Fourier Compressor: Frequency-Domain Visual Token Compression for Vision-Language Models

Huanyu Wang[1,4], Jushi Kai[1,4], Haoli Bai[3], Lu Hou[3], Bo Jiang[4], Ziwei He[2,*], and Zhouhan Lin[1,2,*]

[1] LUMIA Lab, School of Artificial Intelligence, Shanghai Jiao Tong University, China
[2] Shanghai Innovation Institute, China
[3] Noah's Ark Lab, Huawei Technologies Ltd., China
[4] School of Computer Science, Shanghai Jiao Tong University, China
[*] Corresponding authors.
whyisverysmart@gmail.com, ziweihe@outlook.com, lin.zhouhan@gmail.com

Abstract. Vision-Language Models (VLMs) incur substantial computational overhead and inference latency due to the large number of visual tokens introduced by high-resolution image and video inputs. Existing parameter-free token compression methods typically rely on token selection or merging, yet they risk discarding substantial visual information or distorting the original representation distribution, resulting in pronounced performance degradation at high compression ratios. In response, we aim to explore a more effective and efficient visual token compression strategy, with a promising direction in the frequency domain. Motivated by the success of frequency-domain transforms in image compression (*e.g.*, JPEG), we systematically analyze the frequency redundancy in visual representations and uncover a non-uniform distribution of semantic information across frequency bands. Building upon this, we introduce **Fourier Compressor**, an effective, parameter-free, and highly generalizable module that removes redundancy from visual representations within the frequency domain. Implemented via FFT with $\mathcal{O}(n^2 \log n)$ complexity and no additional parameters, Fourier Compressor introduces negligible computational overhead while preserving semantic fidelity. Extensive experiments on image-based benchmarks demonstrate that our method achieves a favorable performance-efficiency trade-off, retaining over 96% of the original accuracy while reducing inference FLOPs by up to 83.8% and boosting generation speed by 31.2%. It consistently outperforms existing parameter-free methods and even surpasses some parameterized approaches. Importantly, Fourier Compressor generalizes consistently across both LLaVA and Qwen-VL architectures, and further extends to video understanding tasks, highlighting its practical applicability for efficient VLMs.

Keywords: Vision-Language Models · Token compression · Efficiency

1 Introduction

Vision-Language Models (VLMs) extend Large Language Models (LLMs) with visual understanding capabilities by integrating a vision encoder via a "glue

layer" (*e.g.*, Multi-Layer Perceptron or Q-Former [14]) to jointly model visual and textual information. Nevertheless, the generation of a large number of visual tokens imposes substantial computational demands, as the backbone LLM must process an extremely long context, resulting in significant overhead and high inference latency. For instance, LLaVA-v1.5 [20] produces 576 visual tokens for a single 336×336 image, while Qwen-VL series [1, 29] generates one token per 28×28 pixels, producing several thousand tokens for high-resolution images that often constitute the majority of the input context, far exceeding the number of textual tokens. This challenge is further amplified when processing multiple images and video inputs.

Fortunately, visual representations in VLMs exhibit considerable redundancy, motivating the development of visual token compression. Existing parameter-free methods generally fall into two paradigms. Approaches such as FastV [3], ATP-LLaVA [33], and SparseVLM [38] rely on heuristic token selection, which risks discarding informative visual content. Others, including VisionZip [32] and LLaVA-PruMerge [26], adopt similarity-based token merging, inevitably perturbing the underlying visual representation distribution. While being computationally efficient, both paradigms suffer pronounced performance degradation under high compression ratios. Parameterized alternatives (*e.g.*, VisToG [10], QueCC [15], MQT-LLaVA [9]) mitigate these issues with learnable queries, but incur additional compression-induced computational overhead. Extreme designs like LLaVA-mini [37] introduce extra transformer blocks, resulting in substantial cost and reduced architectural flexibility.

To address these limitations, we aim to develop a more effective, efficient, and generalizable visual token compression strategy. One promising direction lies in the **frequency domain**. Frequency-domain transforms have long been exploited in traditional image compression, where they reveal substantial redundancy and enable highly efficient and generalizable compression methods (*e.g.*, JPEG [28]). Inspired by this, we systematically investigate whether similar frequency redundancy exists in visual representations within VLMs. Our analysis in Sec. 3.2 confirms the presence of such redundancy, uncovering a non-uniform distribution of semantic information across frequency bands. This property can be directly exploited to design parameter-free token compression strategies that are broadly applicable across different VLM architectures.

To this end, we introduce **Fourier Compressor**, an effective, parameter-free, and highly generalizable module for compressing visual tokens in VLMs. By exploiting frequency-domain redundancy in visual representations, our method achieves a favorable trade-off, significantly reducing token counts at high efficiency while preserving generation performance. Fourier Compressor introduces no additional parameters, resulting in minimal compression-induced computational overhead, and generalizes well across multiple VLM architectures, offering an efficient and broadly applicable solution for visual token compression.

The rest of this paper is organized as follows. Sec. 2 reviews related work. Sec. 3 presents preliminaries on the Discrete Cosine Transform and the analyses of frequency components in visual representations. Sec. 4 describes the detailed

design and theoretical time complexity analysis of our method. Sec. 5 presents performances on image-based benchmarks, evaluates empirical efficiency, and demonstrates applicability to video tasks. Finally, Sec. 6 concludes the paper.

2 Related Work

2.1 Visual Token Compression in VLMs

Vision-Language Models (VLMs) have achieved remarkable progress in image and video understanding by integrating LLMs with vision encoders. Despite strong performance from models like LLaVA series [20,22] and Qwen-VL series [1, 29], the high volume of visual tokens remains a major bottleneck for efficient inference and practical deployment.

To address the challenge of long contexts in VLMs, previous research has primarily focused on reducing the number of visual tokens. One common approach involves selecting the most relevant visual tokens or merging less important ones. For example, ATP-LLaVA [33] computes an importance score for each visual token and dynamically determines a pruning threshold to remove redundant tokens within the backbone LLM, while LLaVA-PruMerge [26] clusters and averages visual tokens according to similarity between the class token and spatial tokens. Other methods leverage learnable parameters to extract visual features. MQT-LLaVA [9] employs a Matryoshka Query Transformer, whereas QueCC [15] introduces a query-based convolutional cross-attention module that enables text embeddings to query visual tokens. Furthermore, recent research has also explored transferring visual information to language tokens, such as LLaVA-mini [37], which employs a prefusion module that allows textual tokens to integrate relevant visual information in advance. However, these approaches have yet to find an optimal balance between the additional computational cost of compression and performance degradation. This challenge motivates us to further investigate cost-efficient token compression methods.

2.2 Frequency-based Compression

Frequency domain techniques have long played an important role in signal processing and data compression. In computer vision, frequency transformations are foundational to standard image compression algorithms, such as JPEG [28]. Previous study [31] has also shown that convolutional neural networks (CNNs) exhibit a strong sensitivity to low-frequency channels, revealing an inherent frequency bias in visual feature extraction. Beyond the visual domain, frequency-based approaches have also been widely explored in natural language processing. For instance, FNet [13] replaces the self-attention mechanism in Transformer-like encoders with frequency transformation, achieving comparable performance with significantly reduced computational costs. Fourier-Transformer [8] applies frequency-domain truncation to downsample hidden states in Transformer models for improved efficiency. More recently, FreqKV [12] introduces a frequency-based key-value compression technique that effectively extends the context window of LLMs. When it comes to VLMs, DocPedia [5] leverages DCT coefficients

extracted directly from RGB images to perform vision encoding, enabling higher input resolutions for document understanding tasks. However, the potential of applying frequency-domain techniques to visual-token-level representations remains largely unexplored.

3 Preliminaries

In this section, we introduce the preliminaries of our method, including the formulation of the Discrete Cosine Transform (Sec. 3.1), and the analyses of frequency components in visual representations that reveal semantic redundancy within the frequency domain (Sec. 3.2).

3.1 Discrete Cosine Transform

The Discrete Cosine Transform (DCT) is a linear invertible function $\mathbf{\Psi}: \mathbb{R}^n \rightarrow \mathbb{R}^n$ that converts a sequence of discrete real numbers from the spatial domain to the frequency domain. It exhibits strong energy compaction properties, as most signal information (*e.g.*, audio, image) tends to concentrate in the low-frequency components after transformation. Among the various variants of DCT, we conduct the most widely used type-II DCT.

Formally, for a real-valued sequence $\langle x_i \rangle = \{x_0, x_1, \ldots, x_{N-1}\}$, the DCT transformation is given by:

$$f_m = \alpha_m \sum_{i=0}^{N-1} x_i \cdot \phi_N(m, i), \tag{1}$$

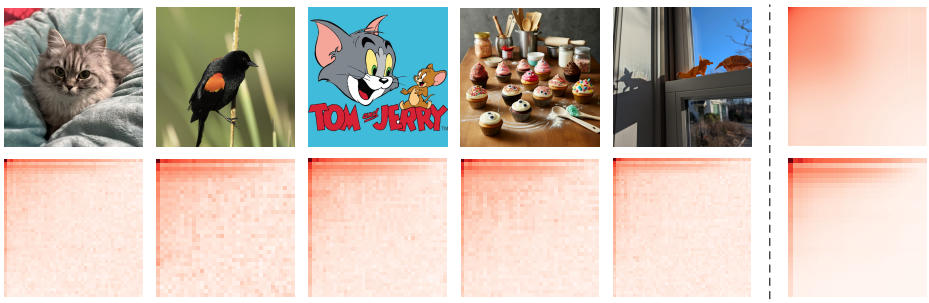
where $m \in \{0, 1, \ldots, N-1\}$. The basis function $\phi_.(\cdot, \cdot)$ and the normalization factor α_m are defined as:

$$\begin{aligned} \phi_N(x, y) &= \cos\left[\frac{\pi}{N} x \left(y + \frac{1}{2}\right)\right] \\ \alpha_m &= \begin{cases} \sqrt{\frac{1}{N}} & \textit{if } m = 0, \\ \sqrt{\frac{2}{N}} & \textit{otherwise.} \end{cases} \end{aligned} \tag{2}$$

The above transformation expresses $\langle x_i \rangle$ as a sum of orthogonal cosine functions at different frequencies, where the coefficients represent the contribution of each frequency component.

Conversely, given the frequency representation $\langle f_m \rangle = \{f_0, f_1, \ldots, f_{N-1}\}$, the original sequence can be recovered by the inverse Discrete Cosine Transform (iDCT):

$$x_i = \sum_{k=0}^{N-1} \alpha_k \cdot f_k \cdot \phi_N(k, i) \tag{3}$$



(a) Cat **(b)** Bird **(c)** Anime **(d)** Cakes **(e)** Origami **(f)** Avg.

Fig. 1: Heatmap visualization of the frequency spectra of visual representations computed by the Qwen2-VL vision encoder. We consider only the magnitude, averaging the absolute values across all hidden dimensions for each frequency component, which is then plotted on a logarithmic scale. In (f), the upper and lower heatmaps show the averaged frequency spectra of raw images and their corresponding visual representations from 10,000 sampled images in GQA, respectively.

3.2 Analyses of Frequency Components

To systematically investigate the potential frequency redundancy of visual representations and examine the semantic information captured by different frequency bands, we conduct three complementary analyses focusing on: (i) energy distribution, (ii) image perturbation, and (iii) semantic-continuous morphing.

Energy distribution. For multiple images, we apply the two-dimensional Discrete Cosine Transform (DCT) to the visual representations produced by the Qwen2-VL [29] vision encoder, and compute the energy of each frequency component by averaging the absolute values across all hidden dimensions. As shown in Fig. 1, the energy of these visual representations exhibits **a strong concentration in low-frequency components** across all hidden dimensions. The prominence of this pattern varies across images of different types, suggesting that the frequency-domain energy distribution inherently encodes structural and semantic information. Importantly, as illustrated in Fig. 1f, the averaged frequency pattern of visual representations closely resembles that of the corresponding raw images, which motivates our exploration of frequency-domain compression, analogous to JPEG [28], for designing more effective and efficient visual token compression strategies.

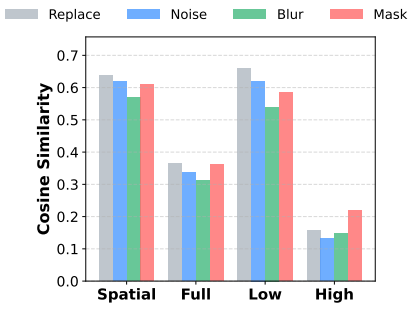


Fig. 2: Cosine similarity of perturbed versus original visual representations for four types of perturbation.

Image Perturbation. For a fixed input image, we apply four types of perturbation: (a) *Replace*, where a fraction of pixels are replaced with uniform pixel values; (b) *Noise*, adding zero-mean Gaus-

sian noise; (c) *Blur*, applying Gaussian blurring; and (d) *Mask*, randomly masking contiguous square regions up to a target area. We extract the corresponding visual representations using the Qwen2-VL [29] encoder, which are then transformed into the frequency domain via the two-dimensional Discrete Cosine Transform (DCT). We compute the cosine similarity between the perturbed and original images for: spatial visual tokens, full frequency components, and the low- (lowest 25%) and high- (highest 25%) frequency components. As illustrated in Fig. 2, **low-frequency components consistently demonstrate higher robustness** across all perturbation types, whereas high-frequency components exhibit pronounced sensitivity, suggesting that low-frequency bands predominantly encode global image structures rather than local appearance details.

Semantic Morphing. We employ a morphing sequence from FreeMorph [2], which exhibits smooth and continuous semantic transitions. Following the same protocol, we compute the cosine similarity between the visual representations of each morphing frame and that of the first (leftmost) image. As shown in Fig. 3, **low-frequency components exhibit substantially greater stability under smooth semantic transitions with consistent global structure** (*e.g.*, frontal pose and facial topology), whereas high-frequency components undergo markedly larger variations, further highlighting their distinct roles in semantic encoding.

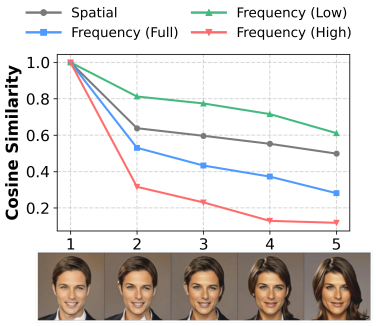


Fig. 3: Cosine similarity of visual representations of morphing frames relative to the leftmost image, for spatial tokens, full-, low- (lowest 25%) and high- (highest 25%) frequency components.

Conclusion. These results reveal a clear functional separation across frequency bands and uncover substantial redundancy in the frequency domain of visual representations: Low-frequency components predominantly capture global, coarse-grained semantic attributes, exhibiting strong robustness to perturbations and smooth transitions under semantic changes, and consequently dominate the overall energy distribution; In contrast, high-frequency components are more sensitive to noise and primarily encode fine-grained details. This observation indicates substantial semantic redundancy in visual representations, providing strong empirical motivation for our frequency-domain token compression strategy, which selectively preserves low-frequency information to achieve efficient token reduction while maintaining semantic fidelity.

4 Method

In this section, we introduce **Fourier Compressor**, a parameter-free and highly generalizable module for visual token compression. Building upon a standard VLM architecture, our framework preserves the vision encoder, projector, and

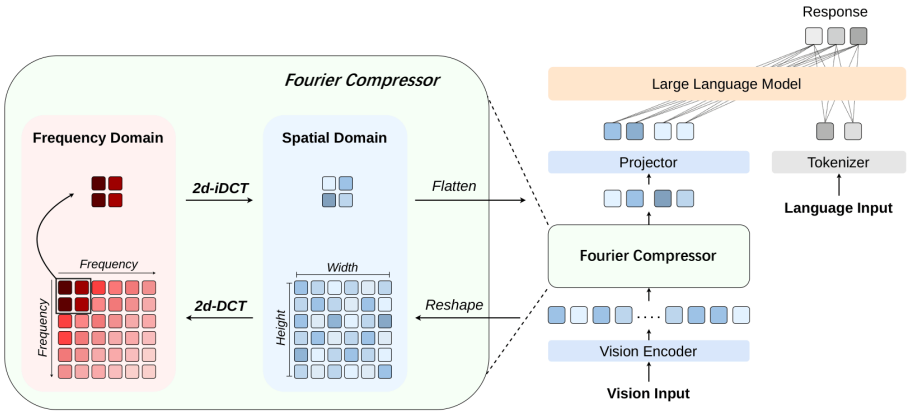


Fig. 4: Illustration of Fourier Compressor. After passing through the vision encoder, visual features are reshaped into a grid and transformed into the frequency domain. Darker colors indicate larger frequency magnitudes, while lighter colors represent smaller magnitudes. Only the low-frequency components are retained and subsequently converted back to the spatial domain, serving as the compressed visual features.

backbone LLM, and inserts the Fourier Compressor after the vision encoder to substantially reduce the number of visual tokens. Below we detail the overall architecture (Sec. 4.1), the design of the Fourier Compressor (Sec. 4.2), and the theoretical time complexity analysis (Sec. 4.3).

4.1 General Architecture

The overall architecture of our framework is illustrated in Fig. 4. The original visual input is first preprocessed (*e.g.*, resizing or cropping for images, frame extraction for videos) before entering the vision encoder. The 3-channel image $\boldsymbol{X}^v \in \mathbb{R}^{h \times w \times 3}$, where h, w represent the input resolution, is initially passed through a CNN to extract low-level features, resulting in grid image features $\hat{\boldsymbol{H}}^v \in \mathbb{R}^{N_h \times N_w \times h_c}$ where $N_h N_w$ is the number of patches and h_c is the output dimension of the CNN. These grid features are then flattened and passed through a pretrained ViT, which encodes the visual information into higher-level representations:

$$\boldsymbol{H}^v = Vision\text{-}Encoder(\boldsymbol{X}^v), \tag{4}$$

where $\boldsymbol{H}^v \in \mathbb{R}^{N_h N_w \times h_v}$ and h_v is the output dimension of the vision encoder.

Subsequently, the visual features are processed by our Fourier Compressor, which reduces the number of visual tokens from $N_h N_w$ to $C_h C_w$:

$$\boldsymbol{H}_c^v = Fourier\text{-}Compressor(\boldsymbol{H}^v), \tag{5}$$

where $\boldsymbol{H}_c^v \in \mathbb{R}^{C_h C_w \times h_v}$.

Finally, the compressed visual features are passed through the projector to align with the text embedding space, and processed alongside the text instructions by the backbone LLM, following the typical VLM architecture.

4.2 Fourier Compressor

Inside the Fourier Compressor, the original visual features are first reshaped back to the grid size:

$$\boldsymbol{G}^v = Reshape(\boldsymbol{H}^v), \tag{6}$$

where $\boldsymbol{G}^v \in \mathbb{R}^{N_h \times N_w \times h_v}$. Then, a two-dimensional Discrete Cosine Transform is applied along the two spatial dimensions to obtain the frequency representation $\hat{\boldsymbol{F}}^v$. Formally:

Algorithm 1 Fourier Compressor

Input: Image feature I of shape $(B, N_h N_w, h_v)$
Output: Compressed I^c of shape $(B, C_h C_w, h_v)$
$X \leftarrow \text{Reshape}(\text{Transpose}(I, (1,2)), (B, h_v, N_h, N_w))$
$F_1 \leftarrow \text{DCT}(X)$
$F_2 \leftarrow \text{DCT}(\text{Transpose}(F_1, (2,3)))$
$F \leftarrow \text{Transpose}(F_2, (2,3))$
$F^c \leftarrow F[:, :, : C_h, : C_w]$
$R_1 \leftarrow \text{IDCT}(F^c)$
$R_2 \leftarrow \text{IDCT}(\text{Transpose}(R_1, (2,3)))$
$Y \leftarrow \text{Transpose}(R_2, (2,3))$
$I^c \leftarrow \text{Transpose}(\text{Reshape}(Y, (B, h_v, C_h C_w)), (1,2))$
return I^c

$$\hat{\boldsymbol{F}}^v_{m,n,:} = \alpha_m \alpha_n \sum_{p=0}^{N_h-1} \sum_{q=0}^{N_w-1} \boldsymbol{G}^v_{p,q,:} \cdot \phi_{N_h}(m,p) \cdot \phi_{N_w}(n,q), \tag{7}$$

where $\hat{\boldsymbol{F}}^v \in \mathbb{R}^{N_h \times N_w \times h_v}$ and $\phi_\cdot(\cdot,\cdot)$, α_m are specified in Eq. (2). As analyzed in Sec. 3.2, low-frequency components capture more global and semantically informative structures, which motivates the pruning of high-frequency components for compression:

$$\boldsymbol{F}^v = \hat{\boldsymbol{F}}^v[0 : C_h, 0 : C_w, :], \tag{8}$$

where $\boldsymbol{F}^v \in \mathbb{R}^{C_h \times C_w \times h_v}$ and $C_h C_w$ is the number of preserved visual tokens.

Finally, we apply a two-dimensional inverse Discrete Cosine Transform (2d-iDCT) to reconstruct the spatial features from the frequency-domain tokens:

$$\boldsymbol{T}^v_{i,j,:} = \sum_{p=0}^{C_h-1} \sum_{q=0}^{C_w-1} \alpha_p \alpha_q \cdot \boldsymbol{F}^v_{p,q,:} \cdot \phi_{C_h}(p,i) \cdot \phi_{C_w}(q,j), \tag{9}$$

where $\boldsymbol{T}^v \in \mathbb{R}^{C_h \times C_w \times h_v}$, which is then flattened to obtain the compressed image features:

$$\boldsymbol{H}^v_c = Flatten\left(\boldsymbol{T}^v\right), \tag{10}$$

where $\boldsymbol{H}^v_c \in \mathbb{R}^{C_h C_w \times h_v}$.

4.3 Time Complexity

A direct computation of the Discrete Cosine Transform (DCT) is inefficient, requiring $\mathcal{O}(N^2)$ operations for an N-point sequence. However, by leveraging the

Fast Fourier Transform (FFT) operator, we can accelerate the DCT computation to $\mathcal{O}(N \log N)$, same for iDCT.

For a real-valued sequence of length N, denoted as $\langle x_i \rangle = \{x_0, x_1, \ldots, x_{N-1}\}$, we first rearrange it by separating the even- and odd-indexed elements and reversing the order of the odd-indexed elements. If N is even, the reordered sequence $\langle y_k \rangle$ is defined as:

$$y_k = \begin{cases} x_{2k} & k = 0, 1, \ldots, \frac{N-2}{2} \\ x_{2N-1-2k} & k = \frac{N}{2}, \ldots, N-1 \end{cases} \tag{11}$$

That is, $\langle y_k \rangle = \{x_0, x_2, \ldots, x_{N-2}, x_{N-1}, x_{N-3}, \ldots, x_1\}$. A similar rearrangement applies when N is odd.

Next, we compute the FFT of $\langle y_n \rangle$, yielding the sequence $\langle z_m \rangle$:

$$\langle z_m \rangle = FFT(\langle y_n \rangle), \tag{12}$$

where $\langle z_m \rangle$ is a complex sequence of length N. The DCT coefficients can then be computed as:

$$f_k = \Re(z_k) \cdot \cos\left(\frac{k\pi}{2N}\right) - \Im(z_k) \cdot \sin\left(\frac{k\pi}{2N}\right), \tag{13}$$

where $\Re(\cdot)$ and $\Im(\cdot)$ denote the real and imaginary parts of a complex number, respectively.

For the two-dimensional DCT (2d-DCT) applied to a matrix $\boldsymbol{X} \in \mathbb{R}^{N_h \times N_w}$, the transformation is equivalent to performing a one-dimensional DCT along each row, followed by another one-dimensional DCT along each column. Consequently, applying 2d-DCT to an $N_h \times N_w$ matrix results in a total time complexity of:

$$\mathcal{O}(N_h N_w \log(N_h N_w)). \tag{14}$$

Therefore, assuming $N_h = N_w = N, C_h = C_w = C$, our Fourier Compressor, which first applies a 2d-DCT on the grid image features of size $N_h \times N_w \times h_v$ and then a 2d-iDCT on the compressed image features of size $C_h \times C_w \times h_v$, has an overall time complexity of:

$$\mathcal{O}(N^2 \log N + C^2 \log C). \tag{15}$$

Tab. 1 summarizes the time complexity of representative modules operating on inputs of shape (B, N^2, h_v). Under a typical regime where $h_v = \mathcal{O}(10^3)$, $N^2 = \mathcal{O}(10^2)$, and $C^2 = \mathcal{O}(10)$, Fourier Compressor attains the lowest theoretical complexity, substantially outperforming attention-based (*e.g.*, ATP-LLaVA [33]) and query-based (*e.g.*, MQT-LLaVA [9]) compression methods in computational efficiency. Empirical evaluations further validate this efficiency advantage, as detailed in Sec. 5.4.

Table 1: Time complexity of common modules processing inputs of shape (B, N^2, h_v). C^2 denotes the number of learnable queries.

Module	Time Complexity
MLP	$\mathcal{O}(B \cdot h_v^2 \cdot N^2)$
Self-Attention	$\mathcal{O}(B \cdot h_v \cdot N^4)$
Query Transformer	$\mathcal{O}(B \cdot h_v \cdot N^2 \cdot C^2)$
Fourier Compressor	$\mathcal{O}(B \cdot h_v \cdot N^2 \log N)$

5 Experiments

In this section, we first describe the training strategy and evaluation settings (Sec. 5.1). Then we present results on a wide range of image-based benchmarks (Sec. 5.2) and demonstrate the generalizability of the Fourier Compressor across different VLM architectures (Sec. 5.3), followed by empirical efficiency metrics (Sec. 5.4). Finally, we evaluate its applicability to video understanding tasks (Sec. 5.5), highlighting its practical utility and broad applicability.

5.1 Experimental Setup

Since frequency-domain truncation inevitably alters the feature distribution, additional training is required for the model to adapt to the compressed visual representations. Detailed training configurations are provided in Tab. 2.

Fourier-LLaVA incorporates the Fourier Compressor into the LLaVA-v1.5 models [20]. Training follows the original two-stage protocol of the base model: first aligning compressed visual features with the language embedding space, then optimizing for multimodal instruction following.

Fourier-Qwen incorporates the Fourier Compressor into the Qwen-VL series [1, 29], inserting it after the original MLP-based merger. We perform continued finetuning on 600k single-image conversation samples from LLaVA-NeXT [21].

Table 2: Training configurations for Fourier-LLaVA and Fourier-Qwen.

Settings	**Fourier-LLaVA**		**Fourier-Qwen**
	Stage 1	Stage 2	
Trainable modules			
Vision Encoder			✓
Projector	✓	✓	✓
LLM		✓	✓
Dataset	558k	665k	600k
Epochs	1	2	2
LoRA r/α	-	128 / 256	-
Batch Size	256	256	128
Learning Rate	1e-3	2e-4	1e-6
MM LR	-	2e-5	1e-5
Vision LR	-	-	1e-6
Optimizer	AdamW		AdamW
Schedule	Cosine		Cosine
Warmup Ratio	0.03		0.03

We evaluate on eight image-based benchmarks covering VQA, domain knowledge, and text recognition [7, 11, 18, 22–24, 27, 34]. For Fourier-Qwen, three additional benchmarks are included [4, 6, 30]. We primarily use *lmms-eval* [36] for evaluation. For benchmarks not supported by lmms-eval (VQA^{T}, MMB, and LLaVA^{W}), we follow the official scripts from LLaVA-v1.5. For LLaVA^{W}, performance is measured with *gpt-4-0613* at temperature 0.2.

5.2 Image Benchmark Performance

We begin by evaluating the effectiveness of the proposed Fourier Compressor on image-based multimodal benchmarks. Specifically, we integrate the compressor into the LLaVA-v1.5 backbone and systematically vary the number of preserved visual tokens from 36 to 256.

Tab. 3 presents the results on eight image benchmarks. Notably, even with a reduced number of visual tokens, Fourier-LLaVA achieves competitive results.

Table 3: Performance of Fourier-LLaVA on 8 image-based benchmarks. "#I" denotes the number of visual tokens per image. All baseline results are reported from original papers, except MQT-LLaVA on VQA^T, which was not originally reported and is newly evaluated by us.

Model	#I	VQA^{v2}	GQA	SQA	VQA^T	POPE	MMB	$LLaVA^W$	MMMU	Avg.
LLaVA-v1.5-7B	576	78.5	62.0	66.8	58.2	85.9	64.3	65.4	35.3	64.6
Compression ratio: 55.6%										
MQT-LLaVA [9]	256	76.8	61.6	67.6	53.2	84.4	64.3	**64.6**	**34.8**	63.4
Fourier-LLaVA	256	**78.6**	**62.7**	**69.9**	**56.0**	**85.3**	**66.4**	64.4	33.1	**64.6**
Compression ratio: 75.0%										
ATP-LLaVA [33]	144	76.4	59.5	**69.1**	-	84.2	**66.0**	-	-	-
MQT-LLaVA [9]	144	76.4	61.4	67.5	52.6	83.9	64.4	61.4	34.4	62.8
Prumerge [26]	144	76.8	-	68.3	**57.1**	84.0	64.9	-	-	-
Fourier-LLaVA	144	**77.7**	**61.7**	69.0	54.7	**85.0**	65.6	**67.8**	**35.3**	**64.6**
Compression ratio: 88.9%										
ATP-LLaVA [33]	88	73.3	56.8	67.2	-	82.6	**64.7**	-	-	-
MQT-LLaVA [9]	64	75.3	60.0	67.0	51.7	83.6	63.5	59.4	**34.4**	61.9
Fourier-LLaVA	64	**76.3**	**60.4**	**69.3**	**52.6**	**85.3**	**64.7**	**63.1**	**34.4**	**63.3**
Compression ratio: 93.75%										
MQT-LLaVA [9]	36	73.7	58.8	66.8	50.4	81.9	63.4	59.6	**34.4**	61.1
PruMerge [26]	32	72.0	-	68.5	**56.0**	76.3	60.9	-	-	-
Fourier-LLaVA	36	**74.9**	**59.5**	**69.0**	51.0	**84.0**	**64.3**	**61.1**	32.8	**62.1**
LLaVA-v1.5-13B	576	80.0	63.3	71.6	61.3	85.9	67.7	72.5	36.4	67.3
Compression ratio: 75.0%										
PruMerge [26]	144	77.8	-	71.0	**58.6**	84.4	65.7	-	-	-
Fourier-LLaVA	144	**78.7**	62.7	**71.1**	57.4	**85.4**	**66.2**	69.5	35.8	65.9

With 256 visual tokens (44% of the original tokens), Fourier-LLaVA achieves superior performance to the vanilla LLaVA-v1.5-7B on four benchmarks. Moreover, with just 144 visual tokens ($25\% \times 576$), it still surpasses the base model in terms of average score. Even at the lowest setting of 36 visual tokens (6.25% of the original tokens), Fourier-LLaVA only exhibits a 3.87% drop in average, while continuing to outperform the base model on SciQA and MMB.

For other token compression approaches, we compare the performance under the same number of visual tokens. Note that all the methods mentioned are implemented based on LLaVA-v1.5 series and require a two-stage training process. Generally, Fourier-LLaVA outperforms prior parameter-free methods such as ATP-LLaVA and LLaVA-PruMerge across nearly all benchmarks at the same visual token count, substantially improving generation performance under high compression ratios. Furthermore, it surpasses MQT-LLaVA, which relies on additional learnable parameters, achieving a 2.2% increase in average scores, further demonstrating the effectiveness of our approach.

Table 4: Performance of Fourier-Qwen series. "#I" denotes the average number of visual tokens per image. We report the performance of all models using *lmms-eval*, with the number of visual tokens standardized to 256 - 2304 for fair comparison.

Model Series		#I	Avg.	MME	VQA^T	POPE	RWQA	MMB	MMS
Qwen2-VL-2B	Vanilla	553	66.8	1894.4	78.2	86.1	61.4	72.3	44.2
	Fourier-	236	65.7	1917.1	75.4	86.5	61.6	72.6	43.3
		-57.3%	*-1.6%*	*+1.2%*	*-3.6%*	*+0.5%*	*+0.3%*	*+0.4%*	*-2.0%*
Qwen2.5-VL-3B	Vanilla	553	71.1	2138.3	77.6	87.2	59.6	77.8	56.2
	Fourier-	236	72.6	2110.8	76.1	87.8	64.3	76.8	55.3
		-57.3%	*+2.1%*	*-1.3%*	*-1.9%*	*+0.7%*	*+7.9%*	*-1.3%*	*-1.6%*

Moreover, when scaled to a 13B backbone, Fourier-LLaVA continues to outperform LLaVA-PruMerge under the same visual token budget. Besides, with only 25% of the original visual tokens, our approach incurs only a 2.1% drop in the average score compared to the base model.

Overall, these results demonstrate that our Fourier Compressor achieves superior performance when compressing visual tokens for image inputs, maintaining strong robustness under aggressive token compression. Besides, it also exhibits strong compression efficiency, as detailed in Subsection 5.4.

5.3 Generalization Across Architectures

To assess the generalizability of the Fourier Compressor beyond the LLaVA series, we further apply it to Qwen2-VL-2B and Qwen2.5-VL-3B [1, 29], and evaluate on six image-based benchmarks. This evaluation examines whether frequency-domain compression remains effective across different vision encoders, backbone LLMs, and input resolutions.

As shown in Tab. 4, our method consistently preserves strong performance under substantial token reduction. With a 57.3% decrease in visual tokens, Fourier-Qwen2-VL-2B incurs only a marginal 1.6% drop in average performance, while even achieving improvements on MME, POPE, RWQA, and MMB. Notably, on Qwen2.5-VL-3B, the compressed model surpasses the vanilla baseline by 2.1% on average, demonstrating that token reduction does not necessarily compromise reasoning capability. In particular, the 7.9% gain on RealWorldQA suggests that frequency-domain compression may help suppress redundant or noisy visual components, benefiting real-world question answering tasks.

Overall, these results indicate that the effectiveness of the Fourier Compressor is not architecture-specific. It generalizes robustly across model scales and input resolutions, maintaining or even improving performance despite aggressive visual token compression.

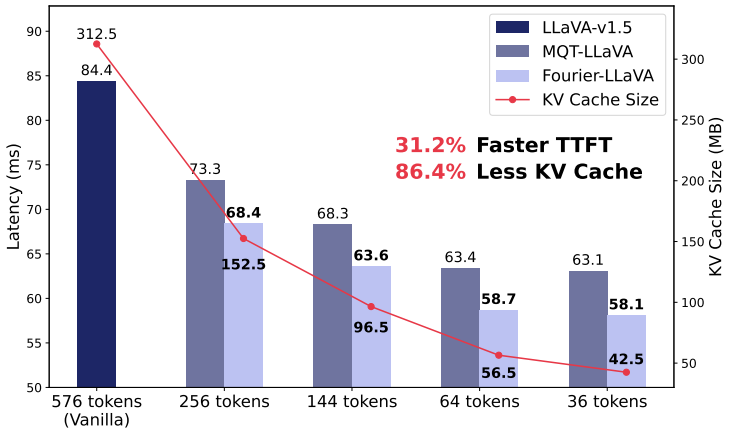


Fig. 5: Latency and KV cache usage of Fourier-LLaVA under varying numbers of preserved visual tokens.

5.4 Floating Point Operations and Latency

Beyond effectiveness, a key advantage of the proposed Fourier Compressor lies in its parameter-free design and inherent computational efficiency due to its FFT-based implementation, as discussed in Sec. 4.3. To empirically validate that these theoretical advantages translate into practical gains, we conduct a systematic evaluation of the efficiency of Fourier Compressor.

We evaluate the FLOPs (Floating Point Operations) of Fourier-LLaVA under varying numbers of visual tokens using *calflops* on an RTX 4090 (24G) GPU. Additionally, we measure the TTFT (Time to First Token) and KV cache usage on an A100 (40G) GPU.

As illustrated in Tab. 5 and Fig. 5, Fourier-LLaVA achieves a substantial reduction in computational cost compared to LLaVA-v1.5-7B, reducing FLOPs by 83.8%. Furthermore, our method lowers KV cache usage by 86.4% and improves inference speed by 31.2% in TTFT, outperforming other token compression baselines such as MQT-LLaVA under equivalent visual token counts.

Table 5: Floating Point Operations of Fourier-LLaVA under varying numbers of preserved visual tokens (#I).

Model	#I	FLOPs (T)	↓ (%)
LLaVA-v1.5-7B	576	8.54	-
Fourier-LLaVA	256	4.30	49.6
	144	2.81	67.1
	64	1.75	79.5
	36	1.38	83.8

These improvements highlight the efficiency of our parameter-free compression approach, distinguishing Fourier Compressor from prior works that rely on attention-based reductions or learnable parameters, which is crucial for enabling real-time deployment of VLMs on resource-constrained devices.

Table 6: Performance of Fourier-LLaVA and Fourier-Qwen on MVBench. "#V" denotes the average number of visual tokens per video. For fair comparison, when evaluating Fourier-Qwen series, the number of visual tokens per frame is set to 256 - 384, and the maximum number of frames is 16.

Model	Size	#V	Avg.	Ac	Ob	Posi	Sc	Cou	At	Pose	Ch	Cog
Video-ChatGPT [25]	7B	356	32.7	32.1	40.7	21.5	31.0	28.0	44.0	29.0	33.0	30.3
Video-LLaMA [35]	7B	40	34.1	34.4	42.2	22.5	43.0	28.3	39.0	32.5	40.0	29.3
Video-LLaVA [19]	7B	4096	43.1	48.0	46.5	27.8	84.5	35.5	45.8	34.0	42.5	34.2
VideoChat [16]	7B	4096	35.5	38.0	41.2	26.3	48.5	27.8	44.3	26.5	41.0	27.7
VideoChat2 [17]	7B	4096	51.1	61.3	57.3	23.0	88.5	40.5	51.3	49.0	36.5	47.0
LLaVA-v1.5 [20]	7B	2304	45.6	52.8	43.7	31.5	83.5	37.0	45.3	44.5	50.0	37.3
Fourier-LLaVA	7B	288	44.0	48.1	46.3	31.0	81.0	40.5	41.3	36.0	49.0	36.3
Δ vs. LLaVA-v1.5		*-87.5%*	*-1.6*	*-4.7*	*+2.6*	*-0.5*	*-2.5*	*+3.5*	*-4.0*	*-8.5*	*-1.0*	*-1.0*
Qwen2-VL [29]	2B	3193	61.4	68.5	65.7	44.8	90.5	60.8	65.3	53.5	59.0	47.8
Fourier-Qwen2	2B	1328	59.8	67.7	63.3	40.0	88.5	55.5	64.8	51.0	57.0	50.2
Δ vs. Qwen2-VL		*-58.4%*	*-1.6*	*-0.8*	*-2.4*	*-4.8*	*-2.0*	*-5.3*	*-0.5*	*-2.5*	*-2.0*	*+2.4*
Qwen2.5-VL [1]	3B	3193	65.2	68.3	67.7	48.8	91.0	63.0	75.0	51.5	75.5	56.0
Fourier-Qwen2.5	3B	1328	62.9	67.2	65.7	48.5	90.5	54.8	70.5	47.5	66.0	57.5
Δ vs. Qwen2.5-VL		*-58.4%*	*-2.3*	*-1.1*	*-2.0*	*-0.3*	*-0.5*	*-8.2*	*-4.5*	*-4.0*	*-9.5*	*+1.5*

5.5 Applicability to Video Tasks

Although our method is primarily designed for visual token compression in image inputs, the proposed Fourier Compressor is inherently model-agnostic and parameter-free, enabling zero-shot application to video tasks. To demonstrate this extension capability, we evaluate both Fourier-LLaVA and Fourier-Qwen models on MVBench [17], a comprehensive multi-modal video understanding benchmark comprising 20 challenging tasks.

As shown in Tab. 6, our method leverages significantly fewer visual tokens while outperforming various open-source VLMs. Specifically, Fourier-Qwen series achieves a 58.4% reduction in visual tokens with only a 3.1% drop in average performance, while Fourier-LLaVA retains 96.5% of the base model's performance utilizing merely 12.5% of the video tokens, and still surpasses the base model on the Object and Count tasks. This zero-shot capability in video understanding highlights the efficiency and robustness of our method, and suggests promising directions for future work on scaling frequency-aware compression to visual representations of video inputs.

6 Conclusion

In this paper, we propose **Fourier Compressor**, an efficient, effective, and highly generalizable framework for visual token compression in Vision-Language

Models. Motivated by the non-uniform frequency distribution of semantic information, our method significantly reduces frequency-domain redundancy from visual representations. Empirically, it achieves a favorable trade-off between performance and efficiency: Without introducing any additional parameters, it preserves over 96% average accuracy across eight image-based benchmarks while retaining only 6.25% of the original visual tokens, leading to a substantial reduction in FLOPs to 16.16% and an inference speedup of 31.2%. Seamlessly integrated into both the LLaVA-v1.5 and Qwen-VL series, Fourier Compressor exhibits strong generalization across model architectures, and further demonstrates promising performance on video understanding tasks. These results collectively highlight a favorable trade-off between computational cost and model performance, facilitating more efficient and scalable deployment of VLMs in real-world scenarios.

References

1. 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)
2. Cao, Y., Si, C., Wang, J., Liu, Z.: FreeMorph: Tuning-Free Generalized Image Morphing with Diffusion Model. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 18111–18120 (2025)
3. 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. In: European Conference on Computer Vision. pp. 19–35. Springer (2024)
4. 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? In: Advances in Neural Information Processing Systems. vol. 37, pp. 27056–27087 (2024)
5. Feng, H., Liu, Q., Liu, H., Tang, J., Zhou, W., Li, H., Huang, C.: DocPedia: Unleashing the Power of Large Multimodal Model in the Frequency Domain for Versatile Document Understanding. Science China Information Sciences **67**(12), 220106 (2024)
6. Fu, C., Chen, P., Shen, Y., Qin, Y., Zhang, M., Lin, X., Yang, J., Zheng, X., Li, K., Sun, X., Wu, Y., Ji, R., Shan, C., He, R.: MME: A Comprehensive Evaluation Benchmark for Multimodal Large Language Models. In: Advances in Neural Information Processing Systems. vol. 38 (2025)
7. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: 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. pp. 6904–6913 (2017)
8. He, Z., Yang, M., Feng, M., Yin, J., Wang, X., Leng, J., Lin, Z.: Fourier Transformer: Fast Long Range Modeling by Removing Sequence Redundancy with FFT Operator. In: Findings of the Association for Computational Linguistics: ACL 2023. pp. 8954–8966 (2023)

9. Hu, W., Dou, Z.Y., Li, L.H., Kamath, A., Peng, N., Chang, K.W.: Matryoshka Query Transformer for Large Vision-Language Models. In: Advances in Neural Information Processing Systems. vol. 37, pp. 50168–50188 (2024)
10. Huang, M., Huang, R., Shi, H., Chen, Y., Zheng, C., Sun, X., Jiang, X., Li, Z., Cheng, H.: Efficient Multi-modal Large Language Models via Visual Token Grouping. arXiv preprint arXiv:2411.17773 (2024)
11. Hudson, D.A., Manning, C.D.: 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. pp. 6700–6709 (2019)
12. Kai, J., Wang, Y., Zeng, B., Bai, H., Jiang, B., He, Z., Lin, Z.: FreqKV: Key-Value Compression in Frequency Domain for Context Window Extension. In: International Conference on Learning Representations (2026)
13. Lee-Thorp, J., Ainslie, J., Eckstein, I., Ontanon, S.: FNet: Mixing Tokens with Fourier Transforms. In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 4296–4313 (2022)
14. Li, J., Li, D., Savarese, S., Hoi, S.: BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In: International Conference on Machine Learning. pp. 19730–19742. PMLR (2023)
15. Li, K., Goyal, S., Semedo, J.D., Kolter, Z.: Inference Optimal VLMs Need Fewer Visual Tokens and More Parameters. In: International Conference on Learning Representations. vol. 2025, pp. 96066–96083 (2025)
16. Li, K., He, Y., Wang, Y., Li, Y., Wang, W., Luo, P., Wang, Y., Wang, L., Qiao, Y.: VideoChat: Chat-centric Video Understanding. Science China Information Sciences **68**(10), 200102 (2025)
17. Li, K., Wang, Y., He, Y., Li, Y., Wang, Y., Liu, Y., Wang, Z., Xu, J., Chen, G., Luo, P., et al.: MVBench: A Comprehensive Multi-modal Video Understanding Benchmark. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22195–22206 (2024)
18. Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, W.X., Wen, J.R.: Evaluating Object Hallucination in Large Vision-Language Models. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. pp. 292–305 (2023)
19. Lin, B., Ye, Y., Zhu, B., Cui, J., Ning, M., Jin, P., Yuan, L.: Video-LLaVA: Learning United Visual Representation by Alignment Before Projection. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing. pp. 5971–5984 (2024)
20. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved Baselines with Visual Instruction Tuning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 26296–26306 (2024)
21. Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., Lee, Y.J.: LLaVA-NeXT: Improved Reasoning, OCR, and World Knowledge. `https://llava-vl.github.io/blog/2024-01-30-llava-next` (2024), accessed: June 27, 2026
22. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual Instruction Tuning. In: Advances in Neural Information Processing Systems. vol. 36, pp. 34892–34916 (2023)
23. 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? In: European Conference on Computer Vision. pp. 216–233. Springer (2024)
24. Lu, P., Mishra, S., Xia, T., Qiu, L., Chang, K.W., Zhu, S.C., Tafjord, O., Clark, P., Kalyan, A.: Learn to Explain: Multimodal Reasoning via Thought Chains for Science Question Answering. In: Advances in Neural Information Processing Systems. vol. 35, pp. 2507–2521 (2022)

25. Maaz, M., Rasheed, H., Khan, S., Khan, F.: Video-ChatGPT: Towards Detailed Video Understanding via Large Vision and Language Models. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 12585–12602 (2024)
26. Shang, Y., Cai, M., Xu, B., Lee, Y.J., Yan, Y.: LLaVA-PruMerge: Adaptive Token Reduction for Efficient Large Multimodal Models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 22857–22867 (2025)
27. Singh, A., Natarajan, V., Shah, M., Jiang, Y., Chen, X., Batra, D., Parikh, D., Rohrbach, M.: Towards VQA Models That Can Read. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8317–8326 (2019)
28. Wallace, G.K.: The JPEG Still Picture Compression Standard. Communications of the ACM **34**(4), 30–44 (1991)
29. Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., Fan, Y., Dang, K., Du, M., Ren, X., Men, R., Liu, D., Zhou, C., Zhou, J., Lin, J.: Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution. arXiv preprint arXiv:2409.12191 (2024)
30. xAI: RealWorldQA: A Benchmark for Real-World Understanding. `https://huggingface.co/datasets/xai-org/RealworldQA` (2024), accessed: June 27, 2026
31. Xu, K., Qin, M., Sun, F., Wang, Y., Chen, Y.K., Ren, F.: Learning in the Frequency Domain. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1740–1749 (2020)
32. 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. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19792–19802 (2025)
33. Ye, X., Gan, Y., Ge, Y., Zhang, X.P., Tang, Y.: ATP-LLaVA: Adaptive Token Pruning for Large Vision Language Models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 24972–24982 (2025)
34. 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. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9556–9567 (2024)
35. Zhang, H., Li, X., Bing, L.: Video-LLaMA: An Instruction-tuned Audio-Visual Language Model for Video Understanding. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. pp. 543–553 (2023)
36. Zhang, K., Li, B., Zhang, P., Pu, F., Cahyono, J.A., Hu, K., Liu, S., Zhang, Y., Yang, J., Li, C., et al.: LMMs-Eval: Reality Check on the Evaluation of Large Multimodal Models. In: Findings of the Association for Computational Linguistics: NAACL 2025. pp. 881–916 (2025)
37. Zhang, S., Fang, Q., Yang, Y., Feng, Y.: LLaVA-Mini: Efficient Image and Video Large Multimodal Models with One Vision Token. In: International Conference on Learning Representations. vol. 2025, pp. 53285–53310 (2025)
38. Zhang, Y., Fan, C.K., Ma, J., Zheng, W., Huang, T., Cheng, K., Gudovskiy, D., Okuno, T., Nakata, Y., Keutzer, K., et al.: SparseVLM: Visual Token Sparsification for Efficient Vision-Language Model Inference. arXiv preprint arXiv:2410.04417 (2024)