
SCOPE: Saliency-Coverage Oriented Token Pruning for Efficient Multimodal LLMs

Jinhong Deng^{1,3,4}, Wen Li^{2*}, Joey Tianyi Zhou^{3,4}, Yang He^{3,4}

¹University of Electronic Science and Technology of China (UESTC)

²Shenzhen Institute for Advanced Study, UESTC

³CFAR, Agency for Science, Technology and Research (A*STAR), Singapore

⁴IHPC, Agency for Science, Technology and Research (A*STAR), Singapore

{jhdengvision, liwenbnu}@gmail.com,

{joey_zhou, he_yang}@a-star.edu.sg

Abstract

Multimodal Large Language Models (MLLMs) typically process a large number of visual tokens, leading to considerable computational overhead, even though many of these tokens are redundant. Existing visual token pruning methods primarily focus on selecting the most salient tokens based on attention scores, resulting in the semantic incompleteness of the selected tokens. In this paper, we propose a novel visual token pruning strategy, called **Saliency-Coverage Oriented token Pruning for Efficient MLLMs (SCOPE)**, to jointly model both the saliency and coverage of the selected visual tokens to better preserve semantic completeness. Specifically, we introduce a set-coverage for a given set of selected tokens, computed based on the token relationships. We then define a token-coverage gain for each unselected token, quantifying how much additional coverage would be obtained by including it. By integrating the saliency score into the token-coverage gain, we propose our SCOPE score and iteratively select the token with the highest SCOPE score. We conduct extensive experiments on multiple vision-language understanding benchmarks using the LLaVA-1.5 and LLaVA-Next models. Experimental results demonstrate that our method consistently outperforms prior approaches. Our code is available at <https://github.com/kinredon/SCOPE>.

1 Introduction

Recent advances in Multimodal Large Language Models (MLLMs) [24, 25, 51, 20, 21] have significantly advanced open-ended visual understanding tasks [12, 27, 47, 8] by integrating powerful vision encoders [34] with autoregressive large language models [37, 1]. These systems typically tokenize visual inputs into sequences of patch-level embeddings (*i.e.*, visual tokens), which are then fed into the language model via either projection modules [24] or attention-based fusion mechanisms [19]. Despite its effectiveness, this paradigm incurs substantial computational overhead, particularly when processing high-resolution images or temporally dense video inputs. For instance, a ViT encoder [11] applied to a 448×448 image can generate over 1,000 visual tokens. This number increases rapidly in high-resolution and video scenarios involving multiple frames. Since these tokens are jointly processed with textual tokens, the computational cost of self-attention grows quadratically with the number of visual tokens [30, 25], limiting their deployment in practical applications such as edge computing and robotics [17, 33, 44].

*Corresponding Author.

This work was completed during Jinhong Deng’s internship at CFAR, A*STAR.

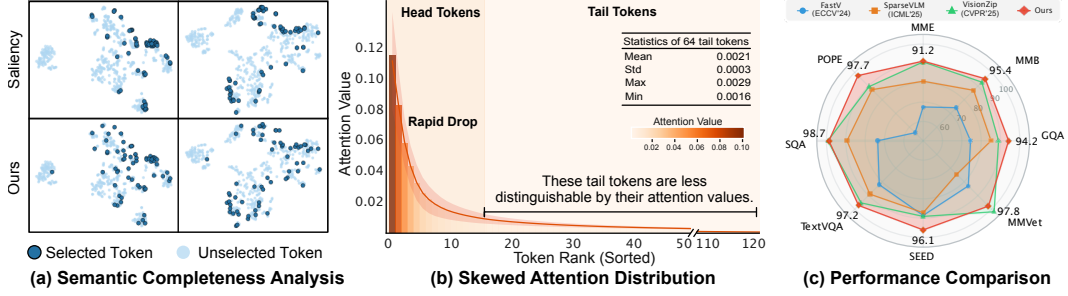


Figure 1: (a) Semantic Completeness Analysis. We visualize the selected tokens using a saliency-based rule (Top) and our method (Bottom). The saliency score corresponds to the visual attention assigned to the CLS token. Our method selects tokens that maximize coverage while preserving the most dominant visual information. (b) Skewed Attention Distribution. We show the averaged attention distribution of the top 128 tokens on the MME benchmark. The attention weights rapidly flatten, making tail tokens less distinguishable based on their attention values. (c) Performance comparison with prior methods across various benchmarks. The model is LLaVA-1.5 7B, and the number of retained tokens is 64.

However, not all visual tokens contribute equally to the final outputs of the language model [7]. Many background or repetitive patches carry redundant or less informative content [6, 11]. This motivates the need for efficient visual token pruning or compression, aiming to retain only the most relevant tokens while discarding those that are redundant. To this end, recent works [7, 41, 49] have proposed various pruning strategies that select salient visual tokens based on attention scores, *i.e.*, visual attention from text prompts or from the CLS token in vision transformers. For instance, VisionZIP [43] selects visual tokens that receive the highest attention from the CLS token.

While effective, saliency-based visual token pruning methods exhibit notable limitations in complex vision-language tasks. First, they inevitably **compromise semantic completeness** by discarding key contextual information essential for comprehensive visual understanding. For example, in response to the question “Where is the cat?”, attention may focus primarily on the object “cat” while neglecting its surrounding context. The saliency-based methods typically concentrate on a small subset of visual tokens (see Fig. 1(a)), resulting in significant semantic loss. Moreover, saliency-based approaches often suffer from highly **skewed attention distribution**, where only a few tokens receive substantial attention while the rest exhibit nearly uniform (*i.e.*, flat) attention values as shown in Fig. 1(b). This hampers the discriminability among tokens, making it difficult to differentiate potentially informative ones from truly redundant ones.

To address the above challenges, we propose a novel visual token pruning strategy, named **Saliency-Coverage Oriented token Pruning for Efficient MLLMs (SCOPE)**, which jointly models the saliency and coverage of selected visual tokens to preserve semantic completeness. Specifically, we first define a set-coverage score for a selected token set based on token relationships and introduce a token-coverage gain for each unselected token, measuring the additional coverage achieved by including that token. We then propose a SCOPE score to integrate the token saliency score into the token-coverage gain, and iteratively select the token with the highest SCOPE score. This enables our method to retain tokens that not only contribute the most salient information but also ensure broad semantic coverage (see Fig. 1(a)).

To evaluate the effectiveness of our SCOPE, we conduct extensive experiments on a variety of vision-language understanding benchmarks using popular MLLMs, including LLaVA-1.5 [24] and LLaVA-Next [25]. The results demonstrate that our method consistently outperforms prior approaches by a significant margin (see Fig. 1(c)). For instance, SCOPE achieves a 9× reduction in the number of visual tokens while retaining 96.0% of the original performance on LLaVA-1.5 7B [24].

Our main contributions are summarized as follows:

- We reveal the limitation of the saliency-based visual token pruning methods, which unfortunately ignore the semantic completeness of the selected visual tokens and suffer from a highly skewed attention distribution problem.

- We propose a novel visual token pruning strategy, named **Saliency-Coverage Oriented token Pruning for Efficient MLLMs (SCOPE)**, which jointly models saliency and coverage of the retained visual tokens to preserve semantic completeness.
- We integrate SCOPE into representative MLLMs such as LLaVA-1.5 and LLaVA-Next without training, and demonstrate its effectiveness on multiple vision-language benchmarks, achieving a favorable trade-off between computational efficiency and task performance.

2 Related Work

Multimodal Large Language Models (MLLMs). Large Language Models (LLMs)[1, 37, 3, 10, 16] have achieved remarkable success in a wide range of language understanding and generation tasks. Building on this foundation, Multimodal LLMs (MLLMs)[24, 25, 21, 50, 19, 4] have shown impressive progress in visual understanding. A prevailing paradigm in MLLMs projects visual features into a sequence of visual tokens via a vision-to-language projector, and feeds them into the LLM alongside text tokens, as exemplified by LLaVA [24, 25], Qwen-VL [4], and Mini-Gemini [21].

However, real-world images are often high-resolution, resulting in long visual token sequences that significantly slow down inference in MLLMs [23, 30, 20, 9]. For example, LLaVA-Next [25] converts a 672×672 image into over 2,000 tokens. The situation worsens when handling multiple images or videos, further increasing the number of visual tokens. This highlights the need for effective strategies to reduce token length and accelerate vision-language inference.

Visual Token Pruning/Compression in MLLMs. A number of recent studies [49, 41, 40, 7] have focused on reducing visual token redundancy in MLLMs without requiring additional model training. Most of these methods [7, 49, 41] rely on specific attention scores to rank token saliency, such as text-to-vision attention in LLMs or CLS-token attention in vision transformers. They typically retain only the top-ranked tokens using a top- k strategy, *i.e.*, selecting tokens with the highest attention scores. For instance, FastV [7] leverages early-layer text-to-vision attention to retain salient tokens. SparseVLM [49] uses important textual words as a rater to guide token selection. VisionZip [43] applies CLS-based attention in the vision transformer for token pruning. To further increase the information density of the selected tokens, several approaches attempt to merge semantically similar tokens [43, 49, 35]. DivPrune [2] selects visual tokens by maximizing the diversity of selected tokens. In contrast, our method jointly considers both saliency and coverage, aiming to preserve semantic completeness while reducing token redundancy.

3 Method

In this section, we first introduce the preliminaries of visual token pruning and discuss the instantiation of saliency-based pruning methods in Sec. 3.1. In Sec.3.2, we provide a coverage analysis and show that saliency-based methods often suffer from low coverage. Finally, we present our proposed **Saliency-Coverage Oriented token Pruning for Efficient MLLMs (SCOPE)** in Sec.3.3.

3.1 Preliminary

Visual Token Pruning. The core architecture of LLMs consists of stacked self-attention layers and feed-forward networks (FFNs)[38], where the computational complexity grows quadratically with the input sequence length. In MLLMs, input images are typically high-resolution, resulting in long sequences of visual tokens. For instance, LLaVA[26] produces 576 visual tokens for a single image, which is often significantly longer than the corresponding text input in many visual understanding tasks. Furthermore, visual tokens often exhibit substantial redundancy [7, 41] due to repeated patterns and limited informational content in background regions.

Therefore, reducing the number of visual tokens is essential for enhancing the computational efficiency of MLLMs. In particular, $\mathcal{V} = \{v_1, \dots, v_N\}$ denotes the full set of N visual tokens extracted from the image, where each token $v_i \in \mathbb{R}^d$ represents a local region of the image. The goal of visual token pruning algorithm $\mathcal{A}$ is to select a small subset of visual tokens $\mathcal{S} = \{v_1, \dots, v_K\} = \mathcal{A}(\mathcal{V})$, where $K \ll N$. The objective of visual token pruning is to ensure that the model’s output based on $\mathcal{S}$ closely approximates the output based on the full set $\mathcal{V}$. Formally, the pruning objective can be

formulated as:

$$\min_{\mathcal{S}} \mathcal{L}(\mathcal{M}(\mathcal{S}, T), \mathcal{M}(\mathcal{V}, T)), \quad (1)$$

where $\mathcal{M}(\cdot, T)$ denotes the output of the vision-language model given visual input (either $\mathcal{V}$ or $\mathcal{S}$) and text input T , and $\mathcal{L}$ is a function to measure the output difference of LLM.

Saliency-based Visual Token Pruning. The saliency-based visual token pruning methods aim to reduce token redundancy by retaining the most salient visual tokens while discarding the less informative ones. The core challenge lies in how to effectively measure the saliency of each visual token. Several prior works [7, 41, 49, 43] estimate saliency by leveraging attention scores. Specifically, the attention matrix $\mathbf{A}$ is calculated as:

$$\mathbf{A} = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right), \quad (2)$$

where d is the embedding dimension, $\mathbf{Q}$ and $\mathbf{K}$ is the query and key matrices in the standard attention mechanism. These attention scores indicate the interaction strength between tokens, guiding the identification of highly salient tokens. In practice, in the vision encoder of CLIP [34], the [CLS] token is used to aggregate global information from the entire image. Therefore, the attention scores from the [CLS] token to the visual tokens serve as a reasonable proxy for token saliency. Based on these saliency scores, token pruning methods typically adopt a top-k selection strategy to retain only the most salient visual tokens. This approach effectively reduces visual token redundancy and significantly accelerates MLLM inference across various tasks.

3.2 Coverage Analysis

Although saliency-based pruning methods can effectively identify important tokens based on attention scores, they inevitably discard certain semantically critical tokens that are essential for comprehensive visual understanding. Semantic completeness, however, is crucial for accurately responding to a wide range of instruction prompts in MLLMs. Furthermore, saliency-based approaches often suffer from highly skewed attention distributions, where a small subset of tokens receives disproportionately high attention, while the remaining tokens exhibit nearly uniform (i.e., flat) attention values. This skewness undermines token discriminability, making it challenging to distinguish between potentially informative tokens and truly redundant ones. To quantitatively assess the representational completeness of the selected tokens, we introduce the notion of the θ -coverage (see **Definition 1**), which measures the degree to which the retained tokens cover the semantic space of the full token set.

Definition 1 (θ -Coverage). Let $\mathcal{V} = \{v_i \in \mathbb{R}^d \mid i = 1, \dots, n\}$ denote the full set of tokens extracted from an input image, and let $\mathcal{V}' \subseteq \mathcal{V}$ be a subset of selected tokens. For a given similarity threshold $\theta \in [0, 1]$, we say that a token $v \in \mathcal{V}$ is covered by $\mathcal{V}'$ if there exists at least one token $v' \in \mathcal{V}'$ such that their cosine similarity satisfies:

$$\text{sim}(v, v') := \frac{v^\top v'}{\|v\| \cdot \|v'\|} \geq \theta. \quad (3)$$

The θ -coverage of $\mathcal{V}'$ over $\mathcal{V}$ is then defined as the proportion of tokens in $\mathcal{V}$ that are covered by $\mathcal{V}'$:

$$\text{Coverage}_\theta(\mathcal{V}', \mathcal{V}) = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \mathbb{I}\left(\max_{v' \in \mathcal{V}'} \text{sim}(v, v') \geq \theta\right), \quad (4)$$

where $\mathbb{I}(\cdot)$ is the indicator function, which equals 1 if the condition holds and 0 otherwise.

This definition provides a semantic-aware metric to quantify how well the selected tokens set $\mathcal{V}'$ represents the full set. A higher value of θ imposes a stricter similarity criterion, typically leading to lower coverage but ensuring that the retained tokens are more semantically representative.

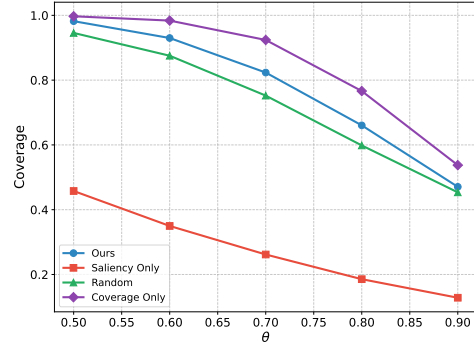


Figure 2: Comparison of θ -coverage across different token pruning criteria. The experiments are conducted on the MME benchmark, with 64 tokens selected out of the original 576 in LLaVA 1.5 7B.

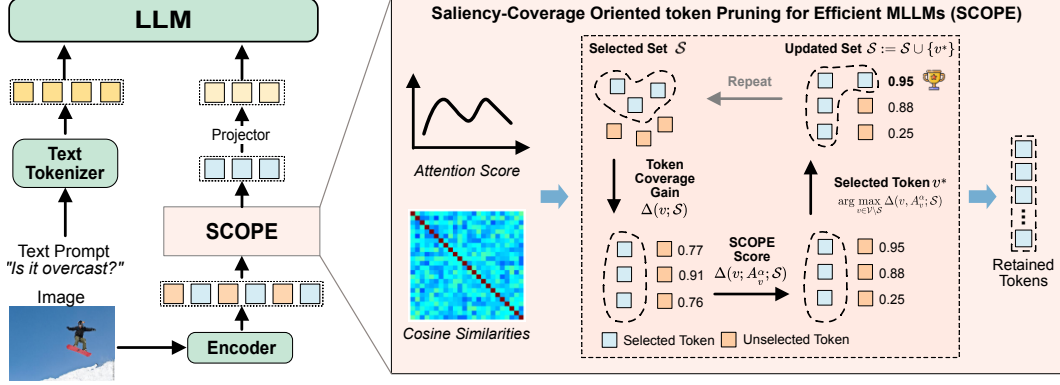


Figure 3: An overview of the proposed visual token pruning framework. The left part illustrates how our method reduces the number of visual tokens before feeding them into the LLM, thereby accelerating inference in MLLMs without requiring additional model training. The right part provides a detailed view of our SCOPE method, which jointly optimizes saliency and coverage to select a compact yet semantically representative subset of visual tokens.

In particular, we present the θ -coverage results on the MME benchmark in Fig. 2. The Saliency Only method selects dominant tokens solely based on the attention scores from the CLS token. However, it consistently exhibits low coverage across different values of θ , even performing worse than the random selection baseline. This observation suggests that although the saliency-based method captures dominant information, it tends to overlook a substantial amount of semantic content. In contrast, our method (detailed in Sec.3.3) incorporates saliency scores into a coverage-aware selection framework, striking a better balance between saliency and semantic coverage. As a result, it achieves significantly higher coverage compared to the Saliency Only method.

3.3 Saliency-Coverage Oriented Token Pruning

In contrast to saliency-based pruning methods, our goal is to jointly optimize saliency and coverage in the visual token selection process. This enables the pruning algorithm to not only preserve the most informative tokens but also maximize the semantic coverage of the selected subset. As a result, the retained tokens are both highly informative and semantically diverse, thereby maintaining semantic completeness under a constrained token budget, which is an essential property for comprehensive visual understanding across a wide range of multimodal tasks.

In the following, we first define the notion of coverage for selected tokens. Next, we introduce the concept of token-coverage gain, *i.e.*, the additional coverage obtained by including a new token in the selected set [14]. Finally, we incorporate the saliency score into the token-coverage gain formulation to balance both selection criteria. The overview of the proposed method is presented in Fig. 3.

Set-coverage for selected tokens. To quantify semantic coverage, we measure the similarity between token vectors using cosine similarity. We first define the individual coverage score $C(u, \mathcal{S})$ for a token $u \in \mathcal{V}$ by a set of selected tokens $\mathcal{S} \subseteq \mathcal{V}$ as:

$$C(u, \mathcal{S}) = \max_{s \in \mathcal{S}} \text{sim}(u, s) \quad (5)$$

where $\text{sim}(u, s)$ is the cosine similarity metric between token u and token s . The overall coverage of the selected subset $\mathcal{S}$ is defined as the sum of the maximum similarities between each token in the full set $\mathcal{V}$ and its most similar token in $\mathcal{S}$:

$$f(\mathcal{S}) = \sum_{u \in \mathcal{V}} C(u, \mathcal{S}) = \sum_{u \in \mathcal{V}} \max_{s \in \mathcal{S}} \text{sim}(u, s) \quad (6)$$

This formulation encourages the selection of tokens that are semantically diverse and broadly representative of the input space. Intuitively, it ensures that each token in the full set has at least one similar counterpart in the selected subset, thus preserving information while reducing the token count.

Token-coverage Gain. To quantify the contribution of each candidate token $v \in \mathcal{V} \setminus \mathcal{S}$, we evaluate its *marginal gain* with respect to the current subset $\mathcal{S}$ [14]. The marginal gain is defined as the

Algorithm 1 SCOPE

Require: A full set of tokens $\mathcal{V} = \{v_1, \dots, v_n\} \subset \mathbb{R}^d$, number of retained token K , pairwise similarities $S_{uv} = \text{sim}(u, v)$ for all $u, v \in \mathcal{V}$, attention score A_v for each token v , and a scaling factor α .

Ensure:

Selected token subset $\mathcal{S} \subseteq \mathcal{V}$ with $|\mathcal{S}| = K$

- 1: Initialize $\mathcal{S} \leftarrow \emptyset$ ▷ Start with an empty selected subset
- 2: Initialize coverage scores: $c_u \leftarrow 0$ for all $u \in \mathcal{V}$ ▷ c_u tracks the best similarity between u and any selected token so far
- 3: **for** $t = 1$ to K **do**
- 4: **for all** $v \in \mathcal{V} \setminus \mathcal{S}$ **do**
- 5: Compute marginal gain: $\Delta(v; \mathcal{S}) = \sum_{u \in \mathcal{V}} [\max(S_{uv}, c_u) - c_u]$ ▷ Compute the additional coverage that token v brings if added to $\mathcal{S}$
- 6: **end for**
- 7: Select next token: $v^* \in \arg \max_{v \in \mathcal{V} \setminus \mathcal{S}} \Delta(v; \mathcal{S}) \cdot A_v^\alpha$ ▷ Choose the token that balances coverage and saliency score
- 8: Update selected subset: $\mathcal{S} \leftarrow \mathcal{S} \cup \{v^*\}$ ▷ Add the selected token to the subset
- 9: Update coverage scores: $c_u \leftarrow \max(c_u, S_{uv^*}) \quad \forall u \in \mathcal{V}$ ▷ Update coverage scores using the newly added token
- 10: **end for**
- 11: **return** $\mathcal{S}$

increase in total coverage achieved by including v , and can be formally expressed as follows:

$$\Delta(v; \mathcal{S}) = f(\mathcal{S} \cup \{v\}) - f(\mathcal{S}), \quad (7)$$

Expanding this definition using Eq. (6), we can express the marginal gain as the sum of the new coverage provided by v to each token u that was not already fully covered by $\mathcal{S}$:

$$\begin{aligned} \Delta(v; \mathcal{S}) &= \sum_{u \in \mathcal{V}} \max_{s \in (\mathcal{S} \cup \{v\})} \text{sim}(u, s) - \sum_{u \in \mathcal{V}} \max_{s \in \mathcal{S}} \text{sim}(u, s) \\ &= \sum_{u \in \mathcal{V}} (\max(C(u, \mathcal{S}), \text{sim}(u, v)) - C(u, \mathcal{S})) \end{aligned} \quad (8)$$

This quantifies how much additional coverage is achieved by selecting token v , taking into account its ability to represent other tokens $u \in \mathcal{V}$ that are not yet well-represented by the current subset $\mathcal{S}$.

SCOPE score. While the token-coverage gain considers only the geometric coverage in semantic space, it overlooks the intrinsic information carried by individual tokens. To address this limitation, we propose the SCOPE gain, which incorporates token saliency into the coverage gain to better preserve visual token information. Specifically, we integrate the visual attention score into the coverage gain function as follows:

$$\Delta(v, A_v^\alpha; \mathcal{S}) = \Delta(v; \mathcal{S}) \cdot A_v^\alpha, \quad (9)$$

where A_v^α denotes the attention score of the visual token v , and α is a scaling factor. The token v^* with the highest SCOPE gain is selected and added to the subset $\mathcal{S}$:

$$v^* \in \arg \max_{v \in \mathcal{V} \setminus \mathcal{S}} \Delta(v, A_v^\alpha; \mathcal{S}) \quad (10)$$

This process is iteratively repeated until the desired subset size is reached. The pseudocode of the proposed pruning method is presented in Algorithm 1.

Integration into MLLMs. The proposed method is applicable to a wide range of MLLMs. In this work, we apply it to the widely adopted LLaVA[26] and LLaVA-Next [25] models, following prior studies [7, 49, 41]. Our method is integrated after the vision encoder to maximize information retention post token pruning. This enables the language model to receive more complete visual signals, thereby supporting comprehensive visual understanding without compromising performance. Our method is train-free and significantly accelerates the inference of MLLMs with minimal performance degradation. For example, our approach preserves over 96% of the original model’s performance while reducing the number of visual tokens by a factor of 8 in LLaVA 1.5 7B.

Table 1: **Performance comparison under different vision token configurations.** We evaluate the LLaVA 1.5 7B model, where the default number of visual tokens is 576. The first row for each method reports the raw accuracy across benchmarks, and the second row indicates the performance relative to the upper bound. [†] denotes the results adapted from [49].

Method	GQA	MMB	MME	POPE	SQA	TextVQA	SEED	MMVet	Avg.
Upper Bound, 576 Tokens (100%)									
Vanilla (CVPR'24)	61.9 100%	64.7 100%	1862 100%	85.9 100%	69.5 100%	58.2 100%	58.6 100%	31.1 100%	100%
Retain 192 Tokens ($\downarrow$ 66.7%)									
FastV (ECCV'24)	52.7 85.1%	61.2 94.6%	1612 86.6%	64.8 75.4%	67.3 96.8%	52.5 90.2%	57.1 97.4%	27.7 89.7%	89.5%
SparseVLM (ICML'25)	57.6 93.1%	62.5 96.6%	1721 92.4%	83.6 97.3%	69.1 99.4%	56.1 96.4%	55.8 95.2%	31.5 101.3%	96.5%
VisionZip (CVPR'25)	59.3 95.8%	63.0 97.4%	1783 95.7%	85.3 99.3%	68.9 99.1%	57.3 98.5%	56.4 96.2%	31.7 101.9%	98.0%
PDrop (CVPR'25) [†]	57.1 92.2%	63.2 97.7%	1766 94.8%	82.3 95.8%	70.2 101.0%	56.1 96.4%	54.7 93.3%	30.5 98.1%	96.2%
Ours	60.1 97.1%	63.6 98.3%	1804 96.9%	86.4 100.6%	68.8 99.0%	57.7 99.1%	58.7 100.2%	32.5 104.5%	99.5% ($\downarrow$ 0.5%)
Retain 128 Tokens ($\downarrow$ 77.8%)									
FastV (ECCV'24)	49.6 80.1%	56.1 86.7%	1490 80.0%	59.6 69.4%	60.2 86.6%	50.6 86.9%	55.9 95.4%	28.1 90.4%	84.4%
SparseVLM (ICML'25)	56.0 90.5%	60.0 92.7%	1696 91.1%	80.5 93.7%	67.1 96.5%	54.9 94.3%	53.4 91.1%	30.0 96.5%	93.3%
VisionZip (CVPR'25)	57.6 93.1%	62.0 95.8%	1761.7 94.6%	83.2 96.9%	68.9 99.1%	56.8 97.6%	54.9 93.7%	32.6 104.8%	96.9%
PDrop (CVPR'25) [†]	56 90.5%	61.1 94.4%	1664 89.4%	82.3 95.8%	69.9 100.6%	55.1 94.7%	53.3 91.0%	30.8 99.0%	94.4%
Ours	59.7 96.4%	62.5 96.6%	1776 95.4%	86.1 100.2%	68.4 98.4%	57.2 98.3%	57.8 98.6%	31.4 101.0%	98.1% ($\downarrow$ 1.9%)
Retain 64 Tokens ($\downarrow$ 88.9%)									
FastV (ECCV'24)	46.1 74.5%	48.0 74.2%	1256 67.5%	48 55.9%	51.1 73.5%	47.8 82.1%	51.9 88.6%	25.8 83.0%	74.9%
SparseVLM (ICML'25)	52.7 85.1%	56.2 86.9%	1505 80.8%	75.1 87.4%	62.2 89.5%	51.8 89.0%	51.1 87.2%	23.3 74.9%	85.1%
VisionZip (CVPR'25)	55.1 89.0%	60.1 92.9%	1690 90.8%	77.0 89.6%	69.0 99.3%	55.5 95.4%	52.2 89.1%	31.7 101.9%	93.5%
PDrop (CVPR'25) [†]	41.9 67.7%	33.3 51.5%	1092 58.6%	55.9 65.1%	69.2 99.6%	45.9 78.9%	40.0 68.3%	30.7 98.7%	73.5%
Ours	58.3 94.2%	61.7 95.4%	1698 91.2%	83.9 97.7%	68.6 98.7%	56.6 97.3%	56.3 96.1%	30.4 97.7%	96.0% ($\downarrow$ 4.0%)

4 Experiment

4.1 Experiments Setup

Evaluation Benchmarks and Baselines. Following prior work[49], we evaluate the effectiveness of the proposed method using a set of widely adopted multimodal benchmarks. Specifically, these include GQA [13], MMBench[27], POPE [22], ScienceQA[29], TextVQA [36], SEEDBench[18], and MMVet [45]. We also compare against several state-of-the-art baselines, including FastV[7], SparseVLM [49], VisionZip[43], and PDrop [41]. For the video benchmarks, we evaluate the MLLMs on the benchmarks TGIF [15], MSVD [5], MSRVT [42], and ActivityNet [46]. For further details on evaluation benchmarks and metrics, we refer the reader to the Appendix B.

Implementation Details. We integrate the proposed method into LLaVA 1.5 [26] and LLaVA-Next [25] for image understanding and Video-LLaVA [23] for video understanding. The pruning module is inserted after the vision encoder. The saliency score is derived from the attention weights of visual tokens with respect to the CLS token at the second-to-last layer (layer -2) of the vision encoder. The scaling factor α is set to 1.0 by default. Our implementation is based on the Imms-evals [48] package. We conduct the experiments on $4 \times A100$ GPUs. The inference batch size is set to 1 for all the evaluation results.

Table 2: **Performance comparison under different vision token configurations.** The evaluated model is LLaVA-Next 7B. The vanilla number of vision tokens is 2,880. The first line of each method is the raw accuracy of benchmarks, and the second line is the proportion relative to the upper bound.

Method	GQA	MMB	MME	SQA	TextVQA	MMMU	Avg.
Upper Bound, 2880 Tokens (100%)							
Vanilla (CVPR’24)	64.2 100%	67.9 100%	1842 100%	70.2 100%	61.3 100%	35.1 100%	100%
Retain 640 Tokens ($\downarrow$ 77.8%)							
SparseVLM (ICML’25)	60.3 93.9%	65.7 96.8%	1772 96.2%	67.7 96.4%	57.8 94.3%	34.6 98.6%	96.0%
VisionZip (CVPR’25)	61.3 95.5%	66.3 97.6%	1787 97.0%	68.1 97.0%	60.2 98.2%	34.7 98.9%	97.4%
Ours	61.9 96.4%	66.2 97.5%	1842 100.0%	67.8 96.6%	60.1 98.0%	36.9 105.1%	98.9% ($\downarrow$ 1.1%)
Retain 320 Tokens ($\downarrow$ 88.9%)							
SparseVLM (ICML’25)	57.7 89.9%	64.3 94.7%	1694 92.0%	67.3 95.9%	55.9 91.2%	34.4 98.0%	93.6%
VisionZip (CVPR’25)	59.3 92.4%	63.1 92.9%	1702 92.4%	67.3 95.9%	58.9 96.1%	35.3 100.6%	95.0%
Ours	61.0 95.0%	65.9 97.1%	1789 97.1%	67.7 96.4%	58.4 95.3%	35.6 101.4%	97.1% ($\downarrow$ 2.9%)
Retain 160 Tokens ($\downarrow$ 94.4%)							
SparseVLM (ICML’25)	51.2 79.8%	63.1 92.9%	1542 83.7%	67.5 96.2%	46.4 75.7%	32.8 93.4%	86.9%
VisionZip (CVPR’25)	55.5 86.4%	60.1 88.5%	1630 88.5%	68.3 97.3%	56.2 91.7%	36.1 102.8%	92.5%
Ours	60.0 93.5%	64.3 94.7%	1700 92.3%	67.4 96.0%	56.8 92.7%	35.6 101.4%	95.1% ($\downarrow$ 4.9%)

4.2 Main Results

Results on LLaVA 1.5. LLaVA 1.5 is one of the most representative MLLMs. We therefore apply the proposed pruning method to LLaVA 1.5 and evaluate its performance on a variety of image understanding tasks, following prior works [41, 49, 43]. Due to the diverse evaluation metrics used across different benchmarks, which result in inconsistent numerical scales, we report performance as a percentage of the original model’s accuracy. We show the results of LLaVA 1.5 7B in Table 1. In particular, we follow previous work [49, 43] and evaluate the performance under three visual token pruning budgets (*i.e.*, 192, 128, and 64) to evaluate the effectiveness of the proposed method. The vanilla model (*i.e.*, LLaVA 1.5 7B with full visual tokens) serves as the upper bound (100%), representing the performance ceiling of any visual token pruning approach. Our method consistently outperforms existing approaches across all token configurations, particularly under aggressive compression settings. As shown in Table 1, when retaining only 192 tokens (a 66.7% reduction from the baseline), our method achieves an average accuracy of 99.5% relative to the upper bound. This surpasses state-of-the-art baselines including FastV [7] (+6.0%), SparseVLM [49] (+3.0%), and VisionZip [43] (+1.5%). Under extreme compression (*e.g.*, 64 tokens, 88.9% reduction), our method maintains 96.0% of the original performance, significantly outperforming baselines such as VisionZip [43] (93.5%) and SparseVLM [49] (85.1%).

Surprisingly, our method preserves or even surpasses the upper bound in performance on several benchmarks. For instance, we observe relative accuracies of 100.2% and 104.5% on POPE [22] and MMVet [45], respectively, when using 192 tokens. These results suggest that visual tokens in MLLMs contain redundancy, and our method not only reduces this redundancy but also improves performance by eliminating interference from redundant information. We further evaluate our method on the larger LLaVA 1.5 13B model to validate its generalization capability in Appendix C.1.

Results on LLaVA-Next. Compared to LLaVA 1.5, LLaVA-Next is a more advanced MLLM that supports high-resolution image processing, thereby significantly improving vision-language understanding. LLaVA-Next partitions an input image into multiple regions based on its original size. Usually, the image is divided into 4 sub-images. Both the original and partitioned images are then encoded into visual tokens, resulting in a total of 2,880 tokens (576×5). While effective in capturing fine-grained visual details, this strategy substantially increases the number of visual

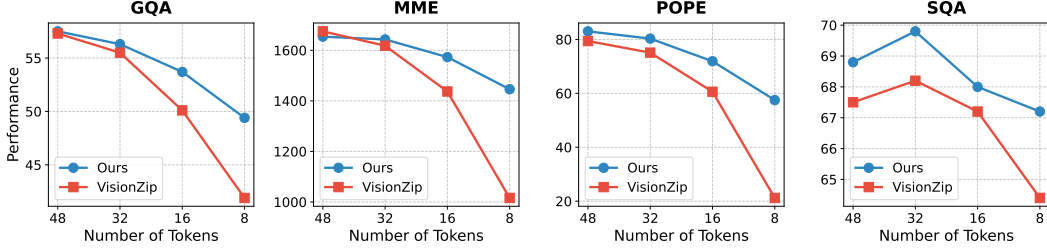


Figure 4: The performance comparison under extreme token number.

tokens and reduces inference efficiency. Therefore, our objective is to minimize the number of visual tokens while maintaining performance as much as possible. To evaluate the proposed method on LLaVA-Next, we follow previous works [49, 43] and adopt three visual token budget settings (*i.e.*, 640, 320, and 160). The results are presented in Table 2. As shown, our method consistently outperforms state-of-the-art approaches under all configurations. Specifically, when retaining only 640 tokens, our approach achieves an average accuracy of 98.9% relative to the upper bound. Under extreme compression (*e.g.*, 160 tokens, 94.4% reduction), our method maintains 95.1% performance, significantly surpassing baselines such as SparseVLM (86.9%) and VisionZip (92.5%). These results further validate the effectiveness of the proposed method across different MLLM architectures. We also evaluate our method on the LLaVA-Next 13B model in Appendix C.1.

Results on Video benchmarks. We further evaluate the effectiveness of the proposed method, and we implement our SCOPE based on Video-LLaVA following VisionZip [43]. The results are reported in Table 3. As shown, our method achieves the best performance among all compared methods. Surprisingly, even with aggressive pruning, our method almost fully preserves the original performance. This demonstrates the strong effectiveness of our method on video-language tasks. These findings also suggest that video benchmarks contain substantial redundancy, and token pruning has great potential for accelerating video LLMs without sacrificing performance.

Table 3: Performance comparison on Video-LLaVA. The original Video-LLaVA’s video token number is 2048, while our method only retains the 136 tokens.

Method	TGIF	MSVD	MSRVTT	ActivityNet	Avg
Video-LLaVA	47.1	69.8	56.7	43.1	100.0%
FastV	23.1	38.0	19.3	30.6	52.1%
	49.0%	54.4%	34.0%	71.0%	
SparseVLM	44.7	68.2	31.0	42.6	86.5%
	94.9%	97.7%	54.7%	98.8%	
VisionZip	42.4	63.5	52.1	43.0	93.2%
	90.0%	91.0%	91.9%	99.8%	
Ours	47.1	69.2	55.9	44.9	100.5%
	100.0%	99.1%	98.6%	104.2%	

4.3 Analysis

Results under Extreme Token Reduction. Our method demonstrates superior performance stability as the number of visual tokens is progressively reduced. As shown in Fig. 4, even when the token count is reduced to as few as 8, our approach consistently outperforms VisionZip [43] by increasingly larger margins. This highlights the strong capability of our framework to retain critical visual information under extreme compression. In contrast, VisionZip exhibits a sharp performance drop in low-token regimes, further validating the effectiveness of our token selection strategy and underscoring the potential of our method for aggressive visual token pruning.

Ablation Studies. As shown in Table 4, our method, which jointly considers token saliency and coverage, consistently outperforms its ablated variants (saliency-only and coverage-only) across all benchmarks. Both ablated models still perform better than the random baseline, indicating the individual effectiveness of each component. For instance, the coverage-only variant achieves moderate performance. However, our full method further improves these results, demonstrating that combining saliency and coverage provides complementary benefits. Explicit modeling of both saliency and coverage leads to superior performance compared to using either criterion alone or selecting tokens randomly.

Table 4: Ablation studies of the proposed method.

	GQA	MMB	MME	POPE	TextVQA
Random	55.5	54.0	1556	75.2	48.4
Saliency-only	55.0	60.8	1665	76.8	55.4
Coverage-only	58.1	60.8	1687	82.1	56.3
Ours	58.3	61.7	1698	83.9	56.6

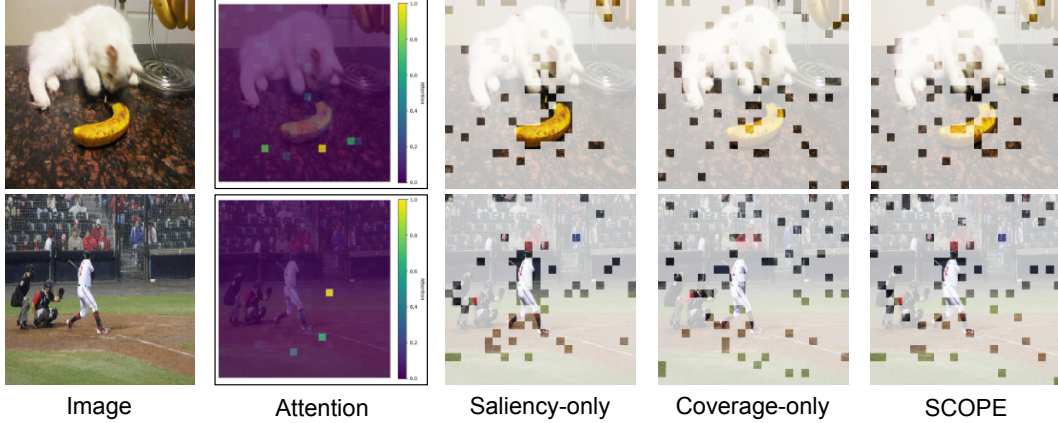


Figure 5: Visualization of token pruning among different pruning strategies.

Efficiency Analysis. Table 5 compares the efficiency of our method with that of a baseline pruning approach (PDrop) on LLaVA-NeXT 7B. Despite reducing the number of visual tokens from 2,880 to 160, a compression ratio exceeding $18\times$, our method maintains strong performance on the POPE

Table 5: Efficiency analysis of our method on LLaVA-NeXT 7B. The experiments are conducted on a system equipped with $4\times A100$. Δ denotes the reduction ratio.

	Token Number	POPE	Latency (s)	Δ
Vanilla	2880	86.4	601.9	-
PDrop	160	53.2	184.0	$3.3\times$
Ours	160	81.3	188.8	$3.2\times$

metric (81.3% vs. 86.4%), demonstrating that our token selection strategy effectively preserves semantic completeness. In contrast, PDrop [41] exhibits a substantial performance drop (53.2%), likely due to its reliance on saliency-based pruning, which may overlook less attended yet semantically important regions. Although our method incurs slightly higher latency than PDrop, it still achieves a $3.2\times$ speedup over the full-token baseline. This indicates that our saliency-coverage oriented pruning strategy is not only effective in preserving performance but also computationally efficient in practice.

Token Pruning Visualization. In Fig. 5, we provide a visualization of token pruning to illustrate the difference of selected tokens among different strategies. Saliency-only mainly concentrates on the most salient patch such as the cat and banana in 1st row, demonstrating object-level focus by pruning the background. Coverage-only selects the tokens that are spread across the image, preserving global context but potentially missing important object details. Our SCOPE maintains the high token density on salient patches (*e.g.*, cat and banana in 1st row), while a sparse set of tokens is strategically kept for the background. This captures critical object features without discarding essential scene context.

5 Conclusion

While existing approaches predominantly rely on attention-based saliency to prune redundant tokens, they often neglect semantic coverage, leading to incomplete visual representations. To overcome this limitation, we propose SCOPE, a novel visual token pruning framework that jointly models both token saliency and coverage. Our method introduces a set-coverage score based on pairwise token similarities and calculates a token-coverage gain for each candidate token. By incorporating saliency scores into this gain, we derive the SCOPE score, which guides an iterative token selection process. Empirical evaluations on LLaVA 1.5 and LLaVA-Next across multiple vision-language benchmarks show that SCOPE consistently outperforms state-of-the-art pruning approaches, achieving strong performance even under aggressive token reduction. We believe that our approach offers a principled and effective framework for evaluating the value of visual tokens in MLLMs.

Acknowledgement

This work was supported by National Natural Science Foundation of China (No.62476051, No.62176047, No.82121003) and Sichuan Natural Science Foundation (No.2024NSFTD0041). This research is supported in part by the National Research Foundation, Singapore under its National Large Language Models Funding Initiative (AISG Award No: AISG-NMLP-2024-003), in part by A*STAR Career Development Fund (CDF) under C243512011. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv:2303.08774*, 2023.
- [2] 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.
- [3] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv:2309.16609*, 2023.
- [4] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A frontier large vision-language model with versatile abilities. *arXiv:2308.12966*, 2023.
- [5] David Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 190–200, 2011.
- [6] Fei-Long Chen, Du-Zhen Zhang, Ming-Lun Han, Xiu-Yi Chen, Jing Shi, Shuang Xu, and Bo Xu. Vlp: A survey on vision-language pre-training. *Machine Intelligence Research*, 20(1):38–56, 2023.
- [7] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models, 2024.
- [8] Lin Chen, Jisong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. *arXiv:2311.12793*, 2023.
- [9] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv:2312.14238*, 2023.
- [10] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. See <https://vicuna.lmsys.org> (accessed 14 April 2023), 2(3):6, 2023.
- [11] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [12] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. MME: A comprehensive evaluation benchmark for multimodal large language models. *arXiv:2306.13394*, 2023.
- [13] Drew A Hudson and Christopher D Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [14] Rishabh Iyer, Ninad Khargoankar, Jeff Bilmes, and Himanshu Asanani. Submodular combinatorial information measures with applications in machine learning. In *Algorithmic Learning Theory*, pages 722–754. PMLR, 2021.
- [15] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2758–2766, 2017.

- [16] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2. Minneapolis, Minnesota, 2019.
- [17] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.
- [18] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023.
- [19] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, 2023.
- [20] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, pages 323–340. Springer, 2024.
- [21] Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv:2403.18814*, 2024.
- [22] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv:2305.10355*, 2023.
- [23] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv:2311.10122*, 2023.
- [24] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv:2310.03744*, 2023.
- [25] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024.
- [26] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 2024.
- [27] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. MMBench: Is your multi-modal model an all-around player? *arXiv:2307.06281*, 2023.
- [28] Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12):220102, 2024.
- [29] 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.
- [30] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. 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 (ACL 2024)*, 2024.
- [31] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022.
- [32] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209, 2021.
- [33] Guanqiao Qu, Qiyuan Chen, Wei Wei, Zheng Lin, Xianhao Chen, and Kaibin Huang. Mobile edge intelligence for large language models: A contemporary survey. *arXiv preprint arXiv:2407.18921*, 2024.
- [34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

- [35] Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. *arXiv preprint arXiv:2403.15388*, 2024.
- [36] Amanpreet Singh, Vivek Natarjan, Meet Shah, Yu Jiang, Xinlei Chen, Devi Parikh, and Marcus Rohrbach. Towards VQA models that can read. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8317–8326, 2019.
- [37] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv:2302.13971*, 2023.
- [38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [39] 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.
- [40] Yuxin Wen, Qingqing Cao, Qichen Fu, Sachin Mehta, and Mahyar Najibi. Efficient vision-language models by summarizing visual tokens into compact registers. *arXiv preprint arXiv:2410.14072*, 2024.
- [41] 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.
- [42] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016.
- [43] 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. *arXiv preprint arXiv:2412.04467*, 2024.
- [44] Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*, 2024.
- [45] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. In *International conference on machine learning*. PMLR, 2024.
- [46] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *AAAI*, pages 9127–9134, 2019.
- [47] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhui Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of CVPR*, 2024.
- [48] Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, et al. Lmms-eval: Reality check on the evaluation of large multimodal models. *arXiv preprint arXiv:2407.12772*, 2024.
- [49] 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. *arXiv preprint arXiv:2410.04417*, 2024.
- [50] Qiang Zhou, Chaohui Yu, Shaofeng Zhang, Sitong Wu, Zhibing Wang, and Fan Wang. Regionblip: A unified multi-modal pre-training framework for holistic and regional comprehension. *arXiv preprint arXiv:2308.02299*, 2023.
- [51] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigt-4: Enhancing vision-language understanding with advanced large language models. *arXiv:2304.10592*, 2023.

A Overview

This appendix provides detailed information on the experimental benchmarks, additional qualitative results, and visualizations that support the main claims of the paper. In Section B, we present comprehensive descriptions of the benchmark datasets. Section C includes supplementary experiments, such as results on larger models (LLaVA 1.5 13B and LLaVA-Next 13B), as well as a hyperparameter analysis. In Section D, we provide additional visualization studies to further illustrate the behavior of our method. Finally, we discuss the broader impact and limitations of our work.

B Benchmarks

We conduct the experiments on several widely used visual understanding benchmarks. In the following, we will give a detailed description of these benchmarks.

GQA. [13]. The GQA benchmark consists of three components: scene graphs, questions, and images. The image component includes raw images, their spatial features, and the features of all objects within the images. The questions in GQA are crafted to evaluate visual scene understanding and reasoning about various aspects of an image. Our method is evaluated on the subset of “testdev_balanced_instructions”, which includes 12,578 samples.

MMBench. [27]. MMBench is a comprehensive benchmark designed to evaluate the multi-modal capabilities of large language models, covering a wide range of tasks including visual question answering, image captioning, cross-modal retrieval, and creative generation. It provides a fine-grained assessment from perception to cognition, containing approximately 3,000 multiple-choice questions aggregated from diverse sources. The benchmark aims to measure whether a model is a true “all-around player” in multi-modal understanding and reasoning.

MME. [12]. The MME benchmark is a comprehensive evaluation suite carefully crafted to assess multiple facets of model performance. It comprises 14 distinct subtasks targeting both perceptual and cognitive capabilities of models. By employing manually curated instruction-answer pairs and succinct instruction formats, MME effectively reduces the risks of data leakage and ensures a fairer assessment of model abilities. We evaluate the performance on the dev split including 4,377 samples. The evaluation metric is the accuracy of the model’s answer.

POPE. [22] The POPE benchmark focuses on assessing object hallucination in models by presenting them with a set of targeted yes/no questions about object existence within images. This approach reframes the evaluation of hallucination, emphasizing the model’s ability to correctly identify whether certain objects are present. To quantitatively analyze performance across three distinct sampling methods, the benchmark utilizes metrics such as accuracy, recall, precision, and F1 score, offering a robust measure of the model’s susceptibility to hallucination. We evaluate the model’s performance on the test split, including 9,000 samples. The evaluation metric is the F1 score.

ScienceQA (SQA). [29] Encompassing a wide array of fields such as natural sciences, linguistics, and social sciences, SQA structures its questions through a hierarchical framework consisting of 26 topics, 127 categories, and 379 distinct skills. This benchmark is designed to rigorously test a model’s proficiency in multimodal comprehension, complex reasoning across multiple steps, and interpretability. By organizing questions first by subject area, then by specific category, and finally by the required skill, SQA ensures a thorough and nuanced assessment of scientific understanding across diverse domains. This layered organization enables a detailed evaluation of a model’s ability to handle a broad spectrum of scientific queries. The evaluation metric is the accuracy.

TextVQA. [36] TextVQA is designed to assess a model’s capability to interpret and reason over textual content embedded in images. This benchmark challenges models with visual question answering tasks that require both comprehension of image context and accurate reading of the text present within the images. To perform well, models must effectively integrate visual and textual cues, demonstrating robust understanding and reasoning skills related to text in complex visual environments. We evaluate the model’s performance on the test split, including 5,000 samples. The evaluation metric is exact match (EM).

SEEDBench [18] SEEDBench features a collection of 19,000 multiple-choice questions curated by human annotators. Covering 12 different evaluation dimensions, this benchmark examines models’

Table 6: **Performance comparison under different vision token configurations.** The evaluated model is LLaVA 1.5 13B, where the default number of visual tokens is 576. The first row for each method reports the raw accuracy across benchmarks, and the second row indicates the performance relative to the upper bound.

Method	GQA	MMB	MME	POPE	SQA	TextVQA	SEED-I	MMVet	Avg.
Upper Bound, 576 Tokens (100%)									
Vanilla (CVPR’24)	63.2 100%	67.7 100%	1818 100%	85.9 100%	72.8 100%	61.3 100%	66.9 100%	35.3 100%	100%
Retain 192 Tokens ($\downarrow$ 66.7%)									
VisionZip (CVPR’25)	59.1 93.5%	66.9 98.8%	1754 96.5%	85.1 99.1%	73.5 101.0%	59.5 97.1%	65.2 97.5%	37.5 106.20%	98.7%
Ours	59.7 94.5%	67.6 99.9%	1775 97.6%	86.7 100.9%	73.8 101.4%	60 97.9%	65.5 97.9%	39.4 111.6%	100.2%
Retain 128 Tokens ($\downarrow$ 77.8%)									
VisionZip (CVPR’25)	57.9 91.6%	66.7 98.5%	1743 95.9%	85.2 99.2%	74 101.6%	58.7 95.8%	63.8 95.4%	37.5 106.2%	97.0%
Ours	59.3 93.8%	67.2 99.3%	1735 95.4%	85.9 100.0%	73.9 101.5%	58.7 95.8%	64.8 96.9%	37.7 106.8%	98.7%
Retain 64 Tokens ($\downarrow$ 88.9%)									
VisionZip (CVPR’25)	56.2 88.9%	64.9 95.9%	1676 92.2%	76.0 88.5%	74.4 102.2%	57.4 93.3%	60.4 90.3%	33.9 96.0%	93.7%
Ours	58.7 92.9%	65.5 96.8%	1762 96.9%	83.0 96.6%	73.2 100.5%	58.3 95.1%	63.6 95.1%	35.7 101.1%	96.9%

capabilities in identifying patterns within both images and videos, taking into account spatial as well as temporal characteristics. The evaluation metric is the accuracy.

MMVet [45] The MMVet benchmark is constructed with the understanding that tackling complex tasks typically requires a generalist model to effectively combine multiple fundamental vision-language skills. MMVet identifies six essential vision-language capabilities and systematically evaluates sixteen specific combinations arising from these core abilities, thereby assessing the model’s proficiency in integrating diverse vision-language functions. We evaluate the model’s performance on the test split, including 218 samples. The score is evaluated by the GPT model.

C Additional Experiments

In the main paper, we present experiments on LLaVA 1.5 7B and LLaVA-Next 7B. To further demonstrate the generalizability of our method across model scales, we provide additional results on LLaVA 1.5 13B and LLaVA-Next 13B. We also provide the results on more MLLMs such as Qwen2-VL and more OCR-related benchmarks.

C.1 Results on LLaVA 1.5 13B

As shown in Table 6, our method consistently outperforms VisionZip [43] across all token budgets. With 192 tokens, our approach achieves 100.2% of the upper bound’s average performance, slightly higher than VisionZip [43] (98.7%). The advantage becomes more evident as the token count decreases: at 64 tokens, our method retains 96.9% performance, compared to VisionZip’s 93.7%. Notably, on benchmarks like MMVet [45] and POPE [22], our method even surpasses the original model’s performance. These results demonstrate that our joint saliency-coverage strategy better preserves essential information under aggressive token pruning.

C.2 Results on LLaVA-Next 13B

We present the results on LLaVA-Next 13B in Table 7. We can observe that our method consistently outperforms VisionZip [43] under all token budgets. For example, with 640 tokens, our approach

Table 7: **Performance comparison under different vision token configurations.** The evaluated model is LLaVA-Next 13B. The vanilla number of vision tokens is 2,880. The first line of each method is the raw accuracy on the benchmarks, and the second line is the proportion relative to the upper bound.

Method	GQA	MMB	MME	POPE	SQA	TextVQA	MMMU	SEED-I	Avg.
Upper Bound, 2880 Tokens (100%)									
Vanilla 13B (CVPR'24)	65.4	70.0	1901	86.2	73.5	64.3	36.2	71.9	100%
	100%	100%	100%	100%	100%	100%	100%	100%	
Vanilla 7B (CVPR'24)	64.2	67.9	1842	86.4	70.2	61.3	35.1	70.2	97.2%
	98.2%	97.0%	96.9%	100.2%	95.5%	95.3%	97.0%	97.6%	
Retain 640 Tokens (↓ 77.8%)									
VisionZip (CVPR'25)	63.0	68.6	1871	85.7	71.2	62.2	36.4	68.8	97.8%
	96.3%	98.0%	98.4%	99.4%	96.9%	96.7%	100.6%	95.7%	
Ours	63.6	69.3	1897	86.4	72.5	62.4	36.6	69.9	98.8%
	97.2%	99.0%	99.8%	100.2%	98.6%	97.0%	101.1%	97.2%	
Retain 320 Tokens (↓ 88.9%)									
VisionZip (CVPR'25)	60.7	67.2	1805	82.0	70.3	60.9	35.6	65.2	94.8%
	92.8%	96.0%	95.0%	95.1%	95.6%	94.7%	98.3%	90.7%	
Ours	63.0	67.7	1830	85.1	71.7	60.8	36.3	67.9	96.9%
	96.3%	96.7%	96.3%	98.7%	97.6%	94.6%	100.3%	94.4%	
Retain 160 Tokens (↓ 94.4%)									
VisionZip (CVPR'25)	57.8	64.9	1739	76.6	69.3	58.4	37.0	61.1	91.7%
	88.4%	92.7%	91.5%	88.9%	94.3%	90.8%	102.2%	85.0%	
Ours	61.4	66.9	1777	82.8	72.0	59.3	36.2	66.1	95.1%
	93.9%	95.6%	93.5%	96.1%	98.0%	92.2%	100.0%	91.9%	

achieves 98.8% of the upper bound’s average performance, compared to VisionZip’s 97.8%. As the token count decreases to 160, our method still retains 95.1% performance, while VisionZip drops to 91.7%. These results further confirm the superior robustness of our method under aggressive token pruning.

C.3 Results on Qwen2-VL

To further evaluate the generalization of the proposed SCOPE, we have also evaluated our method on the Qwen2-VL [39] model. The results are summarized in Table 8. As shown, our method achieves 94.6% of the full-model performance when retaining only 25% of the tokens. Furthermore, our method significantly outperforms prior approaches such as DivPrune [2], with a 3.7% improvement in average score under the 10.0% token ratio.

Table 8: **Results on Qwen2-VL.** The token ratio means the ratio of retained tokens.

Method	Token Ratio	GQA	MMB	MME	POPE	Avg.
Qwen2-VL 7B	100%	61.9	77.4	2286	88.4	100%
DivPrune	25%	59.4	72	2043	85.9	93.90%
Ours	25%	59.8	72.5	2065	86.5	94.6%(+0.7%)
DivPrune	10%	54.3	63.7	1874	80.8	85.90%
Ours	10%	56.6	66.8	1953	84.3	89.6%(+3.7%)

C.4 Results on more OCR Benchmarks.

In the main paper, we have already evaluated our method on several OCR-related benchmarks, such as MME [12] and MMVet [45]. To further demonstrate the effectiveness of SCOPE, we conducted additional experiments on more OCR-specific benchmarks including DocVQA [32], ChartQA [31]

and OCRBench [28]. The results are shown in Table 9. As illustrated, our method consistently preserves performance and outperforms VisionZip across different token counts. This further supports the robustness of our approach for OCR tasks.

Table 9: **Results on more OCR Benchmarks.** The model is LLaVA 1.5 7B.

Method	#Token	DocVQA	ChartQA	OCRBench	Avg.
Vanilla	576	28.0	18.2	31.3	100%
VisionZip	192	26.0	17.3	31.1	95.8%
Ours	192	26.5	17.4	31.2	96.6% (+0.8%)
VisionZip	128	25.1	17.1	30.0	93.1%
Ours	128	25.9	17.3	30.7	95.2% (+2.1%)
VisionZip	64	21.1	16.0	28.2	84.4%
Ours	64	23.2	16.7	29.5	89.6%(+5.2%)

C.5 Hyper-parameter Analysis

The hyperparameter α controls the scaling of the attention scores, thereby influencing token selection in our method. As illustrated in Fig. 6, the optimal performance is typically achieved when $\alpha = 1.0$, suggesting that this setting effectively balances saliency and coverage across most benchmarks.

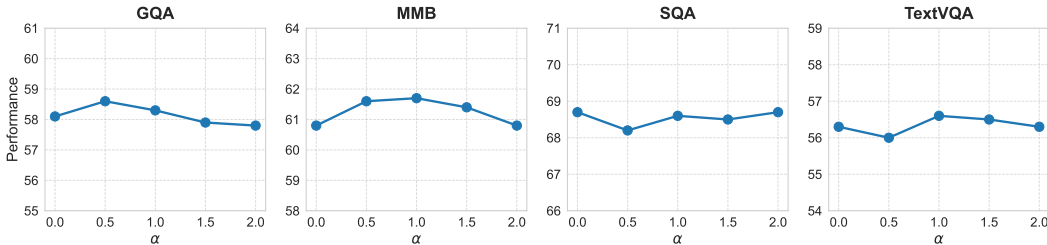


Figure 6: The hyperparameter α analysis on LLaVA 1.5 7B with 64 visual tokens.

D Visualization Results

We present additional results on selected token visualization in Fig. 7. The saliency-based method selects tokens solely based on attention scores, which may overlook semantically important tokens that contribute to the overall completeness of the visual representation. In contrast, our saliency-coverage oriented approach jointly considers both visual saliency and semantic coverage. As a result, the selected tokens span a broader region in the embedding space.

In Fig. 8, we further visualize the attention distribution of selected tokens. Our method preserves the majority of high-attention tokens, demonstrating its ability to retain both informative and representative visual content.

E Broader Impact

Our proposed method aims to improve both the efficiency and effectiveness of multimodal large language models (MLLMs) by reducing the number of visual tokens while preserving semantic completeness. This advancement has the potential to significantly reduce the computational cost and memory footprint of MLLMs, thereby enhancing their feasibility for deployment in resource-constrained environments such as edge devices, mobile platforms, and real-time applications. By enabling more efficient inference, our approach can facilitate the broader adoption of vision-language models across various domains, including education, healthcare, and assistive technologies.

However, as with any technology that enhances the scalability and accessibility of AI systems, there are potential societal risks. For example, more efficient MLLMs could be misused to generate or disseminate misinformation, enable invasive surveillance, or support other malicious activities,



Figure 7: The selected token comparison between the saliency-based method and our saliency-coverage oriented method. The total visual token number is 576, and the selected token number is 64.

particularly when deployed at scale. It is therefore essential to consider these ethical implications and implement appropriate safeguards when deploying such models in practice.

F Limitations

While SCOPE demonstrates strong performance and efficiency gains across multiple benchmarks and model architectures, several limitations remain. (1) Despite our efforts to balance saliency and coverage, aggressive token pruning may still result in the loss of fine-grained or rare semantic information, potentially affecting tasks that require detailed visual understanding. (2) Our experiments are primarily based on widely used vision-language benchmarks and two representative MLLMs, LLaVA 1.5 and LLaVA-Next. Therefore, the generalizability of SCOPE to other tasks or model architectures has yet to be fully validated.

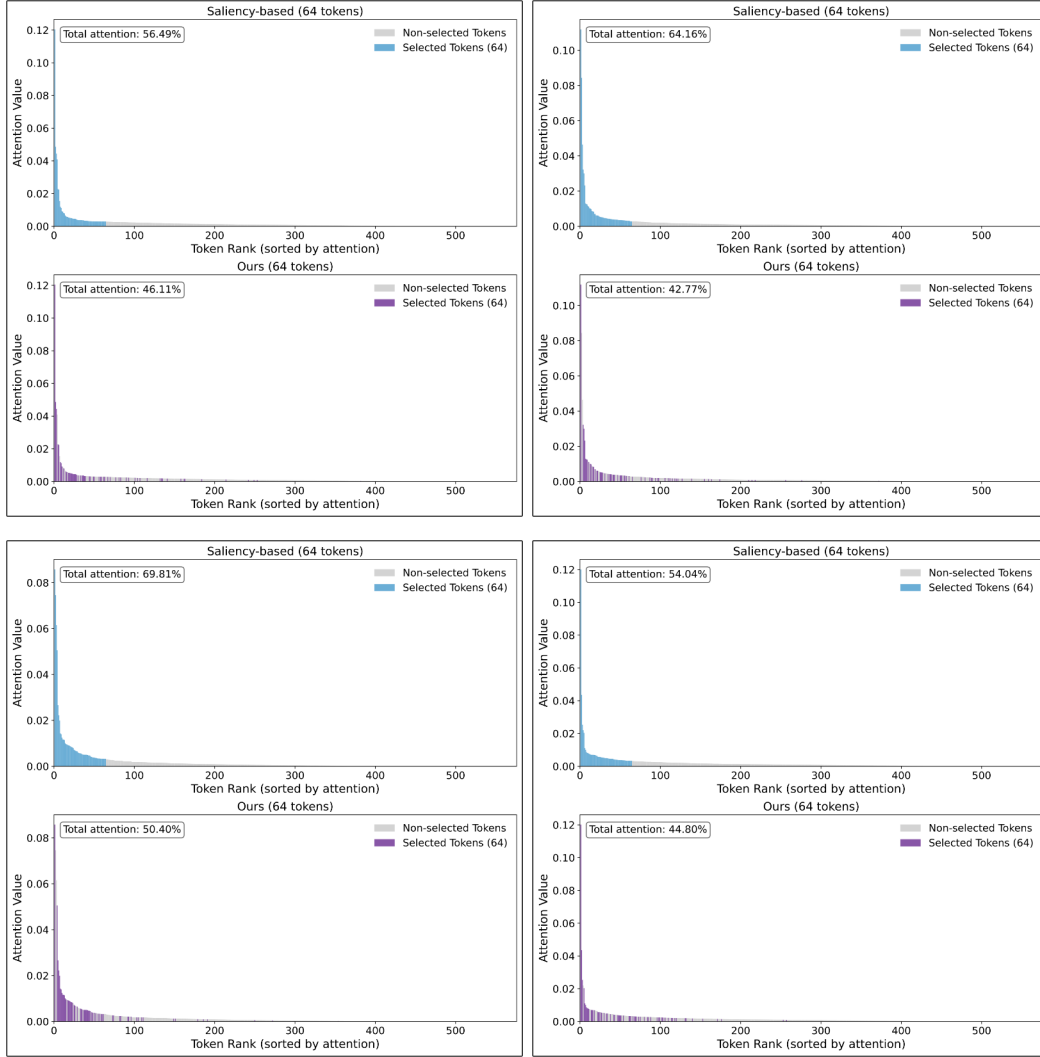


Figure 8: Attention distribution visualization for selected token. The total visual token number is 576, and the selected token number is 64. Our method retained most of the high attention tokens and some low attention tokens to maximize the coverage.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: We illustrate the contributions and scope of this paper in the abstract and introduction.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: We have discussed the limitations.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[NA\]](#)

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: We have provided the implemented details.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The data is open access on the internet. We will release the code once this work is accepted.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We have provided the details in the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Due to the complexity of MLLMs, we do not report error bars following previous works. The effectiveness of the proposed method is validated on various benchmarks and multiple MLLMs across different scales of model size.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We have provided the GPU information.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We conform to the NeurIPS code of ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discussed the broader impacts.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We use the public datasets for academic purposes only.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have cited the original papers.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset’s creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: We provided the details of the dataset and implementation details of the proposed method.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[NA\]](#)

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [\[NA\]](#)

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.