compression. arXiv preprint arXiv:2412.04317 (2024)
24. Wang, A., Sun, F., Chen, H., Lin, Z., Han, J., Ding, G.: [CLS] token tells everything needed for training-free efficient mllms. arXiv preprint arXiv:2412.05819 (2024)

25. Wang, H., Yu, Z., Spadaro, G., Ju, C., Quétu, V., Tartaglione, E.: Folder: Accelerating multi-modal large language models with enhanced performance. arXiv preprint arXiv:2501.02430 (2025)
26. Wen, Z., Gao, Y., Li, W., He, C., Zhang, L.: Token pruning in multimodal large language models: Are we solving the right problem? arXiv preprint arXiv:2502.11501 (2025)
27. Xiao, G., Tian, Y., Chen, B., Han, S., Lewis, M.: Efficient streaming language models with attention sinks. arXiv preprint arXiv:2309.17453 (2023)
28. Xing, L., Huang, Q., Dong, X., Lu, J., Zhang, P., Zang, Y., Cao, Y., He, C., Wang, J., Wu, F., et al.: Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. arXiv preprint arXiv:2410.17247 (2024)
29. Yang, S., Chen, Y., Tian, Z., Wang, C., Li, J., Yu, B., Jia, J.: Visionzip: Longer is better but not necessary in vision language models. arXiv preprint arXiv:2412.04467 (2024)
30. Ye, H., Yu, C., Ye, P., Xia, R., Tang, Y., Lu, J., Chen, T., Zhang, B.: Once for both: Single stage of importance and sparsity search for vision transformer compression. In: CVPR. pp. 5578–5588 (2024)
31. Young, P., Lai, A., Hodosh, M., Hockenmaier, J.: From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. Transactions of the Association for Computational Linguistics **2**, 67–78 (2014)
32. Yue, X., Ni, Y., Zhang, K., Zheng, T., Liu, R., Zhang, G., Stevens, S., Jiang, D., Ren, W., Sun, Y., et al.: Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. arXiv preprint arXiv:2311.16502 (2023)
33. Zhang, C., Ma, K., Fang, T., Yu, W., Zhang, H., Zhang, Z., Xie, Y., Sycara, K.P., Mi, H., Yu, D.: Vscan: Rethinking visual token reduction for efficient large vision-language models. Trans. Mach. Learn. Res. (2025)
34. Zhang, Y., Fan, C.K., Ma, J., Zheng, W., Huang, T., Cheng, K., Gudovskiy, D., Okuno, T., Nakata, Y., Keutzer, K., Zhang, S.: Sparsevlm: Visual token sparsification for efficient vision-language model inference. arXiv preprint arXiv:2410.04417 (2024)