Libra-Merging: Importance-redundancy and Pruning-merging Trade-off for Acceleration Plug-in in Large Vision-Language Model

Longrong Yang[1]*, Dong Shen[2]*, Chaoxiang Cai[3], Kaibing Chen[2],
Fan Yang[2], Tingting Gao[2], Di Zhang[2], Xi Li[1]†
[1]College of Computer Science and Technology, Zhejiang University
[2]Kuaishou Technology
[3]School of Software Technology, Zhejiang University

Abstract

Large Vision-Language Models (LVLMs) have achieved significant progress in recent years. However, the expensive inference cost limits the realistic deployment of LVLMs. Some works find that visual tokens are redundant and compress tokens to reduce the inference cost. These works identify important non-redundant tokens as target tokens, then prune the remaining tokens (non-target tokens) or merge them into target tokens. However, target token identification faces the token importance-redundancy dilemma. Besides, token merging and pruning face a dilemma between disrupting target token information and losing non-target token information. To solve these problems, we propose a novel visual token compression scheme, named Libra-Merging. In target token identification, Libra-Merging selects the most important tokens from spatially discrete intervals, achieving a more robust token importance-redundancy trade-off than relying on a hyper-parameter. In token compression, when non-target tokens are dissimilar to target tokens, Libra-Merging does not merge them into the target tokens, thus avoiding disrupting target token information. Meanwhile, Libra-Merging condenses these non-target tokens into an information compensation token to prevent losing important non-target token information. Our method can serve as a plug-in for diverse LVLMs, and extensive experimental results demonstrate its effectiveness. The code will be publicly available at `https://github.com/longrongyang/Libra-Merging`.

1. Introduction

Large Vision-Language Models (LVLMs) have recently shown considerable progress by incorporating visual processing capabilities into Large Language Models (LLMs). Numerous recent LVLMs [2, 8, 58–60] indicate that both large model size and extensive dataset size are crucial for enhancing intelligence. Even with sufficiently large model sizes, these models demonstrate "Emergent Abilities". As a result, a range of studies [12, 26, 32] have increased the model size of LVLMs, resulting in the state-of-the-art performance across a variety of tasks.

Although LVLMs achieve state-of-the-art performance, they are expensive to employ in realistic applications due to significant inference costs. To reduce inference costs, some recent works [9, 23] propose compressing visual tokens, based on the finding that visual tokens are highly redundant in LVLMs. For example, FastV [9] treats tokens as important when they have high output attention (attention score of output token on visual tokens), and prunes the bottom $R\%$ tokens with the lowest output attention. Turbo [23] computes the information degree of each token, a weighted average of attention and redundancy (the highest similarity between the token and other tokens), and averagely merges the top $R\%$ tokens into their most similar tokens.

In LVLMs, the token compression process can be modeled as identifying important non-redundant tokens (named target tokens) and compressing the remaining tokens (named non-target tokens). However, target token identification and token compression face a dilemma respectively. The dilemma in target token identification lies in the importance-redundancy trade-off. FastV selects the most important tokens, but these tokens may be redundant; Turbo balances importance and redundancy with a hyper-parameter, but the fixed hyper-parameter is hard to apply universally to diverse scenarios. As shown in Figure 1 (a), they still preserve redundant information, leading to the losing of some important tokens. As shown in Figure 1 (b), the dilemma in token compression lies in the pruning-merging trade-off. Pruning non-target tokens loses information in non-target tokens; averagely merging non-target and target tokens may disrupt information in target tokens.

*The authors contributed equally to this paper.
†Corresponding author.

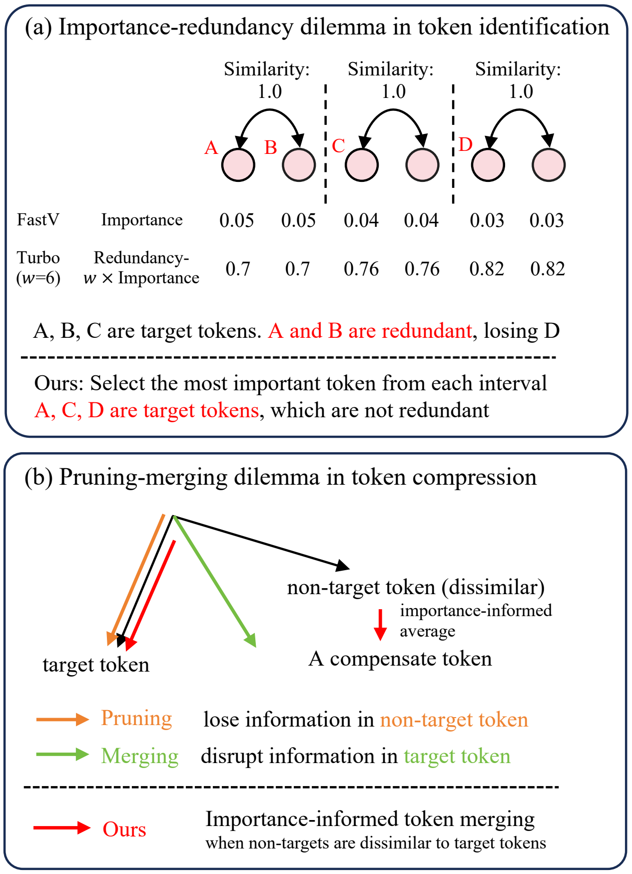

Figure 1. (a) Target Token identification. We select the most important token in each interval, avoiding selecting redundant tokens by disrupting the spatial continuity of target tokens, for achieving importance-redundancy trade-off. (b) Token compression. Similar tokens are merged based on their importance. When non-target tokens are dissimilar to target tokens, we condense these non-target tokens into a compensation token, for achieving pruning-merging trade-off. **In this work, we follow FastV [9] to treat tokens as important when they have high output attention.**

Facing the two dilemmas, this work proposes a novel training-free token compression scheme Libra-Merging, which consists of two key modules: (i) **Position-driven token identification**. *Adjacent tokens are usually redundant* [15, 27]. Thus, we discretize the visual token sequence into different intervals, and within each interval, we select the most important token as the target token. As shown in Fig. 1 (a), we avoid selecting redundant tokens by disrupting the spatial continuity of target tokens. This technique achieves a more robust importance-redundancy trade-off than using a hyper-parameter. (ii) **Importance-informed grouped merging**. *We find that token merging is harmful when dissimilar tokens merge*. Thus, we categorize non-target tokens into *positive set* (high similarity) and *negative set* (low similarity) based on their similarities with target tokens. Tokens in *positive set* directly merge into target tokens, and we use Softmax to convert token importance to merging weight. To avoid disrupting target tokens, tokens in *negative set* should not merge into target tokens. However, these tokens may still contain important information, so we compute their importance-informed average as an information compensation token. We achieve a good pruning-merging trade-off by striving to preserve important information. **Overall**, for a visual token sequence, Libra-Merging first uses position-driven token identification to divide it into target and non-target tokens. Then, importance-informed grouped merging merges non-target tokens into target tokens, resulting in an output token sequence.

In addition, since flash-attention 2 does not output attention scores, we design a hybrid attention mechanism. With the hybrid attention mechanism, we can further employ token compression to accelerate the training of LVLMs.

Our contributions are summarized as:

(i) We propose position-driven token identification to avoid selecting redundant target tokens, which achieves a robust importance-redundancy trade-off in target token identification.

(ii) We propose importance-informed grouped merging to prevent disrupting target tokens while preserving important information in non-target tokens, which achieves a good pruning-merging trade-off in token compression.

(iii) As a plug-in, our method can be easily integrated into LVLMs, including image-text models and video-text models. Extensive experiments on different models have validated that our method Libra-Merging achieves significantly higher performance than existing works.

2. Related Works

2.1. Large Vision-Language Model

Large Language Models (LLMs) have shown remarkable proficiency in following instructions and generalizing across various tasks. To extend these capabilities to incorporate visual information, Large Vision-Language Models (LVLMs) like GPT-4 and LLaVA employ frozen visual encoders and trainable visual projectors. These models typically transform visual data into visual tokens, which are then used to condition the adaptation of text tokens within LLMs [1, 36, 37, 42–46, 50, 51, 53–56]. Recent advancements in LVLMs have primarily focused on two aspects. The first involves optimizing training strategies, *e.g.*, [2, 6]. The second, and more prevalent, focuses on enhancing visual components. This includes expanding datasets [31, 59], improving image encoders [2, 11], and aligning the input and projection layers [5, 12, 29, 57, 60]. These efforts, particularly the expansion of visual instruction-tuning datasets and the scaling up of model sizes, have significantly boosted the visual understanding capabilities of LVLMs.

2.2. Token Compression

There have been studies on improving the efficiency of Vision-Language Models (VLMs) before the era of Large Vision-Language Models (LVLMs). A majority of them focus on token compression for vision transformers (ViTs). Token compression for ViTs can be roughly categorized as token pruning [4, 10, 14, 18, 25, 28, 38, 47] and token merging [3, 4, 10, 13, 15, 17, 19, 24, 27, 35, 39, 49, 61]. EViT [28] extends Top-K pruning by creating a "fused" token. DynamicViT [38] prunes tokens by keep probabilities produced by a small prediction module. ATS [14] and DiffRate [10] design the dynamic compression ratio for toke compression. ToMe [3], K-Medoids [35], and DPC-KNN [13] are token merging methods based on similarity, cluster, and density, respectively. SiT [61], Sinkhorn [17], PatchMerger [39] focus on soft merging. More recently, PYRA [52] has enhanced the training and inference of ViTs via a specialized token merging technique. FastV [9] and IVTP [20] are the initial two works to explore visual token compression for Large Vision-Language Models (LVLMs), which uses language as an interface for various vision-language tasks. ToCom [22] discusses the multi-step token compression. HOMER [41] proposes the use of token compression to enlarge the context length. LLaVolta [7] proposes a training token compression scheme. Turbo [23] is a token merging method designed for LVLMs, which uses information degree to sort tokens and merge tokens with the high information degree. Pruning loses important information in non-target tokens, while merging may disrupt target tokens when dissimilar tokens merge. The proposed importance-informed grouped merging is used to prevent disrupting target tokens while preserving important information in non-target tokens.

3. Methodology

3.1. Overview

Large Vision-Language Model (LVLM): A LVLM is usually designed to integrate a visual model into the pre-trained LLM. Specifically, the input of the vision model is an image or a video, and its output is a visual token sequence $\mathcal{Z} = [z_1, z_2, \cdots, z_N] \in \mathbb{R}^{N\times C}$, where N is the sequence length of visual tokens. Then, a visual projection layer maps $\mathcal{Z} \in \mathbb{R}^{N\times C}$ to $\mathcal{V} \in \mathbb{R}^{N\times D}$, where D represents the hidden size. Besides, the instruction text is projected as instruction text tokens $\mathcal{T} = [t_1, t_2, \cdots, t_P] \in \mathbb{R}^{P\times D}$, where N represents the sequence length of text tokens. LLM consists of stacked multi-head self-attention (MSA) and feed-forward neural networks (FFN), with layer normalization (LN) and residual connections:

$$\mathbf{x}_0 = [v_1, v_2, \cdots, v_N, \cdots, t_1, t_2, \cdots, t_P], \tag{1}$$

$$\mathbf{x}'_\ell = \mathrm{MSA}(\mathrm{LN}(\mathbf{x}_{\ell-1})) + \mathbf{x}_{\ell-1}, \ell \in \{1, \dots, L\}, \tag{2}$$

$$\mathbf{x}_\ell = \mathrm{FFN}(\mathrm{LN}(\mathbf{x'}_\ell)) + \mathbf{x'}_\ell, \ell \in \{1, \dots, L\}, \tag{3}$$

where L is the layer number of LLM.

Token Compression: Given an input visual token sequence $\mathcal{V}$=$\{v_1, v_2, \cdots, v_N\}$. The token compression usually consists of two steps: (i) Target token identification. Important non-redundant tokens should be selected as target tokens, forming a target set T=$\{v^t_1, \cdots, v^t_{N_t}\}$. The remaining tokens are named as non-target tokens, forming a non-target set S=$\{v^s_1, \cdots, v^s_{N_s}\}$. Suppose the compression ratio is R. N_t=(1-R)·N. N_s=R·N. (ii) Token compression. FastV [9] is a token pruning technique, which directly prunes non-target tokens, resulting in the final token sequence $\mathcal{V}'$=$\{v'_1, \cdots, v'_{N_t}\}$, where $v'_i = v^t_i$. Turbo [9] is a token merging technique, which merges non-target tokens into target tokens, resulting in the final token sequence $\mathcal{V}'$=$\{v'_1, \cdots, v'_{N_t}\}$, where $v'_i = \frac{v^t_i + v^s_i}{2}$.

Libra-Merging: In this work, we follow FastV [9] to treat tokens as important when they have high output attention (response-related tokens). The importance metric α is defined as the attention score of output token on visual input tokens during the decoding process of one response:

$$\alpha = \mathrm{softmax}(\frac{\mathbf{Q}_{\mathrm{output}}\,\mathbf{K}^T}{\sqrt{D}}) \in \mathbb{R}^{1\times N_{all}}, \quad \sum_{i=1}^{N_{all}} \alpha_{1,i} = 1, \tag{4}$$

where $\mathbf{Q}_{\mathrm{output}}$ refers to the output token, $\mathbf{K}$ refers to all tokens, and N_{all}=N+P. For convenience, we denote $\alpha_{1,i}$ as α_i. As shown in Figure 2, given a sequence of visual tokens and the importance α of each token, we first use position-driven token identification to identify important and non-redundant tokens as target tokens. The remaining tokens serve as non-target tokens. Then, when non-target tokens are similar to target tokens, we merge non-target tokens into target tokens, where merging weighting results from token importance. When non-target tokens are dissimilar to target tokens, we compute their importance-informed average as an information compensation token. Furthermore, we select different layers and perform the above two operations on each layer to achieve higher token compression efficiency. The details of these modules will be introduced in the subsequent sections.

3.2. Position-driven Token Identification

Generally, important and non-redundant tokens should be selected as target tokens. However, it is challenging to achieve a robust importance-redundancy trade-off during selecting tokens. In this work, we assume that the redundancy of visual tokens is mainly reflected in spatially contiguous visual tokens. Thus, as shown in Figure 2, we divide the visual token sequence into several intervals and select important tokens from different intervals, to disrupt the spatial continuity of target tokens. This technique achieves

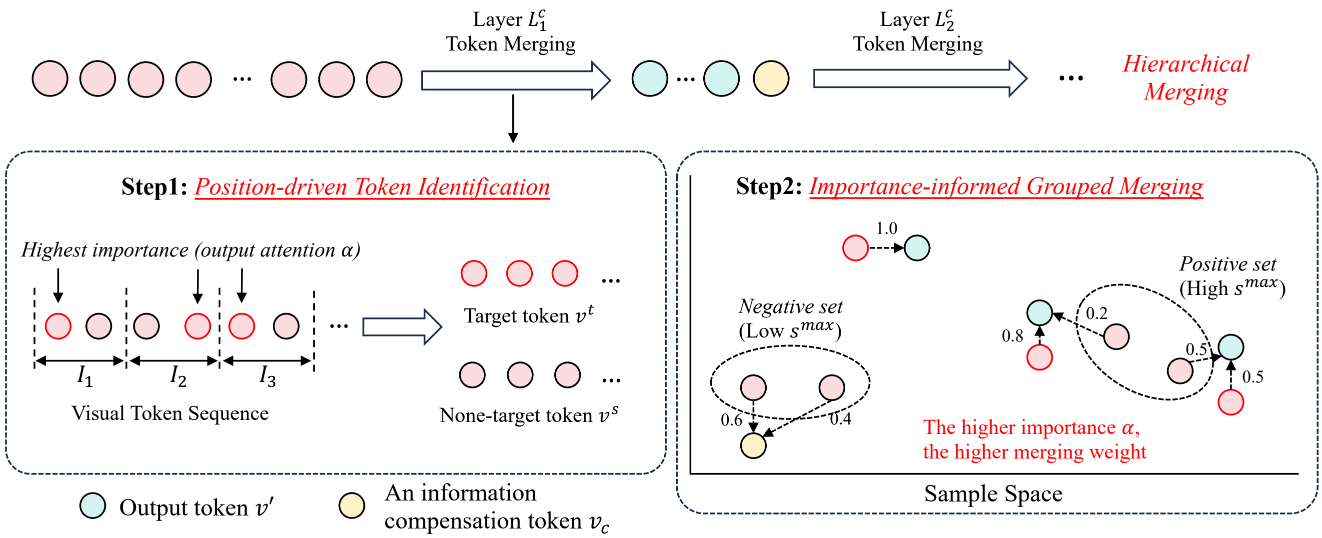

Figure 2. The pipeline of Libra-Merging. *Hierarchical Merging* indicates that we perform token merging operations at different layers. The token merging per layer consists of two steps: (i) *Position-driven Token Identification*. We divide the input visual token sequence into multiple intervals, and within each interval, we select the token with the highest importance as the target token. The remaining tokens serve as non-target tokens. (ii) *Importance-informed Grouped Merging*. We divide non-target tokens into *positive set* (high similarity) and *negative set* (low similarity) based on their similarities s^{max} with target tokens. To prevent disrupting target tokens, tokens in *negative set* do not merge into target tokens. To preserve important information in *negative set* tokens, we merge these tokens as an information compensation token rather than discarding them. All merging weights are generated from token importance.

a robust importance-redundancy trade-off based on a common property of visual tokens rather than a manually designed fixed hyperparameter.

Specifically, (i) we evenly divide the visual token sequence into several intervals. To avoid selecting adjacent tokens, we only select one token from each interval. Thus, the number of intervals should be N_t. Given a compression ratio R, we calculate the length of the visual token sequence $l_{interval}$ in each interval as $l_{interval} = \frac{1}{1-R}$. We have different intervals I=$\{I_1, \cdots, I_{N_t}\}$. For example, when R=75%, the length of the visual token sequence in each interval is 4. This means that every four tokens form an interval, and one token is selected from every four tokens as the target token. (ii) After dividing the intervals, we extract the token with the highest importance from each interval as the target token. For example, for the interval I_1=$\{v_1, \cdots, v_l\}$, we select v_n as v_1^t when $n = \text{argmax}_{n=\{1,\cdots,l\}}(\alpha_n)$.

3.3. Importance-informed Grouped Merging

Token pruning and merging face a dilemma. Directly pruning non-target tokens loses information in non-target tokens. Token merging prevents losing information in non-target tokens by merging non-target tokens into target tokens. However, it may disrupt important information in target tokens. When an especially important target token merges with a dissimilar non-target token, the result can be catastrophic. Therefore, as shown in Figure 2, we propose importance-informed grouped merging to achieve the pruning-merging trade-off.

Firstly, we find the most similar token of each token and group tokens in two sets. Given target tokens $T = \{v_1^t, \cdots, v_{N_t}^t\}$, we compute the cosine similarities between target tokens and all tokens $\mathcal{V} = \{v_1, v_2, \cdots, v_N\}$, forming $S \in R^{N_t \times N}$. Then, we identify the most similar target token of each token and create a token matching matrix M:

$$M_{ij} = \mathbb{I}(S_{ij} = \max_{k=1}^{N_t} S_{kj}), \tag{5}$$

where $\mathbb{I}(\cdot)$ is the indicator function, which is 1 if the condition inside the parentheses is true, and 0 otherwise. When the maximum value of a column j in S is located at row i, the corresponding element M_{ij} is set 1, and all other elements in that column of M are set to 0. The highest similarity between one token and target tokens is masked as s^{max}. During merging, we follow two principles: (i) merging similar tokens; and (ii) preserving important information. Some non-target tokens have a low s^{max} with target tokens, directly merging these tokens into target tokens may disrupt target tokens. However, these non-target tokens may still contain important information with relatively high output attention scores. To address this issue, we divide non-target tokens into two groups: when tokens satisfy $s^{max} > \tau$, they are grouped in *positive set*; the remaining tokens are grouped in *negative set*.

Then, the proposed importance-informed grouped merging operates on two dimensions: (i) Tokens in the *positive set* are similar to target tokens, so we directly merge them into target tokens. During the merging, we use importance α_i to generate the merging weight to avoid introducing unimportant noise information. Specifically, we first replace zero elements in the matching matrix with -inf and non-zero elements with importance.

$$M'_{ij} = -\text{inf} \cdot (1 - M_{ij}) + \alpha_j * M_{ij}. \tag{6}$$

For each M_{ij} in M, when M_{ij}=0, M'_{ij}=-inf; when M_{ij}=1, M'_{ij}=α_j. Then, we use Softmax to achieve the importance-informed merging weighting modeling in each row $M'_{i\cdot}$: $W_{i\cdot}$=Softmax($\frac{M'_{i\cdot}}{\eta}$), where η is a temperature coefficient. Finally, we use W to merge tokens into the final visual token sequence $\mathcal{V}'$=$\{v'_1, \cdots, v'_{N_t}\}$:

$$\mathcal{V}' = W \times \mathcal{V}, \tag{7}$$

where $W \in R^{N_t \times N}$ is the importance-informed merging matrix, $\mathcal{V} \in R^{N \times D}$ is the input visual token sequence, and $\mathcal{V}' \in R^{N_t \times D}$ is the output visual token sequence. (ii) To avoid disrupting target tokens, tokens in the *negative set* cannot merge into target tokens. However, to prevent losing important information, we cannot simply discard them. Thus, we propose to compute their importance-informed average as an information compensation token v_c, which is appended after the token sequence $\mathcal{V}'$. $N_t \gg$1, so the increased Flops can be negligible. The computation of the average weighting is consistent with the computation of W.

Our method achieves a good pruning-merging trade-off by striving to preserve important information not only in target tokens but also in non-target tokens.

3.4. Hierarchical Merging

When the model layer is deeper, the visual token redundancy is higher [61]. Thus, if token compression is only applied in the shallow layers, the tokens in the deep layers remain redundant. To further improve the compression efficiency, we introduce a hierarchical token merging trick to reduce the visual token count layer by layer.

Suppose the layers for visual token compression as $\{L^c_1, L^c_2, \cdots, L^c_K\}$. As shown in Figure 2, we perform the aforementioned two operations at each layer L^c_k, *i.e.*, we first use position-driven token identification to divide the visual token sequence into target tokens and non-target tokens; then, we use importance-informed grouped merging to merge non-target tokens into target tokens, resulting in the final token sequence. Suppose the compression ratio per layer is R and the total count of visual tokens is N. The visual token count in the layer l is N_l=$N \cdot R^k$ when $L^c_k < l \leq L^c_{k+1}$ ($k \in \{1, \cdots K-1\}$). When $l \leq L^c_1$, the visual token count in the layer l is N_l=N. When $l > L^c_K$, the visual token count in the layer l is N_l=$N \cdot R^K$.

4. Experiments

4.1. Experimental Setup

Models: We verify the effectiveness of Libra-Merging by experiments on LVLMs with different model sizes and input resolution. In detail, we study LLaVA-1.5-7B [32], LLaVA-1.5-13B [32], and LLaVA-NeXT-8B [30]. LLaVA-1.5 is the most widely used open-source LVLM for research. The visual token sequence length in LLaVA-Next is 576. LLaVA-Next is the extension of LLaVA-1.5, which supports the dynamic input resolution. The visual token sequence length in LLaVA-Next is dynamic. We also conduct experiments on Qwen2-VL [48] in the supplementary material.

Benchmarks: We follow LLaVA-1.5 [32] to evaluate the performance of LVLMs. Specifically, GQA [21] evaluates the visual perception capabilities of models through open-ended short answers. ScienceQA [34], a multiple-choice benchmark, evaluates the zero-shot generalization of models on scientific question answering. TextVQA [40] focuses on text-rich visual question answering tasks. MME [16] assesses the visual perception of models with yes/no questions. MMBench [33] evaluates the robustness of model answers with all-round shuffling on multiple choice answers.

Efficiency evaluation: In this paper, to evaluate the inference efficiency, we follow FastV [9] and mainly report the Flops of the image token part. For example, when conducting experiments on LLaVA-1.5-7B, the Flops of MSA and FFN is defined as $4nd^2 + 2n^2d + 3ndm$, where n is the visual token count, d is the hidden state size, and m is the intermediate size of the FFN. When the visual token count is n' in the layer l, the Flops for the layer l is computed as $4n'd^2 + 2(n')^2d + 3n'dm$. The reported Flops is the sum of Flops for all layers.

Baselines and Libra-Merging: Our main baselines are FastV [9] and Turbo [23]. FastV [9] is a token pruning technique that prunes the R% tokens with the lowest output attention at the layer K. Turbo [23] is a token merging technique that merges the R% tokens at the layer K, where a little difference is that we do not use bipartite soft matching because bipartite soft matching may fail to find the most similar token of a token. We set K=3 and R=50. The proposed method is Libra-Merging. Libra-Merging compresses visual tokens in multiple layers. In each compression layer, Libra-Merging uses position-driven token identification to identify target tokens. Then, it uses importance-informed grouped merging to merge non-target tokens. We set the compression ratio as $\{50\%, 67\%, 80\%\}$. The compression layers are $\{7, 15, 23\}$ for LLaVA-1.5-7B (32 layers, $\{0, 1, \cdots, 31\}$) and LLaVA-NeXT-8B (32 layers). The compression layers are $\{9, 19, 29\}$ for LLaVA-1.5-13B (40 layers). In Libra-Merging, τ = 0.7 and η = mean(α). More details please refer to the supplementary material.

Table 1. LVLMs (image-text models) with different token compression methods on six benchmarks. We conduct experiments on three different LVLMs to verify the scalability of our method across different model sizes (7b *vs.* 13b) and visual token count (llava-1.5 *vs.* llava-next). The Flops ratio 47% (37%) corresponds to compression ratio 50% (67%). T means trillion. Experiments about more datasets please refer to the supplementary material.

Model		Flops (T)	Ratio	GQA	SQA^I	MME	MMB	MMB^{CN}	TextVQA	Avg
LLaVA-1.5-7B	*vanilla*	3.82	100%	62.0	69.5	1512.0	64.7	58.2	58.2	62.5
	FastV	2.13	56%	60.4	68.8	1511.7	64.2	58.0	57.6	61.8
	Turbo	2.13	56%	61.6	68.7	1471.7	63.7	57.5	57.4	61.8
	Libra-Merging	1.78	47%	61.3	68.9	1502.5	64.3	58.5	57.4	62.1
	Libra-Merging	1.41	37%	60.7	69.2	1480.1	63.9	58.2	57.4	61.9
LLaVA-1.5-13B	*vanilla*	7.44	100%	63.2	72.8	1531.3	68.5	63.6	61.2	65.9
	FastV	4.06	55%	62.7	73.0	1549.8	68.3	63.5	60.8	65.7
	Turbo	4.06	55%	62.8	72.7	1561.0	68.1	63.2	60.7	65.5
	Libra-Merging	3.47	47%	63.3	73.1	1531.1	68.4	63.7	61.1	65.9
	Libra-Merging	2.74	37%	62.5	72.4	1512.3	68.4	63.1	60.4	65.4
LLaVA-NeXT-8B	*vanilla*	17.17	100%	65.9	77.3	1552.1	74.4	70.4	69.8	71.6
	FastV	9.36	55%	65.5	77.2	1572.6	74.5	70.6	69.5	71.5
	Turbo	9.36	55%	64.7	77.7	1505.3	73.4	69.1	65.0	70.0
	Libra-Merging	7.86	47%	65.7	77.6	1565.8	74.7	70.8	69.7	71.7
	Libra-Merging	6.24	37%	65.6	77.2	1565.7	73.9	70.2	69.4	71.3

4.2. Image Understanding Evaluation

We study the inference efficiency on six image-text benchmarks. As shown in Table 1, on LLaVA-1.5-7B, Libra-Merging reduces the inference Flops from 3.82 to 1.78, while maintaining a comparable average performance. Compared to FastV and Turbo, Libra-Merging has a higher average performance and lower Flops. The conclusion is scalable on LLaVA-1.5-13B and LLaVA-NeXT-8B. An exciting phenomenon is that the stronger the LVLM, the more effective the token compression becomes. The average performance of using 47% tokens even surpasses that of using 100% tokens on LLaVA-NeXT-8B (71.7 *vs.* 71.6).

4.3. Video Understanding Evaluation

We extend Libra-Merging to VideoLLaMA-2 [29] and the results in Table 2 reveal that the proposed inference scheme significantly surpasses FastV and Turbo.

4.4. Ablation Study

Component ablation: In this study, we mainly discuss: (i) Merging. Refers to Turbo [23], merging tokens in one layer (the layer 3). (ii) Hierarchical merging, merging tokens in multiple layers (the layers $\{7, 15, 23\}$). (iii) Position-driven Token Identification (PTI), aiming to achieve the importance-redundancy trade-off in target token identification. (iv) Importance-informed Grouped Merging (IGM), aiming to achieve the pruning-merging trade-off in target token identification.

As shown in the Table 3, the experimental results verify: (i) Hierarchical merging works. For example, when R=50%, compared to merging, hierarchical merging improves the performance from 62.9 to 63.1. (ii) Both PTI and IGM contribute to the average performance increase. For example, when R=67%, PTI improves the performance from 62.5 to 62.7, and IGM improves the performance from 62.7 to 63.0. (iii) PTI+IGM brings a more significant average performance increase under a larger compression ratio. For example, when R=67%, PTI+IGM improves the performance by 0.5%. When R=80%, PTI+IGM improves the performance by 0.7%. A possible reason is that when the compression ratio is larger, selecting redundant tokens is more likely to discard important tokens and more non-redundant tokens disrupt target tokens during merging.

Study about pruning-merging trade-off: The premise of token merging is that tokens should be similar. When a non-target token is dissimilar to a target token, token pruning should be preferred, as token merging disrupts the target token; on the contrary, token merging should be preferred. We explore the pruning-merging trade-off from the perspective of different layers. Specifically, we select layers $\{3, 15\}$ to conduct the experiments.

As shown in Table 4, *Pruning* refers to pruning 75% visual tokens at layer 3 or layer 15, *Merging* refers to averaging non-target tokens with the most similar target token, and IGM refers to Importance-informed Grouped Merging. To focus on the token compression study, we only use importance to identify target tokens for all experiments in Ta-

Table 2. LVLMs (video-text models) with different token compression methods on Video-MME. VideoLLaMA-2 is the state-of-the-art Large Video-Language Model. We compress 75% visual tokens at layer 3. More details please refer to the supplementary material.

Model		Flops (T)	Ratio	Overall		Short		Medium		Long	
				w/o subs	w subs	w/o subs	w subs	w/o subs	w subs	w/o subs	w subs
VideoLLaMA-2 (7B)	*vanilla*	23.96	100%	49.8	54.7	58.0	63.6	47.0	53.1	44.3	47.3
	FastV	8.20	34%	46.5	51.4	52.1	57.3	45.2	50.4	42.1	46.5
	Turbo	8.20	34%	47.9	52.1	54.0	59.6	46.9	50.1	42.7	46.5
	Libra-Merging	8.20	34%	48.8	52.9	55.3	59.0	46.4	52.6	44.7	47.0

Table 3. Component ablation on five benchmarks. "Merging" refers to Turbo [23], merging tokens in one layer (3). "Hierarchical" refers to merging tokens in multiple layers ($\{7, 15, 23\}$). "PTI" refers to position-driven token identification. "IGM" refers to importance-informed grouped merging. R refers to the compression ratio.

	Merging	Hierarchical	PTI	IGM	R	Flops (T)	GQA	SQA^I	MME	MMB	MMB^{CN}	Avg
LLaVA-1.5-7B	✓				50%	2.13	61.6	68.7	1471.7	63.7	57.5	62.9
		✓			50%	1.78	61.2	69.3	1482.9	63.8	57.9	63.1
		✓	✓	✓	50%	1.78	61.3	68.9	1502.5	64.3	58.5	63.3
LLaVA-1.5-7B		✓			67%	1.41	60.0	69.4	1443.2	63.8	56.7	62.5
		✓	✓		67%	1.41	60.7	68.7	1487.8	63.4	57.8	62.7
		✓	✓	✓	67%	1.41	60.7	69.2	1480.1	63.9	58.2	63.0
LLaVA-1.5-7B		✓			80%	1.19	58.1	69.7	1419.9	62.6	55.6	61.5
		✓	✓		80%	1.19	58.7	69.6	1478.2	63.1	56.3	61.9
		✓	✓	✓	80%	1.19	59.3	69.7	1465.8	63.3	56.5	62.2

Table 4. Study about pruning-merging trade-off. Token importance is used to identify target tokens, and the only change is token compression technique. *Sim* indicates the mean of s^{max}.

	R	Layer	GQA	R	Layer	GQA
Pruning	80%	3	56.57	80%	15	61.78
Merging	80%	3	57.96	80%	15	61.73
IGM	80%	3	57.27	80%	15	61.82
Sim			0.8433			0.5955

Table 5. Study about different variants of IGM. IGM-pos means that all non-redundant tokens are used for merging (importance-informed). IGM-neg means that all non-redundant tokens are used for generating the information compensation token.

	R	Layer	GQA	MME	MMB
IGM-pos	80%	15	61.79	1506.3	64.79
IGM-neg	80%	15	61.77	1505.1	64.69
IGM	80%	15	61.82	1525.7	65.07

ble 4. *Sim* represents the mean of s^{max}; a smaller *Sim* means the lower mean similarity between target tokens and non-target tokens. Table 4 indicates: (i) When the *Sim* is low (high), token pruning is better than merging. When the *Sim* is high, token merging is better than pruning. (ii) Importance-informed Grouped Merging (IGM) achieves a good trade-off for different scenarios.

Study about importance-informed grouped merging: In importance-informed grouped merging, we want to preserve the important information in non-target tokens while avoiding the disruption of important information in target tokens due to merging. Thus, we divide non-target tokens into *positive set* and *negative set*. To verify the necessity of this division, we set: (i) IGM-pos, all non-target tokens merge into target tokens; (ii) IGM-neg, all non-target tokens merge as an information compensation token, which is similar to EViT [28]. As shown in Table 5, IGM outperforms both IGM-pos and IGM-neg.

Actual runtime latency and memory usage: To better evaluate inference efficiency, we conduct runtime and GPU memory usage analyses during inference, similar to FastV. We randomly select two datasets (GQA and MME) and perform inference using LLaVA-1.5-7B on an A800 GPU. We measure the end-to-end inference duration including reading/writing, and calculate "Latency/Example" to indicate the average inference time per sample. We comprehensively compare FastV and Libra-Merging. As shown in Tab. 6,

Table 6. Actual runtime latency and memory usage. We directly record the total inference time as "Time", which includes file reading/writing time. "Latency/Example" indicates the average inference time per sample.

	Model	R	Layer	Time (one A800)	Memory	Score	Latency/Example
GQA	LLaVA-1.5-7B	-	-	21:45	16.0G	61.95	0.104s
	+FastV	50%	3	19:36	15.6G	60.35	0.093s
	+Libra-Merging	50%	3	19:48	15.6G	61.38	0.094s
	+FastV	80%	3	17:54	15.4G	56.57	0.085s
	+Libra-Merging	80%	3	17:58	15.4G	58.81	0.086s
MME	LLaVA-1.5-7B	-	-	03:59	16.0G	1512.0	0.101s
	+FastV	50%	3	03:27	15.6G	1511.7	0.087s
	+Libra-Merging	50%	3	03:30	15.6G	1513.1	0.088s
	+FastV	80%	3	03:14	15.4G	1427.6	0.082s
	+Libra-Merging	80%	3	03:16	15.4G	1440.0	0.083s

Table 7. The flash-attention compatibility during inference. "hybrid" means the proposed hybrid attention, which is essential for maintaining flash-attention compatibility. The experiments about training are provided in the supplementary material.

	Model	R	Layer	Time (one A800)	Memory	Score	Latency/Example
MME	Qwen2-VL-7B	-	-	13:35	27.7G	1693.6	0.343s
	w flash-attention 2	-	-	08:48	18.5G	1683.6	0.222s
	+FastV	50%	3	12:25	28.6G	1673.9	0.314s
	+FastV *w* hybrid	50%	3	08:12	17.6G	1654.9	0.207s
	+Libra-Merging *w* hybrid	50%	3	08:19	17.6G	1690.3	0.210s
MME	LLaVA-1.5-7B	-	-	03:59	16.0G	1512.0	0.101s
	w flash-attention 2	-	-	03:58	16.0G	1507.5	0.100s
	+FastV	50%	3	03:27	15.6G	1511.7	0.087s
	+FastV *w* hybrid	50%	3	03:25	15.5G	1495.5	0.086s
	+Libra-Merging *w* hybrid	50%	3	03:26	15.5G	1502.1	0.087s

both methods achieve actual inference acceleration over the baseline. While demonstrating comparable efficiency to FastV, Libra-Merging delivers significantly superior performance, particularly when $R = 80\%$.

Flash-attention compatibility: Flash-attention is indispensable for accelerating attention computation, yet it does not output attention scores. Fortunately, our main goal is to preserve response-related visual information, which only requires attention scores between output token and visual tokens. Consequently, we compute only these attention scores, requiring merely $1 \times N_t$ score computations. We term this operation as hybrid attention. Since FLOPs scale quadratically with token length, this introduces approximately $\frac{1}{N_t}$ additional FLOPs, which becomes negligible when $N_t \gg 1$. As shown in Tab. 7, "Qwen2+flash-attention 2" outperforms "Qwen2+FastV" in speed. Then, with hybrid attention, token compression techniques become fully compatible with flash-attention 2.

With flash-attention compatibility, we successfully extend token compression techniques to model training. Experimental results are detailed in Table E of the supplementary material.

5. Conclusion

In this paper, we study the importance-redundancy dilemma and the pruning-merging dilemma in token compression for LVLMs. To solve the two dilemmas, we propose a novel token merging scheme Libra-Merging, which consists of position-driven token identification and importance-informed grouped compression. In target token identification, position-driven token identification is proposed to avoid selecting redundant target tokens, for a robust importance-redundancy trade-off. In token compression, importance-informed grouped compression is proposed to prevent disrupting target tokens while preserving important information in non-target tokens, for a good pruning-merging trade-off. Our method Libra-Merging acts as a plug-in, which can be easily integrated into existing LVLMs. Extensive experiments demonstrate the effectiveness of Libra-Merging across diverse datasets.

Acknowledgements. This work is supported in part by National Science Foundation for Distinguished Young Scholars under Grant 62225605, Zhejiang Provincial Natural Science Foundation of China under Grant LD24F020016, "Pioneer" and "Leading Goose" R&D Program of Zhejiang (No.2024C01020), Project 12326608 supported by NSFC, the Ningbo Science and Technology Innovation Project (No.2024Z294), and is supported by Kuaishou Technology.

References

[1] 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 preprint arXiv:2309.16609*, 2023. 2

[2] 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 preprint arXiv:2308.12966*, 2023. 1, 2

[3] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *International Conference on Learning Representations*, 2023. 3

[4] Qingqing Cao, Bhargavi Paranjape, and Hannaneh Hajishirzi. Pumer: Pruning and merging tokens for efficient vision language models, 2023. 3

[5] Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. Honeybee: Locality-enhanced projector for multimodal llm. *arXiv preprint arXiv:2312.06742*, 2023. 2

[6] Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*, 2023. 2

[7] Jieneng Chen, Luoxin Ye, Ju He, Zhao-Yang Wang, Daniel Khashabi, and Alan Yuille. Llavolta: Efficient multi-modal models via stage-wise visual context compression. *Advances in neural information processing systems*, 2024. 3

[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 preprint arXiv:2311.12793*, 2023. 1

[9] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pages 19–35. Springer, 2025. 1, 2, 3, 5

[10] Mengzhao Chen, Wenqi Shao, Peng Xu, Mingbao Lin, Kaipeng Zhang, Fei Chao, Rongrong Ji, Yu Qiao, and Ping Luo. Diffrate: Differentiable compression rate for efficient vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17164–17174, 2023. 3

[11] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Zhong Muyan, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint arXiv:2312.14238*, 2023. 2

[12] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. 1, 2

[13] Mingjing Du, Shifei Ding, and Hongjie Jia. Study on density peaks clustering based on k-nearest neighbors and principal component analysis. *Knowledge-Based Systems*, 99: 135–145, 2016. 3

[14] Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Juergen Gall. Adaptive token sampling for efficient vision transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 3

[15] Zhanzhou Feng and Shiliang Zhang. Efficient vision transformer via token merger. *IEEE Transactions on Image Processing*, 2023. 2, 3

[16] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023. 5

[17] Joakim Bruslund Haurum, Meysam Madadi, Sergio Escalera, and Thomas B. Moeslund. Multi-scale hybrid vision transformer and sinkhorn tokenizer for sewer defect classification. *Automation in Construction*, 144:104614, 2022. 3

[18] Joakim Bruslund Haurum, Sergio Escalera, Graham W Taylor, and Thomas B Moeslund. Which tokens to use? investigating token reduction in vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 773–783, 2023. 3

[19] Joakim Bruslund Haurum, Sergio Escalera, Graham W Taylor, and Thomas B Moeslund. Agglomerative token clustering. In *European Conference on Computer Vision*, pages 200–218. Springer, 2025. 3

[20] Kai Huang, Hao Zou, Ye Xi, BoChen Wang, Zhen Xie, and Liang Yu. Ivtp: Instruction-guided visual token pruning for large vision-language models. In *European Conference on Computer Vision*, 2025. 3

[21] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019. 5

[22] Shibo Jie, Yehui Tang, Jianyuan Guo, Zhi-Hong Deng, Kai Han, and Yunhe Wang. Token compensator: Altering inference cost of vision transformer without re-tuning. In *European Conference on Computer Vision*, pages 76–94. Springer, 2025. 3

[23] Chen Ju, Haicheng Wang, Haozhe Cheng, Xu Chen, Zhonghua Zhai, Weilin Huang, Jinsong Lan, Shuai Xiao, and

Bo Zheng. Turbo: Informativity-driven acceleration plug-in for vision-language large models. In *European Conference on Computer Vision*, pages 436–455. Springer, 2025. 1, 3, 5, 6, 7

[24] Minchul Kim, Shangqian Gao, Yen-Chang Hsu, Yilin Shen, and Hongxia Jin. Token fusion: Bridging the gap between token pruning and token merging. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1383–1392, 2024. 3

[25] Zhenglun Kong, Peiyan Dong, Xiaolong Ma, Xin Meng, Mengshu Sun, Wei Niu, Xuan Shen, Geng Yuan, Bin Ren, Minghai Qin, Hao Tang, and Yanzhi Wang. Spvit: Enabling faster vision transformers via soft token pruning, 2022. 3

[26] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022. 1

[27] Jin Li, Yaoming Wang, Xiaopeng Zhang, Bowen Shi, Dongsheng Jiang, Chenglin Li, Wenrui Dai, Hongkai Xiong, and Qi Tian. Ailurus: a scalable vit framework for dense prediction. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 3

[28] Youwei Liang, Chongjian GE, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. EVit: Expediting vision transformers via token reorganizations. In *International Conference on Learning Representations*, 2022. 3, 7

[29] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023. 2, 6

[30] Chen Lin and Xing Long. Open-llava-next: An open-source implementation of llava-next series for facilitating the large multi-modal model community. https://github.com/xiaoachen98/Open-LLaVA-NeXT, 2024. 5

[31] Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. Aligning large multi-modal model with robust instruction tuning. *arXiv preprint arXiv:2306.14565*, 2023. 2

[32] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023. 1, 5

[33] 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 preprint arXiv:2307.06281*, 2023. 5

[34] 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. 5

[35] Dmitrii Marin, Jen-Hao Rick Chang, Anurag Ranjan, Anish Prabhu, Mohammad Rastegari, and Oncel Tuzel. Token pooling in vision transformers for image classification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 12–21, 2023. 3

[36] OpenAI. Gpt-4 technical report, 2023. 2

[37] Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*, 2023. 2

[38] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems*, 2021. 3

[39] Cedric Renggli, André Susano Pinto, Neil Houlsby, Basil Mustafa, Joan Puigcerver, and Carlos Riquelme. Learning to merge tokens in vision transformers. *arXiv preprint arXiv:2202.12015*, 2022. 3

[40] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326, 2019. 5

[41] Woomin Song, Seunghyuk Oh, Sangwoo Mo, Jaehyung Kim, Sukmin Yun, Jung-Woo Ha, and Jinwoo Shin. Hierarchical context merging: Better long context understanding for pre-trained llms. *arXiv preprint arXiv:2404.10308*, 2024. 3

[42] Tianxiang Sun, Xiaotian Zhang, Zhengfu He, Peng Li, Qinyuan Cheng, Hang Yan, Xiangyang Liu, Yunfan Shao, Qiong Tang, Xingjian Zhao, et al. Moss: Training conversational language models from synthetic data. *arXiv preprint arXiv:2307.15020*, 7, 2023. 2

[43] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7, 2023.

[44] InternLM Team. Internlm: A multilingual language model with progressively enhanced capabilities, 2023.

[45] 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 preprint arXiv:2302.13971*, 2023.

[46] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 2

[47] Hongjie Wang, Bhishma Dedhia, and Niraj K Jha. Zero-tprune: Zero-shot token pruning through leveraging of the attention graph in pre-trained transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16070–16079, 2024. 3

[48] 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. 5

[49] Siyuan Wei, Tianzhu Ye, Shen Zhang, Yao Tang, and Jiajun Liang. Joint token pruning and squeezing towards more aggressive compression of vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2092–2101, 2023. 3

[50] Tao Wu, Xuewei Li, Zhongang Qi, Di Hu, Xintao Wang, Ying Shan, and Xi Li. Spherediffusion: Spherical geometry-aware distortion resilient diffusion model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6126–6134, 2024. 2

[51] Tao Wu, Yong Zhang, Xintao Wang, Xianpan Zhou, Guangcong Zheng, Zhongang Qi, Ying Shan, and Xi Li. Customcrafter: Customized video generation with preserving motion and concept composition abilities. *arXiv preprint arXiv:2408.13239*, 2024. 2

[52] Yizhe Xiong, Hui Chen, Tianxiang Hao, Zijia Lin, Jungong Han, Yuesong Zhang, Guoxin Wang, Yongjun Bao, and Guiguang Ding. Pyra: Parallel yielding re-activation for training-inference efficient task adaptation. In *European Conference on Computer Vision*, pages 455–473. Springer, 2025. 3

[53] Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*, 2023. 2

[54] Longrong Yang, Xianpan Zhou, Xuewei Li, Liang Qiao, Zheyang Li, Ziwei Yang, Gaoang Wang, and Xi Li. Bridging cross-task protocol inconsistency for distillation in dense object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 17175–17184, 2023.

[55] Longrong Yang, Dong Shen, Chaoxiang Cai, Fan Yang, Size Li, Di Zhang, and Xi Li. Solving token gradient conflict in mixture-of-experts for large vision-language model. *arXiv preprint arXiv:2406.19905*, 2024.

[56] Longrong Yang, Hanbin Zhao, Yunlong Yu, Xiaodong Zeng, and Xi Li. Rcs-prompt: Learning prompt to rearrange class space for prompt-based continual learning. In *European Conference on Computer Vision (ECCV)*, 2024. 2

[57] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*, 2023. 2

[58] Pan Zhang, Xiaoyi Dong Bin Wang, Yuhang Cao, Chao Xu, Linke Ouyang, Zhiyuan Zhao, Shuangrui Ding, Songyang Zhang, Haodong Duan, Hang Yan, et al. Internlm-xcomposer: A vision-language large model for advanced text-image comprehension and composition. *arXiv preprint arXiv:2309.15112*, 2023. 1

[59] Yanzhe Zhang, Ruiyi Zhang, Jiuxiang Gu, Yufan Zhou, Nedim Lipka, Diyi Yang, and Tong Sun. Llavar: Enhanced visual instruction tuning for text-rich image understanding. *arXiv preprint arXiv:2306.17107*, 2023. 2

[60] Bo Zhao, Boya Wu, and Tiejun Huang. Svit: Scaling up visual instruction tuning. *arXiv preprint arXiv:2307.04087*, 2023. 1, 2

[61] Zhuofan Zong, Kunchang Li, Guanglu Song, Yali Wang, Yu Qiao, Biao Leng, and Yu Liu. Self-slimmed vision transformer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 3, 5