ViTCoP: Accelerating Large Vision-Language Models via Visual and Textual Semantic Collaborative Pruning

Wen Luo, Peng Chen, Xiaotao Huang*, LiQun Huang*

School of Software Engineering, Huazhong University of Science and Technology, WuHan, China
{wen_chaser, hustchenpeng}@hust.edu.cn, {huangxiaotao, huangliqun}@mail.hust.edu.cn

Abstract

Large Vision-Language Models (LVLMs) incur high computational costs due to significant redundancy in their visual tokens. To effectively reduce this cost, researchers have proposed various visual token pruning methods. However, existing methods are generally limited, either losing critical visual information prematurely due to pruning in the vision encoder, or leading to information redundancy among the selected tokens due to pruning in the Large Language Models (LLMs). To address these challenges, we propose a Visual and Textual Semantic Collaborative Pruning framework (ViTCoP) that combines redundancy filtering in the vision encoder with step-wise co-pruning within the LLM based on its hierarchical characteristics, to efficiently preserve critical and informationally diverse visual tokens. Meanwhile, to ensure compatibility with acceleration techniques like FlashAttention, we introduce the L2 norm of K-vectors as the token saliency metric in the LLM. Extensive experiments on various Large Vision-Language Models demonstrate that ViTCoP not only achieves state-of-the-art performance surpassing existing methods on both image and video understanding tasks, but also significantly reduces model inference latency and GPU memory consumption. Notably, its performance advantage over other methods becomes even more pronounced under extreme pruning rates.

Code — https://github.com/chaser682/ViTCoP

1 Introduction

The monumental success of Large Language Models (LLMs) in the domain of language understanding (Achiam et al. 2024; Chiang et al. 2023; Touvron et al. 2023; Yang et al. 2025) has catalyzed the proliferation and remarkable advancement of Large Vision-Language Models (LVLMs). LVLMs (Lin et al. 2024; Liu et al. 2023, 2024a; Zhang et al. 2024d) operate by encoding visual information from images and videos into a vast number of visual tokens. Through a lightweight modality-alignment module (Liu et al. 2023; Bai et al. 2025; Li et al. 2023a), these visual tokens are concatenated with text tokens and subsequently fed into an LLM for instruction fine-tuning (Liu et al. 2023). This paradigm has endowed LVLMs with powerful multimodal perception and reasoning capabilities across a spectrum of tasks, including image comprehension and video question-answering.

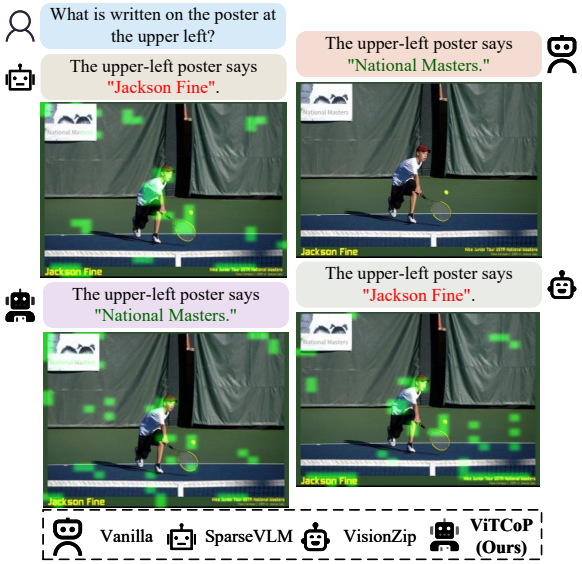


Figure 1: Visual question answering results of LLaVA-1.5-7B with different pruning methods.

However, despite their exceptional performance, the substantial computational cost of LVLMs presents a critical bottleneck. The inherent density of visual information, particularly in high-resolution images or long videos, results in the generation of thousands, or even tens of thousands, of visual tokens (Zhang et al. 2024b; Chen et al. 2024a; Maaz et al. 2024). Given that the computational complexity of the Transformer architecture scales quadratically with the input sequence length, this deluge of visual tokens leads to prohibitive inference latency and GPU memory consumption. This overhead severely constrains the efficient deployment and application of LVLMs in resource-constrained environments such as autonomous driving, robotics, and edge computing (Kim et al. 2024; Liu et al. 2024b; Qu et al. 2025; Yang et al. 2024b; Yao et al. 2024).

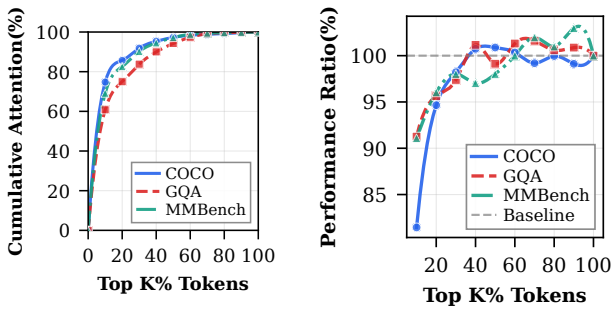
Existing research indicates that a high degree of in-

*Corresponding Author.

formation redundancy exists among the visual tokens in LVLMs (Chen et al. 2024b; Shang et al. 2024; Xing et al. 2025; Zhang et al. 2025; Yang et al. 2024a). To address this challenge, visual token pruning has emerged as a promising technical direction, with current work broadly categorized into two paradigms. The first is text-agnostic pruning, which operates solely on visual information without considering the specific text instruction. For instance, VisionZip (Yang et al. 2024a) identifies dominant tokens via attention scores and employs a token fusion strategy to extract contextually rich representations. The fundamental limitation of such methods, however, is their disregard for guidance from the language instruction. As illustrated in Figure 1, when asked "What is written on the poster at the upper left?", a text-agnostic method like VisionZip retains many visually salient tokens from the player and court, but may fail to focus on the specific poster, leading to an incorrect answer.Since user queries often pertain to specific regions, this text-agnostic strategy may preserve task-irrelevant visual information, degrading model performance.The second category, text-guided pruning, leverages the textual instruction to direct the process. Methods like FastV (Chen et al. 2024b) and PyramidDrop (Xing et al. 2025) use text-attention scores to identify and discard unimportant tokens, but this may leads to high redundancy among the selected tokens.Similarly, SparseVLM (Zhang et al. 2025) employs visually-relevant text tokens as raters to filter for important visual tokens. However, it also has its limitations. When text instructions are broad or focus on similar concepts, the visual tokens selected under this guidance may exhibit significant content overlap, leading to high information redundancy and insufficient diversity.Consequently, existing methods face a significant challenge: purely visual pruning risks losing critical details, while purely text-guided pruning in the LLM tends to yield high informational redundancy.

To resolve these challenges, we propose ViTCoP, a Visual-Text Collaborative Pruning framework. Our core insight is that an optimal pruning strategy must synergistically leverage semantic information from different modalities at distinct stages of the LVLM's processing pipeline. To this end, ViTCoP employs an innovative three-stage strategy. First, within the vision encoder, we perform a coarse-grained, visually-guided pruning to remove patently redundant tokens from backgrounds or repetitive textures. Second, in the shallow layers of the LLM, where the model performs initial global cross-modal understanding (Neo et al. 2025; Zhang et al. 2024c), we employ a vision-text synergistic pruning to ensure the retained tokens are both highly relevant to the query and semantically diverse. Finally, in the deep layers of the LLM, as the model's understanding of the instruction becomes progressively more focused (Parekh et al. 2024; Chen et al. 2024b; Xing et al. 2025), we transition to a text-guided, fine-grained pruning to further refine the selection down to the core visual evidence most directly pertinent to the final answer.

Through this hierarchical and progressive strategy, ViTCoP adeptly balances the preservation of critical information with the promotion of token diversity. Furthermore, to ensure compatibility with modern acceleration techniques such as FlashAttention (Dao et al. 2022; Dao 2023), we innovatively introduce the L2 norm of key vectors as a lightweight yet effective saliency metric for token selection in LVLMs. Extensive experiments on multiple mainstream LVLMs demonstrate that ViTCoP not only achieves state-of-the-art performance on image and video understanding benchmarks but also significantly reduces inference latency and GPU memory footprint.



(a) Attention Cumulative Distribution (b) Performance vs. Top K% Token

Figure 2: Analysis of initial visual token redundancy. (a) A small fraction of tokens captures a majority of the attention score. (b) Model performance shows minimal degradation even when a large portion of tokens is pruned.

2 Insights

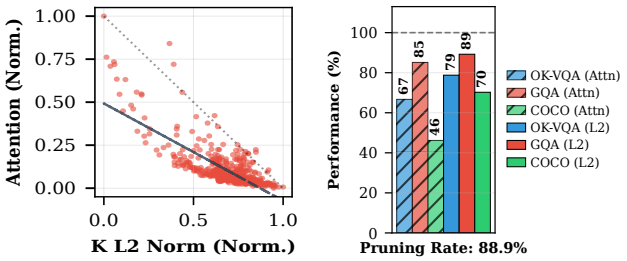
2.1 Initial Redundancy of Visual Tokens

Our study reveals significant initial redundancy in visual tokens generated by the Vision Transformer. On the LLaVA-1.5-7B model (Liu et al. 2023), we found that the top 10% of tokens with the highest attention scores contribute over 60% of the total attention weight (Figure 2a). More importantly, retaining just the top 20% of tokens is sufficient to maintain approximately 95% of the model's performance across various image-language understanding benchmarks (Figure 2b). This confirms that a small subset of visual tokens can represent the vast majority of an image's information.

Key Insight 1: A large number of visual tokens can be pruned before entering the LLM with minimal impact on model performance.

2.2 K-Vector L2 Norm: An Efficient Proxy for Token Saliency

Pruning based on attention scores, as used in methods like FastV (Chen et al. 2024b), is effective but often incompatible with modern computational optimizations like FlashAttention (Dao et al. 2022; Dao 2023). Inspired by recent work (Devoto et al. 2024), we investigate the L2 norm of Key (K) vectors as a lightweight proxy. Our analysis reveals a strong negative correlation between the K-vector L2 norm and attention scores (Figure 3a). Furthermore, comparative experiments show that pruning based on the smallest L2 norm achieves performance that is competitive with, and



(a) Correlation Analysis (b) Performance Comparison

Figure 3: Validation of K-vector L2 norm as a saliency proxy. (a) A strong negative correlation exists between L2 norm and attention. (b) L2 norm-based pruning is competitive with, or superior to, attention-based methods.

at times superior to, attention-based pruning across multiple benchmarks (Figure 3b).

Key Insight 2: In LVLMs, the K-vector L2 norm is a lightweight and effective proxy for token saliency within the LLM, where a smaller norm corresponds to higher importance.

2.3 Evolving Importance of Visual Tokens in LLM

The importance of visual tokens is not static but evolves as they propagate through the LLM layers. By analyzing the distribution of attention scores across different layers (Figure 4), we observe a clear functional shift: the LLM transitions from aggregating diverse, global information in the shallow layers to focusing on key local details in the deep layers.

Key Insight 3: The LLM aggregates global visual information in shallow layers and focuses on absorbing key local visual information in deep layers.

3 Method

In this paper, we propose ViTCoP, a dynamic token pruning framework based on Visual-Textual Semantic Collaborative Pruning. The core strategy of ViTCoP is to synergistically leverage visual-textual semantic information to perform a multi-stage, differentiated pruning adapted to the different phases of a LVLM. As illustrated in Figure 5, ViTCoP consists of three stages: (I) Coarse-grained pruning guided by visual saliency in the vision encoder; (II) Collaborative visual-textual semantic-guided pruning in the shallow layers of the LLM to acquire tokens that are both semantically diverse and text-relevant; and (III) Fine-grained pruning guided by textual saliency in the deep layers of the LLM. Through this synergistic visual-textual pruning strategy, ViTCoP strikes a balance between preserving critical and diverse information.

3.1 Stage I: Visual Saliency-Guided Pruning in the Vision Encoder

As discussed in Section 2.1, a significant number of redundant tokens already exist in the vision encoder. Therefore,

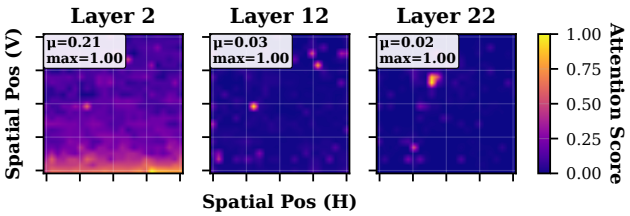


Figure 4: Heatmap of visual token attention scores across LLM layers.

this initial stage aims to eliminate highly redundant tokens, such as those from information-sparse backgrounds or repetitive textures, to provide a high-quality input for the subsequent fine-grained pruning within the LLM. Specifically, for the visual tokens entering the LVLM's projection layer (including the `[CLS]` token in CLIP (Radford et al. 2021)), we define the saliency score of the i-th visual token based on the attention it receives from the `[CLS]` token:

$$S_i = \sum_{h=1}^{H} \mathbf{A}_{0,i}^{(h)}, \tag{1}$$

where H is the number of attention heads, and $\mathbf{A}_{0,i}^{(h)}$ represents the attention score from the `[CLS]` token (at index 0) to the i-th visual token in the h-th attention head. By ranking the visual tokens based on their saliency scores S_i and selecting the top-ranking ones, this stage preserves high-saliency tokens rich in information, removing useless redundancy for the subsequent pruning stages in the LLM.

3.2 Stage II: Visual-Textual Collaborative Pruning in Shallow LLM Layers

As established in Section 2.3, the LLM needs to perform a preliminary global understanding by integrating both visual and textual information in its shallow layers. Therefore, we employ a collaborative visual-textual semantic-guided pruning strategy to ensure that the retained tokens are not only semantically diverse but also highly relevant to the text.

Visual Semantic Guidance: VIC Algorithm For visual semantic guidance, we introduce the Visual Information Clustering (VIC) algorithm, designed to preserve the diversity of visual semantic information. Specifically, the inputs to VIC are the feature vectors of the high-saliency tokens retained from Stage I and their corresponding position vectors in the original image. The output of our algorithm depends on three parameters: a cutoff distance (d_c), a spatial threshold (τ), and a ratio of cluster centers. We calculate feature and spatial distances, and the local density ρ_i for each token i is computed as:

$$\rho_i = \sum_{j \neq i} \exp\left(- \left(\frac{d_{ij}}{d_c} \right)^2 \right), \tag{2}$$

where d_{ij} denotes the feature distance between tokens i and j.

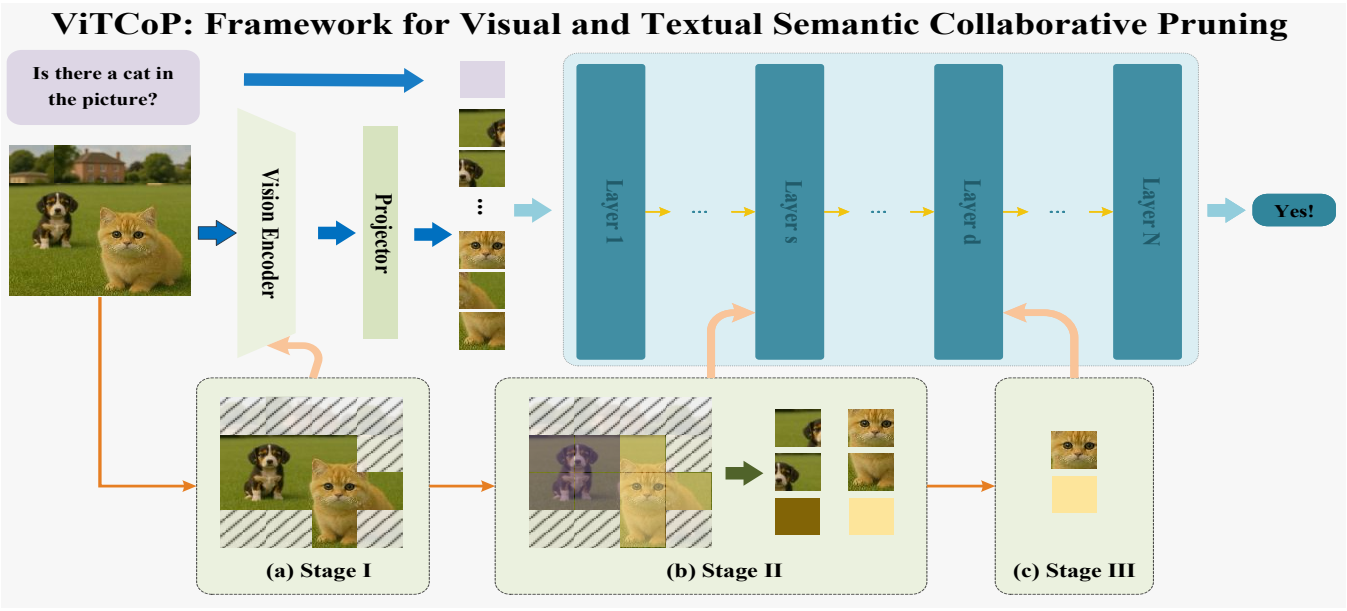


Figure 5: The ViTCoP framework's three-stage process: (a) Coarse pruning in the Vision Encoder via '[CLS]' attention, (b) Collaborative pruning in shallow LLM layers using VIC clustering and K-norm merging, and (c) Aggressive text-saliency pruning in deep LLM layers.

For each token i, we find the minimum feature distance δ_i to another token j that has a higher density ($\rho_j > \rho_i$) and is within the spatial distance threshold τ:

$$\delta_i = \min_{\substack{j:\rho_j>\rho_i \\ d_{\text{spatial}}(i,j)\leq\tau}} d_{ij}, \tag{3}$$

where $d_{\text{spatial}}(i, j)$ represents the spatial distance between tokens i and j.

We then calculate an importance score $\gamma_i = \rho_i \cdot \delta_i$ for each token, where tokens with the highest importance scores are designated as cluster centers. Subsequently, each non-center token is assigned to the cluster of its nearest center. Our algorithm ensures that each token is clustered into a semantically coherent group, thereby satisfying the subsequent need to retain semantically diverse tokens.

Textual Semantic Guidance As noted in Section 2.2, the L2 norm of the Key (K) vectors exhibits a strong negative correlation with attention scores. That is, visual tokens more relevant to the text tend to have smaller K-vector L2 norms. Therefore, we use the L2 norm of the K vectors from the LLM's attention module as a token saliency metric. The L2 norm of a token's K vector is calculated as:

$$\|\mathbf{K}_i\|_2 = \sqrt{\sum_{h=1}^{H} \|\mathbf{K}_i^{(h)}\|_2^2}, \tag{4}$$

where H is the number of attention heads and $\mathbf{K}_i^{(h)}$ is the K vector of the i-th token in the h-th head.

Collaborative Pruning and Merging To achieve collaborative pruning guided by both visual and textual semantics, we proceed as follows. Given a set of visual tokens with their cluster labels from the VIC algorithm and their K-vector L2 norms, we first assign a retention quota q_c to each cluster c. This quota determines the number of elite tokens to be retained from that cluster and is proportional to the cluster's relative size, ensuring minimal information loss:

$$q_c = \left\lfloor \frac{|C_c|}{\sum_{k=1}^{N_c} |C_k|} \cdot (B - N_c) \right\rfloor, \tag{5}$$

where B is the total budget for elite tokens, $|C_c|$ is the size of cluster c, and N_c is the total number of clusters. For the selection of elite tokens within each cluster, we select the top q_c tokens with the smallest K-vector L2 norms, as a smaller norm indicates higher relevance to the text. Finally, the remaining tokens within each cluster are merged into a single representative token by averaging their feature vectors:

$$\mathbf{t}_c^{\text{merged}} = \frac{1}{|C_c^{\text{remaining}}|} \sum_{i \in C_c^{\text{remaining}}} \mathbf{t}_i, \tag{6}$$

where $C_c^{\text{remaining}}$ denotes the set of remaining tokens in cluster c after elite selection, and $\mathbf{t}_i$ represents the feature vector of token i. This collaborative approach ensures that both fine-grained details and generalized context are preserved.

3.3 Stage III: Textual Saliency-Guided Pruning in Deep LLM Layers

Once the token sequence propagates to the deep layers of the LLM, the model has progressively absorbed a substantial amount of semantic information from the visual tokens. As per Section 2.3, the LLM in its deep layers focuses on

Method	COCO	Flickr	GQA	MMB	MME	NoCaps	OK-VQA	POPE	QBench	SQA	VQA-v2	Avg (%)
Vanilla	**1.102** (100.0%)	**0.750** (100.0%)	**0.619** (100.0%)	**64.08** (100.0%)	**1862** (100.0%)	**1.055** (100.0%)	**0.534** (100.0%)	**0.858** (100.0%)	**0.585** (100.0%)	**0.695** (100.0%)	**0.716** (100.0%)	**100.0%**
Retain 192 Tokens (↓ 66.7%)												
FastV	1.082 (98.1%)	**0.741** (**98.7%**)	0.527 (85.1%)	60.57 (94.5%)	1612 (86.6%)	**1.033** (**97.9%**)	0.512 (95.9%)	0.646 (75.3%)	0.581 (99.3%)	0.672 (96.7%)	0.663 (92.6%)	92.8%
PyramidDrop	**1.091** (**99.0%**)	0.734 (97.9%)	0.574 (92.7%)	63.75 (99.5%)	1797 (96.5%)	1.023 (97.0%)	0.508 (95.1%)	0.810 (94.4%)	**0.581** (**99.3%**)	**0.692** (**99.6%**)	0.678 (94.7%)	96.9%
SparseVLM	1.087 (98.6%)	0.720 (95.9%)	0.576 (93.0%)	62.92 (98.2%)	1721 (92.4%)	1.010 (95.7%)	0.520 (97.4%)	0.837 (97.5%)	0.575 (98.3%)	0.692 (99.6%)	**0.706** (**98.6%**)	96.8%
VisionZip	1.070 (97.0%)	0.737 (98.3%)	0.593 (95.8%)	63.66 (99.3%)	1782 (95.7%)	1.023 (97.0%)	0.525 (98.3%)	0.853 (99.4%)	0.575 (98.3%)	0.689 (99.1%)	0.686 (95.8%)	97.6%
ViTCoP (Ours)	1.078 (97.8%)	0.735 (98.0%)	**0.600** (**96.9%**)	**64.26** (**100.3%**)	**1816** (**97.5%**)	1.019 (96.6%)	**0.536** (**100.4%**)	**0.855** (**99.6%**)	0.579 (99.0%)	0.684 (98.4%)	0.705 (98.5%)	**98.5%**
Retain 128 Tokens (↓ 77.8%)												
FastV	1.044 (94.7%)	0.719 (95.8%)	0.496 (80.1%)	57.29 (89.4%)	1490 (80.0%)	0.995 (94.3%)	0.486 (91.0%)	0.597 (69.5%)	0.579 (99.0%)	0.602 (86.6%)	0.632 (88.3%)	88.9%
PyramidDrop	1.039 (94.2%)	0.692 (92.2%)	0.572 (92.4%)	59.89 (93.5%)	1761 (94.6%)	0.969 (91.8%)	0.491 (91.9%)	0.738 (86.0%)	**0.581** (**99.3%**)	0.684 (98.4%)	0.650 (90.8%)	93.2%
SparseVLM	0.940 (85.3%)	0.583 (77.7%)	0.561 (90.6%)	60.71 (94.7%)	1696 (91.1%)	0.823 (78.0%)	0.509 (95.3%)	0.805 (93.8%)	0.572 (97.8%)	0.672 (96.7%)	**0.684** (**95.5%**)	90.6%
VisionZip	1.037 (94.1%)	0.713 (95.1%)	0.576 (93.0%)	62.37 (97.3%)	1761 (94.6%)	0.989 (93.7%)	0.507 (95.0%)	0.833 (97.1%)	0.570 (97.4%)	**0.689** (**99.1%**)	0.665 (92.9%)	95.4%
ViTCoP (Ours)	**1.064** (**96.5%**)	**0.724** (**96.5%**)	**0.592** (**95.6%**)	**63.83** (**99.6%**)	**1785** (**95.9%**)	**1.008** (**95.5%**)	**0.531** (**99.4%**)	**0.846** (**98.6%**)	0.577 (98.6%)	0.684 (98.4%)	0.682 (95.2%)	**97.3%**
Retain 64 Tokens (↓ 88.9%)												
FastV	0.815 (73.9%)	0.511 (68.1%)	0.462 (74.6%)	50.43 (78.7%)	1256 (67.5%)	0.768 (72.8%)	0.370 (69.3%)	0.483 (56.3%)	0.540 (92.3%)	0.512 (73.7%)	0.503 (70.2%)	72.5%
PyramidDrop	0.648 (58.8%)	0.372 (49.6%)	0.475 (76.7%)	56.10 (87.5%)	1561 (83.8%)	0.627 (59.4%)	0.395 (74.0%)	0.692 (80.6%)	0.551 (94.2%)	0.608 (87.5%)	0.578 (80.7%)	76.6%
SparseVLM	0.731 (66.3%)	0.419 (55.9%)	0.527 (85.1%)	57.90 (90.4%)	1505 (80.8%)	0.584 (55.4%)	0.451 (84.5%)	0.758 (88.3%)	0.563 (96.2%)	0.622 (89.5%)	0.615 (85.9%)	80.7%
VisionZip	0.948 (86.0%)	0.651 (86.8%)	0.551 (89.0%)	60.31 (94.1%)	1690 (90.8%)	0.900 (85.3%)	0.478 (89.5%)	0.771 (89.9%)	0.559 (95.6%)	**0.690** (**99.3%**)	0.631 (88.1%)	90.4%
ViTCoP (Ours)	**1.032** (**93.6%**)	**0.696** (**92.8%**)	**0.574** (**92.7%**)	**63.06** (**98.4%**)	**1744** (**93.7%**)	**0.973** (**92.2%**)	**0.508** (**95.1%**)	**0.807** (**94.1%**)	**0.568** (**97.1%**)	0.688 (99.0%)	**0.663** (**92.6%**)	**94.7%**

Table 1: Performance on LLaVA-1.5-7B. Each cell shows the score and retention rate (%). The **best** result in each group is highlighted.

assimilating key local visual information. As the model's understanding of visual information deepens and becomes more focused, a high degree of redundancy emerges among the visual tokens because their core information has been effectively captured. Therefore, we employ a text-saliency-only guided pruning in the deep LLM layers to eliminate a large number of visual tokens that are either irrelevant to the text or whose information has already been aggregated and understood by the model. Specifically, we use the L2 norm of the visual token's K vectors (as defined in Eq. 4) to retain the top-ranking salient tokens that contain key local information.

This three-stage, coarse-to-fine filtering significantly enhances ViTCoP's efficiency while maintaining performance.

4 Experiments

4.1 Experimental Settings

Baselines and Models To evaluate the effectiveness of our proposed ViTCoP framework, we compare it against four recent and competitive token pruning baselines: FastV(Chen et al. 2024b), PyramidDrop(Xing et al. 2025), SparseVLM(Zhang et al. 2025), and VisionZip(Yang et al. 2024a). Our experiments are conducted on a suite of LVLMs to demonstrate its broad applicability. Specifically, we use LLaVA-1.5-7B (Liu et al. 2023)for image task evaluation, and the more advanced LLaVA-NeXT-7B(Liu et al. 2024a) and LLaVA-NeXT-Video-7B(Zhang et al. 2024d) for high-resolution image and video evaluations, respectively.

Datasets Our evaluation covers a wide range of standard benchmarks to ensure a comprehensive assessment of performance across both image and video understanding tasks. For the image-language evaluation, we used 11 diverse datasets: COCO-2017(Lin et al. 2015), Flickr30k(Young et al. 2014), GQA(Hudson and Manning 2019), MMBench(Liu et al. 2024c), MME(Fu et al. 2024), Nocaps(Agrawal et al. 2019), OK-VQA(Marino et al. 2019), POPE(Li et al. 2023b), QBench(Wu et al. 2024), ScienceQA(Lu et al. 2022), and VQA-v2(Goyal et al. 2017).

Method	COCO	GQA	MMB	POPE	Avg(%)
Vanilla	**1.000 (100.0%)**	**0.643 (100.0%)**	**67.01 (100.0%)**	**0.865 (100.0%)**	**100.0%**
Retain 320 Tokens (↓ 88.9%)					
FastV	0.629 (62.9%)	0.533 (82.9%)	58.68 (87.6%)	0.599 (69.2%)	75.7%
SparseVLM	0.839 (83.9%)	0.578 (89.9%)	**64.78 (96.7%)**	0.827 (95.7%)	91.6%
PyramidDrop	0.625 (62.5%)	0.375 (58.3%)	59.36 (88.6%)	0.659 (76.2%)	71.4%
VisionZip	0.826 (82.6%)	0.593 (92.2%)	63.83 (95.2%)	0.824 (95.3%)	91.4%
ViTCoP (Ours)	**0.912 (91.2%)**	**0.610 (94.9%)**	**64.78 (96.7%)**	**0.846 (97.8%)**	**95.1%**
Retain 160 Tokens (↓ 94.4%) *					
VisionZip	0.697 (69.7%)	0.556 (86.5%)	60.05 (89.6%)	0.757 (87.5%)	83.3%
ViTCoP (Ours)	**0.844 (84.4%)**	**0.584 (90.8%)**	**62.89 (93.8%)**	**0.816 (94.3%)**	**90.8%**

Table 2: Performance comparison on 4 key datasets from LLaVA-NeXT-7B. *At 94.4% compression, some methods are omitted due to incompatibility.

For the video-language evaluation, we utilized 4 representative datasets: EgoSchema(Mangalam, Akshulakov, and Malik 2023), MVBench(Li et al. 2024), Next-QA(Xiao et al. 2021), and Video-MME(Fu et al. 2025).

Implementation Details For our ViTCoP framework, we configure the three-stage pruning process as follows: the first stage occurs at the output of the vision encoder, while the second and third stages are applied at the 2nd and 22nd layers of the LLM, respectively. For the VIC clustering algorithm, we set the distance threshold $d_c = 8$ and the spatial threshold $\tau = 0.6$. These hyperparameters were established based on preliminary experiments. They were kept fixed across all benchmarks without any dataset-specific fine-tuning to validate the robustness and strong generalization capabilities of our method. To ensure a fair comparison, all baseline methods adhere to their original experimental settings. All experiments were conducted on NVIDIA V100s GPUs, and all benchmarks were run using the `lmms-eval` package (Zhang et al. 2024a).

4.2 Image-Language Understanding Tasks

In this section, we systematically evaluate the performance and robustness of ViTCoP on two mainstream large vision-language models. We first conduct comprehensive tests on the LLaVA-1.5-7B model across 11 mainstream benchmark datasets. Subsequently, we further validate the scalability of ViTCoP under extreme compression scenarios on the higher-resolution LLaVA-NeXT-7B model.

Performance on LLaVA-1.5-7B We evaluate the performance of ViTCoP under three pruning intensities: retaining 192 (66.7% pruning), 128 (77.8% pruning), and 64 (88.9% pruning) tokens from the original 576 visual tokens. As shown in Table 1, ViTCoP achieves the best average performance across all compression settings, significantly outperforming existing methods. For instance, at a moderate pruning rate (192 tokens), ViTCoP improves upon the next-best method, VisionZip, by 0.9%. At an aggressive pruning of 64 tokens, ViTCoP still maintains 94.7% performance, surpassing VisionZip and SparseVLM by 4.3% and 14%, respectively. It is worth noting that on some datasets, ViTCoP even exceeds the performance of the original model, reaching 100.3% on MMBench and 100.4% on OK-VQA. This suggests that our method not only effectively removes redundancy but can also mitigate the impact of interfering information on the model.

Method	EgoSch	MVB	Next-QA	V-MME	Avg (%)
Vanilla	**0.414 (100.0%)**	**44.95 (100.0%)**	**26.64 (100.0%)**	**32.41 (100.0%)**	**100.0%**
Retain 128 Tokens (↓ 88.9%)					
FastV	0.345 (83.2%)	40.78 (90.7%)	23.99 (90.1%)	29.26 (90.3%)	88.6%
PyramidDrop	0.357 (86.3%)	38.80 (86.3%)	21.52 (80.8%)	29.81 (92.0%)	86.4%
SparseVLM	**0.406 (98.0%)**	43.13 (96.0%)	24.77 (93.0%)	30.30 (93.5%)	95.1%
VisionZip	0.370 (89.3%)	40.80 (90.8%)	23.36 (87.7%)	30.26 (93.4%)	90.3%
ViTCoP (Ours)	0.405 (97.7%)	**43.30 (96.3%)**	**25.60 (96.1%)**	**32.67 (100.8%)**	**97.7%**

Table 3: Performance on 4 video benchmarks from LLaVA-NeXT-Video-7B.

Performance on LLaVA-NeXT-7B To verify the generalization capability of ViTCoP on high-resolution images, we conducted further experiments on the LLaVA-NeXT-7B model, which uses 2880 visual tokens, making more extreme pruning settings possible. We focused on evaluating two pruning rates: 88.9% and 94.4%. As shown in Table 2, ViTCoP retains 95.1% of the average performance at an 88.9% pruning rate, significantly outperforming VisionZip's 91.4%. At the more aggressive 94.4% pruning rate, ViTCoP still achieves a 90.8% retention rate, far exceeding VisionZip's 83.3%. Notably, other methods such as FastV, PyramidDrop, and SparseVLM failed to run under this compression intensity and were therefore not included in the comparison. These results further validate the stability and strong generalization capability of ViTCoP under extreme compression conditions.

4.3 Video-Language Understanding Tasks

This section further evaluates the generalization and robustness of ViTCoP on dynamic temporal data. We extend the evaluation from static images to the video domain, conducting experiments with the LLaVA-NeXT-Video-7B model on four representative video question-answering datasets: EgoSchema, MVBench, Next-QA, and Video-MME. For these tasks, we uniformly apply an aggressive pruning rate of 88.9%.

Ablation	TFLOPs	COCO	GQA	MMB	POPE
ViTCoP (Ours)	0.82	1.032	**0.574**	**63.06**	**0.807**
w/o K-norm Guidance	0.82	1.011	0.556	62.54	0.760
w/o Attention Guidance	0.82	1.015	0.563	62.63	0.771
w/o Stage I Pruning	0.91	0.086	0.389	20.27	0.283
w/o Stage III Pruning	0.81	**1.046**	0.571	62.46	0.784

Table 4: Ablation study. "w/o K-norm Guidance" uses only attention; "w/o Attention Guidance" uses only K-vector L2-norms. TFLOPs is avg. cost on COCO.

	POPE	TFLOPs	GPU Mem	Prefill	Time/Tok
LLaVA-NeXT	**0.863 (100%)**	**31.55 (100%)**	**30.80 (100%)**	**914 (100%)**	**62.67 (100%)**
VisionZip	0.661 (76.6%)	1.79 (5.7%)	**27.12 (88.1%)**	**126 (13.8%)**	**53.39 (85.2%)**
ViTCoP (Ours)	**0.755 (87.5%)**	**1.69 (5.4%)**	27.13 (88.1%)	139 (15.2%)	53.53 (85.4%)

Table 5: Efficiency analysis of ViTCoP on LLaVA-NeXT-13B. Units: TFLOPs for computation, GB for GPU Memory, and ms for latency (Prefill and Time/Token).

As shown in Table 3, ViTCoP retains 96.3% and 96.1% of the performance on MVBench and Next-QA, respectively. In terms of average performance, ViTCoP (97.7%) significantly outperforms SparseVLM (95.1%) and achieves the best results on three of the four benchmarks. ViTCoP's performance on Video-MME is particularly outstanding, reaching 100.8% and even surpassing the original, unpruned model. Overall, our method achieves an average performance retention rate of 97.7% across the four datasets, fully demonstrating ViTCoP's excellent generalization capabilities in video-language large models. These results indicate that ViTCoP not only excels in static image understanding but also maintains exceptional performance in complex video-language tasks, establishing it as a general and robust token pruning framework.

4.4 Ablation Study

To evaluate ViTCoP's key components, we conduct an ablation study on the COCO, GQA, MMBench, and POPE datasets. We assess the multistage pruning and saliency metrics by creating four variants: *w/o K-norm Guidance*, using only attention scores; *w/o Attention Guidance*, using only the K-vector L2-norm; *w/o Stage I Pruning*, removing the initial coarse-grained pruning; and *w/o Stage III Pruning*, removing the final fine-grained pruning. Stage II is not ablated as it is integral to the pruning pipeline.

The ablation results in Table 4 demonstrate that the full ViTCoP method significantly outperforms all variants. In particular, when using only the K-vector L2-norm as the saliency metric (*w/o Attention Guidance*), performance does not degrade compared to *w/o K-norm Guidance*, but it even slightly improves on some tasks. This highlights the K-vector L2-norm as an effective and robust proxy for token importance, with strong generalization and compatibility with modern acceleration techniques like FlashAttention. Additionally, the absence of visual guidance from VIC in the second stage, which relies only on the L2-norm for text-guided pruning, results in redundancy among retained key tokens, and thus degrades performance compared to the full ViTCoP.

However, removing the first stage pruning (*w/o Stage I Pruning*) resulted in a catastrophic performance decline. This outcome demonstrates that the initial removal of irrelevant tokens–such as redundant backgrounds, low-information regions, or repetitive textures–is crucial for alleviating the burden on subsequent pruning stages. Without this stage, the following stages struggle to discern redundancy, thereby retaining excessive noisy tokens that severely interfere with the model's representation capabilities.

Interestingly, removing the third stage pruning (*w/o Stage III Pruning*) led to a slight improvement in the COCO dataset. This may be because the image-text matching task in COCO is highly sensitive to the aggregation of fine-grained visual semantics, and the further pruning in the third stage might inadvertently remove some detailed information, affecting the final performance.

In summary, the three stages of our method form a complementary and synergistic relationship. Removing any single stage leads to a performance drop or even significant degradation, highlighting the critical role of ViTCoP's multistage, progressive pruning strategy in achieving both effectiveness and robustness.

4.5 Efficiency Analysis

ViTCoP achieves significant inference acceleration and computational savings by substantially reducing the number of visual tokens processed by the LLM. On the POPE dataset, we conduct a comparison based on LLaVA-NeXT-13B (Liu et al. 2024a) against the vanilla model and VisionZip. As shown in Table 5, ViTCoP reduces TFLOPs by over 94%, decreases prefill latency by 85%, and significantly shortens the generation time per token. Despite having efficiency comparable to VisionZip, ViTCoP demonstrates about 10% higher performance retention, showcasing a superior trade-off between efficiency and performance.

5 Conclusion

In this paper, we introduce ViTCoP, a visual-textual semantic collaborative pruning framework designed to ensure that retained visual tokens are both crucial and informationally diverse. Extensive experiments on image and video understanding tasks demonstrate its effectiveness. ViTCoP maintains nearly 95% of baseline performance at a high compression rate of 88.9% and achieves performance retention of up to 97.7% on video tasks, comprehensively outperforming existing state-of-the-art methods. As a tuning-free framework, ViTCoP reduces the TFLOPs of the model by more than 94% while significantly reducing inference latency and GPU memory consumption. This offers a superior solution for the efficient deployment of Large Vision-Language Models in resource-constrained environments.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2024. GPT-4 Technical Report. arXiv:2303.08774.

Agrawal, H.; Desai, K.; Wang, Y.; Chen, X.; Jain, R.; Johnson, M.; Batra, D.; Parikh, D.; Lee, S.; and Anderson, P. 2019. nocaps: novel object captioning at scale. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE.

Bai, S.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Song, S.; Dang, K.; Wang, P.; Wang, S.; Tang, J.; Zhong, H.; Zhu, Y.; et al. 2025. Qwen2.5-VL Technical Report. arXiv:2502.13923.

Chen, L.; Wei, X.; Li, J.; Dong, X.; Zhang, P.; Zang, Y.; Chen, Z.; Duan, H.; Lin, B.; Tang, Z.; Yuan, L.; Qiao, Y.; Lin, D.; Zhao, F.; and Wang, J. 2024a. ShareGPT4Video: Improving Video Understanding and Generation with Better Captions. arXiv:2406.04325.

Chen, L.; Zhao, H.; Liu, T.; Bai, S.; Lin, J.; Zhou, C.; and Chang, B. 2024b. An Image is Worth 1/2 Tokens After Layer 2: Plug-and-Play Inference Acceleration for Large Vision-Language Models. arXiv:2403.06764.

Chiang, W.-L.; Li, Z.; Lin, Z.; Sheng, Y.; Wu, Z.; Zhang, H.; Zheng, L.; Zhuang, S.; Zhuang, Y.; Gonzalez, J. E.; Stoica, I.; and Xing, E. P. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality.

Dao, T. 2023. FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. arXiv:2307.08691.

Dao, T.; Fu, D. Y.; Ermon, S.; Rudra, A.; and Ré, C. 2022. FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. arXiv:2205.14135.

Devoto, A.; Zhao, Y.; Scardapane, S.; and Minervini, P. 2024. A Simple and Effective L_2 Norm-Based Strategy for KV Cache Compression. *arXiv preprint arXiv:2406.11430*.

Fu, C.; Chen, P.; Shen, Y.; Qin, Y.; Zhang, M.; Lin, X.; Yang, J.; Zheng, X.; Li, K.; Sun, X.; Wu, Y.; and Ji, R. 2024. MME: A Comprehensive Evaluation Benchmark for Multimodal Large Language Models. arXiv:2306.13394.

Fu, C.; Dai, Y.; Luo, Y.; Li, L.; Ren, S.; Zhang, R.; Wang, Z.; Zhou, C.; Shen, Y.; Zhang, M.; Chen, P.; Li, Y.; Lin, S.; Zhao, S.; Li, K.; Xu, T.; Zheng, X.; Chen, E.; Shan, C.; He, R.; and Sun, X. 2025. Video-MME: The First-Ever Comprehensive Evaluation Benchmark of Multi-modal LLMs in Video Analysis. arXiv:2405.21075.

Goyal, Y.; Khot, T.; Summers-Stay, D.; Batra, D.; and Parikh, D. 2017. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6325–6334.

Hudson, D. A.; and Manning, C. D. 2019. 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*, 6700–6709.

Kim, M. J.; Pertsch, K.; Karamcheti, S.; Xiao, T.; Balakrishna, A.; Nair, S.; Rafailov, R.; Foster, E.; Lam, G.; Sanketi, P.; et al. 2024. OpenVLA: An Open-Source Vision-Language-Action Model. arXiv:2406.09246.

Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023a. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. arXiv:2301.12597.

Li, K.; Wang, Y.; He, Y.; Li, Y.; Wang, Y.; Liu, Y.; Wang, Z.; Xu, J.; Chen, G.; Luo, P.; Wang, L.; and Qiao, Y. 2024. MVBench: A Comprehensive Multi-modal Video Understanding Benchmark. arXiv:2311.17005.

Li, Y.; Du, Y.; Zhou, K.; Wang, J.; Zhao, W. X.; and Wen, J.-R. 2023b. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*.

Lin, B.; Ye, Y.; Zhu, B.; Cui, J.; Ning, M.; Jin, P.; and Yuan, L. 2024. Video-LLaVA: Learning United Visual Representation by Alignment Before Projection. arXiv:2311.10122.

Lin, T.-Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C. L.; and Dollár, P. 2015. Microsoft COCO: Common Objects in Context. arXiv:1405.0312.

Liu, H.; Li, C.; Li, Y.; Li, B.; Zhang, Y.; Shen, S.; and Lee, Y. J. 2024a. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023. Visual Instruction Tuning. arXiv:2304.08485.

Liu, J.; Liu, M.; Wang, Z.; An, P.; Li, X.; Zhou, K.; Yang, S.; Zhang, R.; Guo, Y.; and Zhang, S. 2024b. RoboMamba: Efficient Vision-Language-Action Model for Robotic Reasoning and Manipulation. arXiv:2406.04339.

Liu, Y.; Duan, H.; Zhang, Y.; Li, B.; Zhang, S.; Zhao, W.; Yuan, Y.; Wang, J.; He, C.; Liu, Z.; Chen, K.; and Lin, D. 2024c. MMBench: Is Your Multi-modal Model an All-around Player? arXiv:2307.06281.

Lu, P.; Mishra, S.; Xia, T.; Qiu, L.; Chang, K.-W.; Zhu, S.-C.; Tafjord, O.; Clark, P.; and Kalyan, A. 2022. Learn to Explain: Multimodal Reasoning via Thought Chains for Science Question Answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*.

Maaz, M.; Rasheed, H.; Khan, S.; and Khan, F. S. 2024. Video-ChatGPT: Towards Detailed Video Understanding via Large Vision and Language Models. arXiv:2306.05424.

Mangalam, K.; Akshulakov, R.; and Malik, J. 2023. EgoSchema: A Diagnostic Benchmark for Very Long-form Video Language Understanding. arXiv:2308.09126.

Marino, K.; Rastegari, M.; Farhadi, A.; and Mottaghi, R. 2019. OK-VQA: A Visual Question Answering Benchmark Requiring External Knowledge. arXiv:1906.00067.

Neo, C.; Ong, L.; Torr, P.; Geva, M.; Krueger, D.; and Barez, F. 2025. Towards Interpreting Visual Information Processing in Vision-Language Models. arXiv:2410.07149.

Parekh, J.; Khayatan, P.; Shukor, M.; Newson, A.; and Cord, M. 2024. A Concept-Based Explainability Framework for Large Multimodal Models. arXiv:2406.08074.

Qu, G.; Chen, Q.; Wei, W.; Lin, Z.; Chen, X.; and Huang, K. 2025. Mobile Edge Intelligence for Large Language Models: A Contemporary Survey. arXiv:2407.18921.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. arXiv:2103.00020.

Shang, Y.; Cai, M.; Xu, B.; Lee, Y. J.; and Yan, Y. 2024. LLaVA-PruMerge: Adaptive Token Reduction for Efficient Large Multimodal Models. arXiv:2403.15388.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; Rodriguez, A.; Joulin, A.; Grave, E.; and Lample, G. 2023. LLaMA: Open and Efficient Foundation Language Models. *ArXiv*, abs/2302.13971.

Wu, H.; Zhang, Z.; Zhang, E.; Chen, C.; Liao, L.; Wang, A.; Li, C.; Sun, W.; Yan, Q.; Zhai, G.; and Lin, W. 2024. Q-Bench: A Benchmark for General-Purpose Foundation Models on Low-level Vision. arXiv:2309.14181.

Xiao, J.; Shang, X.; Yao, A.; and Chua, T.-S. 2021. NExT-QA:Next Phase of Question-Answering to Explaining Temporal Actions. arXiv:2105.08276.

Xing, L.; Huang, Q.; Dong, X.; Lu, J.; Zhang, P.; Zang, Y.; Cao, Y.; He, C.; Wang, J.; Wu, F.; and Lin, D. 2025. PyramidDrop: Accelerating Your Large Vision-Language Models via Pyramid Visual Redundancy Reduction. arXiv:2410.17247.

Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; Lin, H.; et al. 2025. Qwen2.5 Technical Report. arXiv:2412.15115.

Yang, S.; Chen, Y.; Tian, Z.; Wang, C.; Li, J.; Yu, B.; and Jia, J. 2024a. VisionZip: Longer is Better but Not Necessary in Vision Language Models. arXiv:2412.04467.

Yang, S.; Tian, Z.; Jiang, L.; and Jia, J. 2024b. Unified Language-driven Zero-shot Domain Adaptation. arXiv:2404.07155.

Yao, Y.; Yu, T.; Zhang, A.; Wang, C.; Cui, J.; Zhu, H.; Cai, T.; Li, H.; Zhao, W.; He, Z.; Chen, Q.; Zhou, H.; et al. 2024. MiniCPM-V: A GPT-4V Level MLLM on Your Phone. arXiv:2408.01800.

Young, P.; Lai, A.; Hodosh, M.; and Hockenmaier, J. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2: 67–78.

Zhang, K.; Li, B.; Zhang, P.; Pu, F.; Cahyono, J. A.; Hu, K.; Liu, S.; Zhang, Y.; Yang, J.; Li, C.; and Liu, Z. 2024a. LMMs-Eval: Reality Check on the Evaluation of Large Multimodal Models. arXiv:2407.12772.

Zhang, P.; Dong, X.; Zang, Y.; Cao, Y.; Qian, R.; Chen, L.; Guo, Q.; Duan, H.; Wang, B.; Ouyang, L.; Zhang, S.; et al. 2024b. InternLM-XComposer-2.5: A Versatile Large Vision Language Model Supporting Long-Contextual Input and Output. arXiv:2407.03320.

Zhang, X.; Quan, Y.; Shen, C.; Yuan, X.; Yan, S.; Xie, L.; Wang, W.; Gu, C.; Tang, H.; and Ye, J. 2024c. From Redundancy to Relevance: Information Flow in LVLMs Across Reasoning Tasks. arXiv:2406.06579.

Zhang, Y.; Fan, C.-K.; Ma, J.; Zheng, W.; Huang, T.; Cheng, K.; Gudovskiy, D.; Okuno, T.; Nakata, Y.; Keutzer, K.; and Zhang, S. 2025. SparseVLM: Visual Token Sparsification for Efficient Vision-Language Model Inference. arXiv:2410.04417.

Zhang, Y.; Li, B.; Liu, h.; Lee, Y. j.; Gui, L.; Fu, D.; Feng, J.; Liu, Z.; and Li, C. 2024d. LLaVA-NeXT: A Strong Zero-shot Video Understanding Model.