IVTP: Instruction-guided Visual Token Pruning for Large Vision-Language Models

Kai Huang*, Hao Zou*, Ye Xi, BoChen Wang, Zhen Xie, and Liang Yu

Alibaba Group, China
{zhouwan.hk, zh372956, yx150449, bochen.wbc, xiezhen.xz, liangyu.yl}@alibaba-inc.com

Abstract. Inspired by the remarkable achievements of Large Language Models (LLMs), Large Vision-Language Models (LVLMs) have likewise experienced significant advancements. However, the increased computational cost and token budget occupancy associated with lengthy visual tokens pose significant challenge to the practical applications. Considering that not all visual tokens are essential to the final response, selectively pruning redundant visual tokens can effectively alleviate this challenge. In this paper, we present a novel Instruction-guided Visual Token Pruning (IVTP) approach for LVLMs, which is designed to strike a better balance between computational efficiency and the performance. Specifically, a Group-wise Token Pruning (GTP) module based on attention rollout is integrated into the grouped transformer layer to achieve intra-group attention aggregation via residual connection, thereby improving the assessment of visual token importance, especially for LVLMs with a frozen visual encoder. We then extend the module to LLM in order to further filter out visual tokens that are pertinent to the current textual instructions, by introducing a semantically related pseudo CLS token to serve as a reference for token pruning. This two-stage token pruning mechanism permits a systematic and efficient reduction in the quantity of visual tokens while preserving essential visual information. We apply the proposed method to the most representative LVLM, i.e. LLaVA-1.5. Experimental results demonstrate that when the number of visual tokens is reduced by 88.9%, the computational complexity is decreased by over 46%, with only an average 1.0% accuracy drop across 12 benchmarks, and remarkably surpasses the state-of-the-art token pruning methods.

Keywords: Visual Token Pruning, Large Vision-Language Models (LVLMs)

1 Introduction

Large Vision-Language Models (LVLMs) have garnered widespread interest from both the academic and industrial communities due to their impressive ability in handling cross-modality tasks [3,20,22,26,41]. Despite the remarkable progress of LVLMs, they still face challenges of high computational costs caused by lengthy

* Equal contribution.

image tokens. As a result, considerable works have been devoted to optimizing the trade-off between computational efficiency and model effectiveness in large models. To reduce the number of image tokens fed into Large Language Models (LLMs), some approaches [3, 20, 22] employ a trainable token compression module placed after the visual encoder, as shown in Fig. 1(a). Though capable of effectively pruning redundant visual tokens, the above methods are often tightly coupled with the model architecture, and the effectiveness of these structures often lacks comprehensive validation, which makes them difficult to transfer to other model frameworks. Building on the advancements in token pruning for traditional visual tasks [4, 8, 24, 31, 39], one direct approach is to adapt these existing vision-only pruning methods to LVLMs, where token pruning is performed only in the visual encoder, as shown in Fig. 1(b). Although these methods are transferable, the standard practice of freezing the visual encoder in LVLMs prevents the pruning process from being optimized end-to-end with the training of the LVLMs, consequently suffering from suboptimal stability.

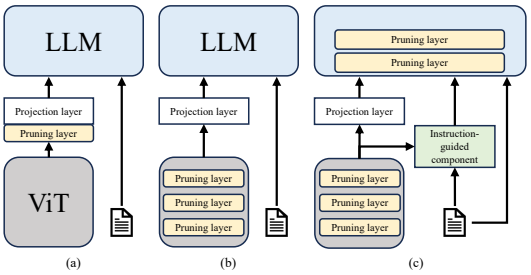

Fig. 1: Comparison of different token pruning schemes for LVLMs.

In this work, we propose an Instruction-guided Visual Token Pruning (IVTP) method to address the aforementioned issues, achieving a balance between accuracy and computational efficiency while also offering improved transferability and stability. As illustrated in Fig. 1(c), we divide the visual token pruning into two stages. The first stage is similar to vision-only token pruning, where redundant visual tokens with low informative content are eliminated in the ViT based on the attention connections within tokens. The second stage occurs within the LLM, it aims to remove visual tokens that exhibit a low correlation with the current textual instructions. Specifically, we determine the significance of each patch token by calculating its attentiveness in relation to the CLS token within the visual encoder. In order to ensure that the importance assessment of visual tokens remains effective within a frozen ViT, a Group-wise Token Pruning (GTP) module based on attention rollout [1] is proposed to integrate into the grouped visual transformer layers to achieve hierarchical attention weight aggregation. However, the continuous decrease in visual tokens results in less visual information being conveyed to the LLM, hindering its ability to follow diverse and variable visual instructions. To adaptively adjust the visual information contained by visual patch tokens according to different instructions, we further integrate the GTP into the LLM for instruction-guided visual token pruning. Since decoder-only LLMs do not incorporate a CLS token to represent text semantics, aggregating all textual tokens indiscriminately may introduce noise information that is not referred to image. The text encoder branch of CLIP [30], aligned with

the visual encoder of LVLMs, is introduced to filter out irrelevant textual tokens by calculating the token-wise relevance with the visual CLS token. Text tokens that match to the visual content are aggregated to form a textual pseudo CLS token so that the GTP module can be extended to LLM.

Our main contributions are: (1) We present a novel two-stage visual token pruning method for LVLMs, which can significantly reduce the visual tokens while preserving essential visual information, leading to a substantial increase in model computational efficiency with minimal loss of accuracy. (2) We propose a module that can work compatibly both within the ViT and the LLM, named Group-wise Token Pruning (GTP). Benefiting from the aggregation of attention weights based on attention rollout within groups across layers, the importance assessment of visual tokens is more robust and precise. Furthermore, a textual pseudo CLS gathered from visual relevant textual tokens, is used in GTP to introduce instruction preferences, thereby filtering out semantically irrelevant visual patch tokens and further compressing the number of visual tokens. (3) We introduce the IVTP to LLaVA-1.5 [25], the extensive experimentation on 12 standard benchmarks demonstrate that the proposed method achieves a superior trade-off between accuracy and computational efficiency.

2 Related Work

Large Vision-Language Models. Inspired by the success of LLMs [2, 29, 34], recent advancements in LVLMs have also demonstrated satisfying performance, such as [3, 22, 26, 41]. These approaches typically combine a pre-trained visual encoder with an LLM, enabling the LLM to handle multimodal data involving images. Furthermore, to align the multimodal data, a projection layer is appended after the visual encoder, such as a Q-former, linear mapping, or cross attention. BLIP2 [20] achieves stronger performance at a lower computation cost by freezing the visual encoder and LLM and only updating the weights of Q-former, which serves as the bridge for the modality gap. LLaVA [26] utilizes multimodal language-image instruction-following data generated by GPT-4 [2] for visual instruction tuning. Visual instruction tuning enables the LLaVA to accommodate users' diverse requests for instructions that involve visual content. During the training phase, QWen-VL [3] enhances the performance of model by updating the parameters of the visual encoder and utilizing higher-resolution images. Although increasing the number of model parameters and the resolution of input images can effectively improve the performance of LVLMs, the computational cost that they bring cannot be ignored. Our proposed ITVP can be directly integrated into a LVLM without training, reducing the computational cost of the LVLM by pruning redundant image tokens.

Token Pruning. Token pruning aims to retain attentive tokens and prune inattentive ones by designing importance evaluation strategies to create more efficient transformers in both natural language processing (NLP) and Computer Vision (CV). Benefiting from the success of BERT [7], many works [10, 16, 17, 38] in the NLP field have achieved text token pruning based on BERT, but these

methods require training and are difficult to adapt to the current LLMs. In the CV field, a classic method is DynamicViT [31], which inserts a trainable prediction module into the transformer to predict the importance score of each token. Subsequent works [8, 39] go further by sampling tokens with an input-dependent number. To address the issue of information loss in token pruning, [18, 24, 37] retain inattentive tokens by collapsing the pruned tokens into one through token reorganization. The above methods perform token pruning based on importance scores, but tokens that are close in the feature space will be assigned similar scores, leading to the possibility that similar tokens may be simultaneously retained or removed. Another classic approach is ToMe [4], which achieves token pruning by calculating the similarity between different tokens and merging tokens with high similarity. However, merging tokens can cause image distortion, making it difficult for the model to capture fine-grained information. Recent approaches [35, 36] attempt to mitigate the shortcomings of both types by combining the two methods, but how to effectively combine them remains an area for exploration. Based on [21], recent studies [12, 15] simultaneously utilize the CLS token of the visual branch and the CLS token of the text branch to implement image token pruning. The aforementioned methods of directly incorporating all textual information to guide the selection of visual tokens may introduce additional noise information, potentially causing instability in token selection. In this paper, we utilize the text branch of the CLIP to facilitate a more focused selection of visual tokens that are relevant to the provided instructions, thereby ensuring a refined filtration process to align with the context of the instructions.

3 Methodology

This research represents a preliminary effort to investigate the visual token pruning for LVLMs that achieves an optimal balance among performance, speed, cost and numbers of visual tokens. As shown in Fig. 2, our method is fundamentally based on established LVLMs, such as LLaVA-1.5 [25], which are succinctly reviewed in Sec. 3.1. Sec. 3.2 delves into the details of group-wise token pruning that leverages attention rollout to facilitate coarse-grained pruning within visual transformers. In Sec. 3.3, we further demonstrate the visual token pruning in LLMs guided by textual instruction. Finally, details of the two-stage manner applied to the ViT and LLM of LVLMs are presented in Sec. 3.4.

3.1 Revisit of Large Vision-Language Models

The mechanism of a LLM is complex and involves multiple layers of neural networks. Mathematically, a simplified conceptual representation could be expressed through the following formula:

$$p(\mathbf{X}) = \prod_{i=1}^{M} p(x_i^L | x_1^V, \ldots, x_F^V, x_1^L, \ldots, x_{i-1}^L; \Theta), \tag{1}$$

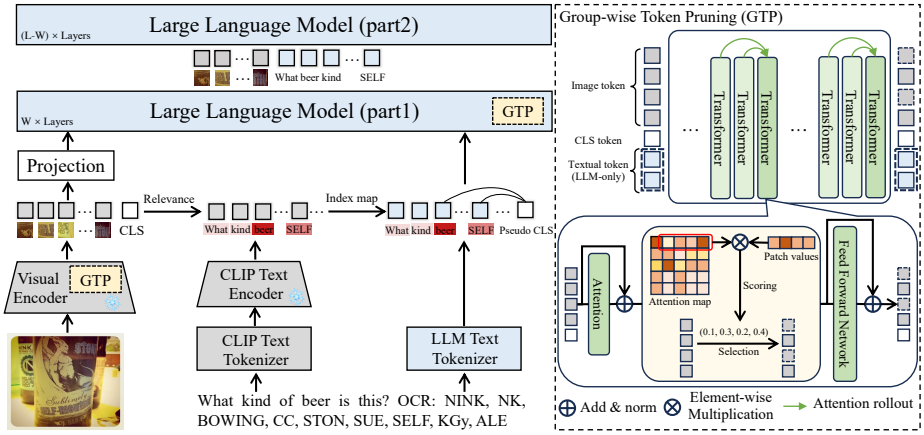

Fig. 2: Overview of Instruction-guided Visual Token Pruning (IVTP) for large vision-language models. The entire process is divided into two stages: The first stage takes place within the visual encoder of the LVLM, where a Group-wise Token Pruning (GTP) module is employed to discard redundant tokens with low informative content based on the inherent visual CLS token in ViT. The second stage operates on the first W layers of the LLM, in which we aggregate text instructions into a textual CLS token by introducing a frozen CLIP text encoder, and once again integrate GTP to remove visual tokens with low relevance to the current query.

where $p(\mathbf{X})$ represents the joint probability distribution of the entire sequence, and x_i denotes the ith token in the sequence. However, unlike the uniform token type in the standard LLMs, the input to the LLM in LVLMs consists of both visual tokens x^V and language tokens x^L. It predicts the next language token given the complete set of F visual tokens and the preceding $i-1$ language tokens. Generally, the visual tokens of an image are derived from the flattened grid features with a visual encoder, which are then mapped into the word embedding space. An image with a resolution of 336px results in 576 tokens when using ViT-L/14 as the visual transformer. Handling such a high number of tokens becomes challenging under the standard LLM's input limit of 2048 tokens. This limitation not only results in increased training and inference time consumption for LVLMs but also restricts the ability to process higher resolution or multiple images within the limited token budget.

3.2 Group-wise Token Pruning

Similar to vision-only pruning approaches, we aim to evaluate the significance of each patch token by examining their importance scores in LVLMs, which allows us to selectively prune patch tokens with minimal influence. Existing token pruning methods relied on importance scores are typically conducted in an incremental layer-specific manner, in which each layer prunes a certain number of tokens according to its importance scores, cumulatively achieving the overall to-

ken reduction goal. However, assessing token significance solely based on importance score from individual layer introduces a considerable amount of instability. It is essential to retrain the ViT to optimize the aggregation of patch information, while this makes it challenging in LVLMs as the visual encoder is typically kept frozen. To mitigate this, we employ an attention rollout mechanism [1] to derive group-wise token importance metric based on the aggregated attention weights across successive layers, which takes into account the inter-layer transfer of attention information. Compared to the original single-layer attention, the importance scores obtained from the attention with attention rollout exhibit a higher correlation with contributions to the prediction output. In NLP prediction tasks and visual visualization based on the transformer [1,6], the attention derived from attention rollout has been proven to be more stable and effective. Consequently, we intend to incorporate it into visual token pruning to enhance token decision-making for the frozen ViT.

In the transformer architecture, the attention weight A in the self-attention layer is calculated by the scaled dot product of the queries $\mathbf{Q} \in \mathbb{R}^{(N+1)\times d}$ and keys $\mathbf{K} \in \mathbb{R}^{(N+1)\times d}$:

$$\mathbf{A} = \text{Softmax}(\mathbf{Q}\mathbf{K}^T/\sqrt{d} + \mathbf{A}_{\text{mask}}) \in \mathbb{R}^{(N+1)\times(N+1)}, \tag{2}$$

where $\mathbf{A}_{\text{mask}}$ is the attention mask, which is None in the visual encoder and a causal matrix in the LLMs, N is the number of patch tokens and d is the embedding dimension. As previously described, we perform the attention rollout to better model the transfer of attention across layers by considering the residual connections within the network, thereby rendering the attention-based token importance assessment more reliable. Specifically, it adds an identity matrix to the attention matrix and subsequently re-normalizing the weights. To integrate the attention spanning from layer l_1 to layer l_2, a recursive multiplication is applied to the attention weights across all the corresponding range of layers as:

$$\tilde{\mathbf{A}}^l = \begin{cases} \mathbf{A}^{l_1}, & \text{if } l = l_1, \\ (\mathbf{A}^l + \mathbf{I})\tilde{\mathbf{A}}^{l-1}, & \text{if } l_1 < l \leq l_2. \end{cases} \tag{3}$$

The re-normalize operation is omitted to simplify the formula. To preserve the exceptional traits of hierarchical token pruning [4,18,31], we divide the entire network into groups. Within each group, we perform attention rollout to aggregate attention information from adjacent layers, and execute token pruning at the topmost layer of each group. Similar to [8], the normalized values of patch tokens $\mathbf{V} \in \mathbb{R}^{N\times d}$ are weighted by their corresponding entries in the CLS row of the attention matrix, taking into account both the positions and the corresponding strength of activation. The final importance score of ith patch $\mathcal{S}_i$ is derived from the normalization as:

$$\mathcal{S}_i = \frac{\tilde{\mathbf{A}}_{\text{cls},i} \times \|\mathbf{V}_i\|}{\sum_j \tilde{\mathbf{A}}_{\text{cls},j} \times \|\mathbf{V}_j\|} \in \mathbb{R}^{1\times N}, \tag{4}$$

where $\tilde{\mathbf{A}}_{\text{cls}}$ represents the row from the attention weight matrix associated with the CLS token, after excluding the CLS token itself. By ranking the patch tokens

according to their corresponding importance scores, we can selectively discard those low importance patch tokens within each group.

3.3 Instruction-guided Token Pruning in LLM

As the number of remaining patch tokens diminishes, the granularity of information preserved by the visual patch tokens also decreases. This reduction is inconsequential for standard visual token pruning methods, which are predominantly applied to classification tasks focused on extracting salient subject information. Moreover, hierarchical pruning provides an opportunity for the CLS token to aggregate information from patch tokens before they are dropped, thereby mitigating potential information loss. LVLMs are tasked with extracting pertinent detailed information from visual patches in response to diverse linguistic instructions. Consequently, the correlation between the residual visual patches and the instructions critically influences the performance in visual question answering.

A plausible strategy involves leveraging the instruction to guide the assessment of visual token significance, with the aim of selectively retaining those patch tokens that bear the most relevance to the text. Nonetheless, this approach encounters a significant challenge that the decoder-only LLM does not incorporate a CLS token representative of the semantic information, and aggregating all textual tokens in a crude manner also introduces the interference of noise. Therefore, the text branch of the CLIP is utilized to facilitate a more focused selection of visual tokens that are relevant to the provided instructions, thereby ensuring a refined filtration process to align with the context of the instructions. Specifically, given the text T, we can obtain its corresponding sequence of textual tokens $\{\bar{x}_1^L, \cdots, \bar{x}_{S'}^L\}$ after being processed by the pre-trained text encoder of CLIP as:

$$\{\bar{x}_1^L, \cdots, \bar{x}_{S'}^L\} = \mathcal{T}(T) \in \mathbb{R}^{S' \times d}, \tag{5}$$

where $\mathcal{T}$ is the text encoder of CLIP and S' denotes the length of textual tokens. Then, the visual relevance of textual tokens can be ascertained by computing the cosine similarity between the sequence of CLIP text tokens and the visual CLS tokens:

$$c_i = \frac{x_{\text{cls}}^{\text{V}} \bar{x}_i^L}{\|x_{\text{cls}}\| \left\|\bar{x}_i^L\right\|}, \text{ for } i \in 1, \cdots, S'. \tag{6}$$

Given that both the CLIP text encoder and general LLMs utilize tokenization derived from Byte Pair Encoding (BPE) [32], we can readily map the visual relevance computed by CLIP to the textual tokens of the LLMs as:

$$\mathcal{C}_i = \frac{1}{K+1} \sum_{k=j}^{j+K} c_k, \text{with } T_{\text{LLM}}(i) \subseteq \{T_{\text{CLIP}}(j), \cdots, T_{\text{CLIP}}(j+K)\}, \tag{7}$$

where $T_{\text{LLM}}(i)$ is the ith character string tokenized by LLM, and $T_{\text{CLIP}}(j)$ is the jth character string tokenized by text encoder of CLIP. Considering that

BPE may fragment a single word into multiple subword units, it is possible to encounter the situation where one token in LLMs corresponds to several tokens in the text encoder of CLIP. In such scenario, we take the average relevance of these $K+1$ tokens as the final mapping relevance. When K is zero, it means that the character string tokenized by the LLMs constitutes a substring within the text encoder of CLIP, and in this case, the current relevance can be used directly. By using the calculated visual relevance and applying average pooling with a predefined truncation threshold, we can pinpoint and extract the key textual tokens from the instructions that are most relevant to the corresponding image:

$$\hat{x}_{\text{cls}}^{T} = \begin{cases} x_{\text{cls}}^{V}, & \text{if } \sum \mathbb{I}_{\{\mathcal{C} \geq \tau\}} = 0, \\ \frac{1}{\sum \mathbb{I}_{\{\mathcal{C} \geq \tau\}}} \sum_{i=1}^{S} x_i^L \mathbb{I}_{\{\mathcal{C}_i \geq \tau\}}, & \text{Otherwise}, \end{cases} \tag{8}$$

where $\hat{x}_{\text{cls}}^{T}$ is the textual pseudo CLS token of LLMs, S represents the length of tokens tokenized by LLMs and $\mathbb{I}(\cdot)$ is an indicator function. If the relevance scores for the entire sequence of text tokens fall below the predefined threshold τ, it indicates that the current instructions do not specify particular visual details. In such scenario, we directly use the visual CLS token as the pseudo CLS token.

Building upon the methodology in Sec. 3.2, we continue to reuse the GTP module in the LLMs, and consider the attention weights corresponding to the pseudo CLS token as the basis to calculate the importance scores of each visual token. The calculated importance scores are then methodically used to inform the process of token pruning. Given the nature of the causal mask within the LLMs, we position the pseudo CLS token in between the sequence of visual tokens and textual tokens during the attention weights computation. This placement ensures that the pseudo CLS token can assess the entire sequence of visual tokens without being affected by subsequent textual tokens. Note that the role of the pseudo CLS token is confined to assisting in the token pruning process, and it does not serve as an actual token during the model's training and inference.

3.4 Two-stage Token Pruning Manner for LVLMs

Based on the above discussion, we divide the entire pruning process into two stages. The first stage operates across the entire ViT, where the GTP module is executed within the grouped visual transformer layers to eliminate tokens of low informational content based on the association between patch tokens and the visual CLS token. The second stage focuses on a subset of LLM layers, selectively dropping visual tokens unrelated to the current instruction. This decision is informed by the attention between the remaining visual tokens and the textual pseudo CLS token, which aggregates from selected key tokens of the instructions. By adopting this two-stage process, the proposed method can significantly reduce the number of visual tokens while effectively retaining essential visual information. Detailed information regarding the algorithm can be found in the appendix.

4 Experiments

4.1 Experimental Setup

In the following, we describe our experimental setup including datasets and evaluation, implementation details, and specifics of the comparison.

Datasets and Evaluation. The datasets employed for both training and evaluation in our study strictly conform to the specifications of LLaVA-1.5 [25]. In the pretraining phase, the 558K subset of the LAION-CC-SBU dataset with BLIP [21] captions are utilized. For instruction tuning, a comprehensive dataset with a diverse mix of 665K instruction-based examples are employed. We assess the performance of methods across 12 evaluation benchmarks, with the mean accuracy across these datasets as the principal metric for comparison. The scores of MME dataset [9] are normalized against the theoretical maximum value to facilitate uniformity in the scale of measurement. In the LLaVA-1.5 [25] framework, we primary experiment involved reducing visual tokens from 576 to 64.

Implementation Details. We perform token pruning every three layers, with the final count of visual tokens determined by the number pruned in each time. To balance model performance and efficiency, we employ all layers of the ViT and the initial 12 layers of the LLM for token pruning. The relevance threshold τ is empirically set to 0.2. The model training and inference configurations are strictly aligned with the original settings of LLaVA-1.5 [25] for a fair comparison, and all experiments are conducted on 8×A100 GPUs.

Comparison Details. We primarily compare three categories of token pruning methods. The first includes straightforward token sampling or linear aggregation approaches, such as random sampling, topK based on patch token similarity with the CLS token, subsampling predicated on spatial correlations. The second category encompasses state-of-the-art methods in vision-only tasks, such as EViT [24] and ToMe [4]. The third category features token pruning methods that have recently emerged in LVLMs, including the abstractor structure from Honeybee [5], context attention in LLaMA-ViD [22], and the visual adapter in Qwen-VL [3]. For the sake of brevity, we directly use the model names to represent these structures. To eliminate the influence of factors such as model architecture, data, and training strategies, we reapply the aforementioned methods within the LLaVA-1.5 [25] framework to conduct a fair and thorough comparison. Specific details regarding the application can be found in the appendix.

4.2 Experimental Results

Main Results. Table 1 and 2 demonstrate the comparison of performance with various methods as the number of visual tokens is reduced from 576 to 64 under the LLaVA-1.5-7B and LLaVA-1.5-13B frameworks [25] respectively. Despite some discrepancies between the replicated results and those reported in the literature across different datasets, the overall average differences are negligible, with the replication showing a slight edge in performance. Consequently, the replicated results is adopted as the baseline for comparison. The tables

Table 1: The comparison of the visual token pruning methods with Vicuna-7B on 12 benchmarks. *The replicate results with same experimental setting. The TFLOPs is measured with batch size of 1 as well as 512 text tokens.

Method	VQA^{v2} [11]	GQA [14]	VisWiz [13]	SQA^{I} [28]	VQA^{T} [33]	POPE [23]	MME [9]	MMB [27]	MMB^{CN} [27]	SEED [19]	LLaVA^{W} [26]	MM-Vet [40]	Avg. ↑	TFLOPs ↓
LLaVA-1.5-7B [25]	78.5	62.0	50.0	66.8	58.2	85.9	75.5	64.3	58.3	58.6	63.4	30.5	62.7	15.4
LLaVA-1.5-7B* [25]	79.1	62.7	49.0	67.8	58.6	86.3	72.8	66.2	59.3	58.5	63.7	31.7	63.0	15.4
Random sampling	69.0	57.1	37.9	67.2	48.5	82.5	65.6	55.4	48.0	51.0	55.8	23.6	55.1 (-7.9)	8.0 (-48.1%)
TopK	72.4	58.1	47.0	66.9	52.5	83.8	67.1	63.3	55.2	54.5	59.2	26.5	58.9 (-4.1)	8.0 (-48.1%)
Spatial pooling	73.9	59.6	46.5	67.7	52.5	82.3	68.5	63.3	56.6	54.9	59.7	28.3	59.5 (-3.5)	8.1(-47.4%)
EViT [24]	74.1	59.4	47.0	67.7	54.7	82.8	69.2	63.5	57.8	55.4	60.0	27.3	59.9 (-3.1)	8.0 (-48.1%)
ToMe [4]	75.1	60.0	47.1	67.5	55.3	82.4	70.4	63.9	56.5	55.2	60.5	26.6	60.0 (-3.0)	8.0 (-48.1%)
Honeybee [5]	74.8	59.0	47.2	67.8	50.9	84.0	68.7	61.6	57.8	55.2	59.4	27.1	59.5 (-3.5)	8.1 (-47.4%)
LLaMA-VID [22]	74.3	59.2	46.8	67.9	51.4	83.1	69.7	63.5	57.0	55.4	58.9	29.7	59.7 (-3.3)	8.2 (-46.8%)
Qwen-VL [3]	74.9	58.9	47.3	68.1	54.4	83.4	69.4	63.2	57.4	55.0	59.2	27.2	59.9 (-3.1)	8.1 (-47.4%)
IVTP (Ours)	77.8	60.4	47.9	67.8	58.2	85.7	72.6	66.1	57.4	56.4	62.8	30.5	62.0 (-1.0)	8.2 (-46.8%)

Table 2: The comparison of the visual token pruning methods with Vicuna-13B on 12 benchmarks. *The replicate results with same experimental setting. The TFLOPs is measured with batch size of 1 as well as 512 text tokens.

Method	VQA^{v2} [11]	GQA [14]	VisWiz [13]	SQA^{I} [28]	VQA^{T} [33]	POPE [23]	MME [9]	MMB [27]	MMB^{CN} [27]	SEED [19]	LLaVA^{W} [26]	MM-Vet [40]	Avg. ↑	TFLOPs ↓
LLaVA-1.5-13B [25]	80.0	63.3	53.6	71.6	61.3	85.9	76.6	67.7	63.6	61.6	70.7	35.4	65.9	29.4
LLaVA-1.5-13B* [25]	80.0	63.4	54.5	70.4	60.0	86.4	78.4	68.3	63.1	60.8	69.4	36.8	66.0	29.4
Random sampling	72.3	56.7	46.6	68.0	51.5	83.3	64.9	58.0	54.8	53.0	58.8	24.6	57.7 (-8.3)	15.4 (47.5%)
TopK	74.7	58.5	50.8	69.3	54.2	85.4	68.0	64.5	59.6	54.5	62.8	26.6	60.7 (-5.3)	15.4 (47.5%)
Spatial pooling	75.1	59.7	51.1	69.9	55.0	84.8	71.6	64.2	60.2	54.9	63.3	27.4	61.4 (-4.6)	15.6 (46.9%)
EViT [24]	77.2	60.2	53.4	70.1	57.9	84.6	73.6	65.3	60.1	55.4	64.9	28.6	62.6 (-3.4)	15.4 (47.5%)
ToMe [4]	76.9	61.4	53.9	70.1	57.6	85.5	73.1	65.0	61.2	56.0	65.9	32.6	63.3 (-2.7)	15.4 (47.5%)
Honeybee [5]	76.2	61.2	52.1	70.5	59.7	83.6	73.5	63.2	61.2	55.7	66.5	32.0	63.0 (-3.0)	15.4 (47.5%)
LLaMA-VID [22]	76.5	61.7	52.9	70.4	57.2	83.3	74.4	64.2	60.5	55.2	66.0	32.7	62.9 (-3.1)	15.5 (47.3%)
Qwen-VL [3]	77.3	61.1	52.1	70.8	56.4	84.0	71.7	65.8	61.7	56.3	66.7	31.5	63.0 (-3.0)	15.4 (47.5%)
IVTP (Ours)	78.4	62.3	54.1	70.1	60.0	85.4	77.1	67.7	63.3	59.3	68.6	35.5	65.2 (-0.8)	15.6 (46.9%)

show that despite reducing visual tokens by approximately 88.9%, the proposed method incurs an average performance decline of only about 1% across 12 evaluation datasets, coupled with over 46% reduction in computational demand. Our method exceeds the performance of other token pruning methods by over 2%, despite incurring a slightly higher TFLOPs due to hierarchical token pruning in the LLM. Table 3 further illustrates the change in TFLOPs corresponding to an increase in text tokens from 128 to 1024. It is clear that as the number of text tokens reduced, the proportion of visual tokens becomes more significant, thereby intensifying their influence on the computational demands of the model. As a result, the effectiveness of visual token pruning in reducing computational demand becomes more evident.

Inference-only Results. Since the proposed method does not need any extra modules that require parameter optimization, it can be seamlessly integrated with pre-trained models. This eliminates the need for retraining and simultaneously enhances computational efficiency during inference. Fig. 3 presents a performance comparison of various methods in a pure inference scenario when reducing the number of visual tokens from 16 to 512. As the number of target tokens decreases, the advantages of our proposed method become more pronounced. Notably, when the number of visual tokens is reduced to 16, our method surpasses other approaches by approximately 5%. The proposed method guided by instruction semantics can more effectively maintain the task relevant visual information, thereby achieving better results.

Table 3: Comparison results of TFLOPs across different methods as text token count varies from 128 to 1024.

Methods	128	256	512	1024
LLaVA-1.5-7B [25]	9.9	11.7	15.4	22.9
TopK	2.7 (-72.7%)	4.5 (-61.5%)	8.0 (-48.1%)	15.2 (-33.6%)
ToMe [4]	2.7 (-72.7%)	4.5 (-61.5%)	8.0 (-48.1%)	15.2 (-33.6%)
Honeybee [5]	2.9 (-70.7%)	4.6 (-60.7%)	8.1 (-47.4%)	15.4 (-32.8%)
Qwen-VL [3]	2.9 (-70.7%)	4.6 (-60.7%)	8.1 (-47.4%)	15.3 (-33.2%)
IVTP (Ours)	3.0 (-69.7%)	4.7 (-59.8%)	8.2 (-46.8%)	15.5 (-32.3%)

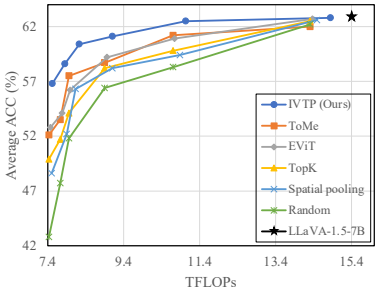

Fig. 3: Comparison of model performance with pure inference setting.

4.3 Ablation and Analysis

To analyze different setups of our IVTP, we perform extensive ablation studies. All the experiments are conducted with LLaVA-1.5-7B [25] as the backbone, and report the average accuracy across the aforementioned 12 evaluation datasets in both retraining mode and pure inference mode (PI).

Attention Rollout. As outlined in Sec. 3.2, we organize several contiguous layers into a group and apply attention rollout through the residual aggregation of intra-group attention weights. This process is intended to simulate the flow of information through hidden layer features, thereby yielding more reliable and effective attention weights for assessing the significance of visual patch tokens. Table 4 shows that substituting the group-wise attention rollout with a variant that uses only the CLS token's attention weight of current layer results in a performance decrease of 4.5% in pure inference mode and 3.9% in retraining mode. When switching from group-wise to layer-wise, the performance is improved, yet there is still a noticeable gap compared to the method using group-wise attention rollout. Although the proposed method requires to prune a greater number of tokens each time compared to layer-wise, it achieves better results with similar TFLOPs. This improvement is due to the more stable and efficient aggregation of attention, where the attention scores can more accurately reflect the contribution of patch tokens. Table 5 delves deeper into the impact of different attention information aggregation strategies. We evaluate several approaches for aggregating attention weights produced by different layers within a group, such as averaging, selecting the maximum, and implementing element-wise multiplication. The finding demonstrates that the residual method employed in our approach yields superior performance, owing to its improved alignment with the forward propagation of visual tokens.

Instruction Guided Selection. Another fundamental component of the proposed approach is the further selection of visual tokens guided by textual instructions, as elaborated in Sec. 3.3. To validate the effectiveness of this module, we also perform a series of ablation studies, which are presented in Table 6. As shown in the first row of the table, the attention weights of the visual

Table 4: Ablation study of the weights of attention rollout and vanilla attention. The metrics for comparison with the original LLaVA-1.5-7B are indicated in parentheses.

Model	Avg. Acc (PI)	Avg. Acc	TFLOPs
Group-wise	55.1 (-7.9)	58.1 (-4.9)	8.2 (-46.8%)
Layer-wise	56.2 (-6.8)	59.5 (-3.5)	8.2 (-46.8%)
Rollout	59.6 (-3.4)	62.0 (-1.0)	8.2 (-46.8%)

Table 5: Comparison with different attention weights aggregation strategies.

Model	Avg. Acc (PI)	Avg. Acc
mean	56.1 (-6.9)	60.1 (-2.9)
max	55.9 (-7.1)	60.7 (-2.3)
multiply	56.7 (-6.3)	60.5 (-2.5)
Residual	59.6 (-3.4)	62.0 (-1.0)

Table 6: Ablation study of instruction guided selection. 'OTT', 'TE', and 'RT' refer to original textual tokens, CLIP text encoder, and relevenance threshold, respectively.

OTT	TE	RT	Avg. Acc (PI)	Avg. ACC
			56.7 (-6.3)	59.2 (-3.8)
✓			58.3 (-4.7)	60.2 (-2.8)
✓		✓	57.3 (-5.7)	59.9 (-3.1)
✓	✓	✓	59.6 (-3.4)	62.0 (-1.0)

Table 7: $^{+}$Extending token pruning to LLM with other methods.

Model	Avg. Acc (PI)	Avg. Acc
TopK	54.1 (-8.9)	58.9 (-4.1)
TopK^{+}	54.9 (-8.1)	58.7 (-4.3)
ToMe [4]	57.5 (-5.5)	60.0 (-3.0)
ToMe^{+}	57.7 (-5.3)	60.1 (-2.9)
IVTP-V	58.5 (-4.5)	60.8 (-2.2)
IVTP	59.6 (-3.4)	62.0 (-1.0)

CLS token are used as the foundation for token selection within the LLM, establishing the baseline for these experiments. We then aggregate the original textual instruction tokens into a pseudo CLS token via average pooling to steer the token selection process. This approach yield a significant improvement over the baseline, demonstrating that guiding the pruning of visual tokens with textual instructions effectively isolates instruction-relevant visual patches through cross-modal associations, thus further reducing the number of required visual tokens. To minimize the influence of irrelevant noise in the textual instructions, we introduced a relevance threshold to selectively extracts key textual tokens. However, due to the lack of associative contrastive training between the visual CLS token and the original textual tokens of LLM, directly calculating vector correlation and applying a threshold for filtration is insufficient for isolating key textual tokens. The last row of the table demonstrates that integrating the CLIP text encoder, which naturally excels in image-text alignment, enables the precise identification of relevant sections within the textual tokens.

Pruning in LLM. We also extend our two-stage visual token pruning to TopK and ToMe [4], denoted as TopK^{+} and ToMe^{+} in Table 7, respectively. For a direct comparison, we apply the proposed method solely to the visual encoder, indicated as IVTP-V in the table. It calculation that extending the purely visual methods to LLM does not yield significant improvements. This is because performing token pruning solely within visual patches through representational

Table 8: Comparison with the numbers of layers in token pruning groups.

Layers	Avg. Acc (PI)	Avg. Acc
ALL	56.7 (-6.3)	58.2 (-4.8)
2	58.8 (-4.2)	60.1 (-2.9)
3	59.6 (-3.4)	62.0 (-1.0)
4	59.1 (-3.9)	61.5 (-1.5)
6	58.6 (-4.4)	59.9 (-3.1)

Table 9: Computational complexity with different methods (TFLOPs).

Methods	ViT	LLM	extra	Total
LLaVA-1.5-7B [25]	0.361	15.003	-	15.364
TopK	0.190	7.772	-	7.962
ToMe [4]	0.190	7.772	-	7.962
Qwen-VL [3]	0.360	7.772	0.003	8.135
IVTP (our)	0.202	7.946	0.080	8.228

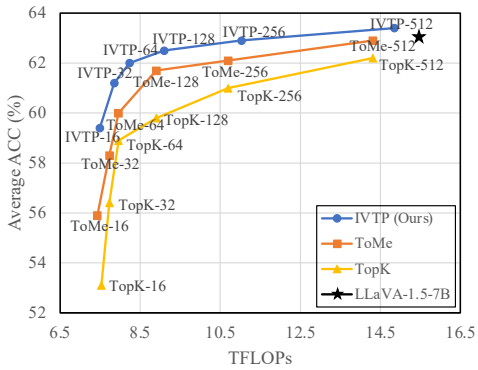

Fig. 4: Performance comparison with token pruning scaling. We show the average accuracy over all 12 benchmarks with the target number of visual tokens varying from 16 to 512 under comparable TFLOPs.

similarity fails to address the scenario with sparse target tokens. Rather than applying the token pruning from the visual encoder to the LLM directly, we select relevant visual tokens based on the diverse instructions.

Scaling the Token Pruning. Figure 4 illustrates the trade-off between TFLOPs and average accuracy by adjusting the final count of visual tokens from 576 down to a range between 16 and 512. The trend shown in the figure indicates that when the final number of visual tokens is reduced to between 256 and 512, the performance of most methods are nearly identical to those of the original model. The proposed method even slightly exceed the original model while achieving a 10% to 30% reduction in TFLOPs. This suggests that reducing the number of visual tokens to an optimal level can not only preserves essential visual information but also minimizes interference from irrelevant visual tokens. As the number of visual tokens is further reduced, relying solely on the relevance and uniqueness of patch tokens becomes inadequate for preserving visual information. To overcome this limitation, the proposed method introduces an instruction-guided mechanism that adaptively selects visual tokens highly relevant to the specific given prompt, thereby completing VQA tasks with fewer visual tokens.

Group Layers. We vary the number of layers in each token pruning group, with a specific focus on the scenario in which all layers of the ViT and all the first 12 layers of the LLM are employed. Table 8 shows the optimal performance is achieved with each group consists of three layers. Groups comprising fewer layers yield less effective attention rollout aggregation, whereas groups with more layers require pruning more tokens, hindering the layer-wise interaction within tokens.

Computational complexity. Table 9 shows the comparison of computational complexity among different models. We itemize the computational costs of the

Fig. 5: Visualization of the visual token pruning results. Each sample, viewed from left to right, consists of the raw image, the token pruning by the TopK, and the token pruning by the proposed method, respectively.

visual encoder, LLM and extra structures separately to provide a detailed breakdown. It is clear that the LLM contributes the most to the overall computational complexity of the LVLM, due to its significantly higher number of parameters compared to the ViT and other supplementary structures. Consequently, the disparities in TFLOPs observed with purely visual methods are not discernible when these methods are incorporated into the LVLMs. Although our method inputs more visual tokens into the LLM compared to other pruning methods, by limiting pruning to the initial 12 layers of the LLM, we manage to keep the computational complexity within 2% of that achieved by single-stage approaches.

Visualizations. We further visualize the results of token pruning for a more intuitive understanding in Fig. 5. key textual tokens are highlighted in the instruction text and discarded visual tokens are masked in the input image. Our observations indicate that the proposed method excels at precisely identifying essential textual tokens related to visual content and adeptly preserving the most relevant visual tokens aligned with the semantics of diverse instructions.

5 Conclusion

This paper presents a visual token pruning technique for Large Vision-Language Models (LVLMs). It introduces a group-wise token pruning (GTP) module aimed at improving the stability and robustness of patch token importance assessment within a frozen visual encoder of LVLMs. The proposed method extends the designed GTP module to LLM with incorporating instruction semantics as guidance, thereby discarding irrelevant visual tokens based on instructional preferences. Comprehensive experiments and in-depth ablation studies show that the proposed method can prune more visual tokens with minimal loss of accuracy, as well as accelerate the training and inference of LVLMs. The reduction of visual tokens can also be transferred to multi-image or video tasks in LVLMs to boost their performance, which will be a key focus of our future work.

References

1. Abnar, S., Zuidema, W.: Quantifying attention flow in transformers. arXiv preprint arXiv:2005.00928 (2020) 2, 6
2. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023) 3
3. Bai, J., Bai, S., Yang, S., Wang, S., Tan, S., Wang, P., Lin, J., Zhou, C., Zhou, J.: Qwen-vl: A frontier large vision-language model with versatile abilities. arXiv preprint arXiv:2308.12966 (2023) 1, 2, 3, 9, 10, 11, 13
4. Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token merging: Your vit but faster. In: The Eleventh International Conference on Learning Representations (2022) 2, 4, 6, 9, 10, 11, 12, 13
5. Cha, J., Kang, W., Mun, J., Roh, B.: Honeybee: Locality-enhanced projector for multimodal llm. arXiv preprint arXiv:2312.06742 (2023) 9, 10, 11
6. Chefer, H., Gur, S., Wolf, L.: Transformer interpretability beyond attention visualization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 782–791 (2021) 6
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018) 3
8. Fayyaz, M., Koohpayegani, S.A., Jafari, F.R., Sengupta, S., Joze, H.R.V., Sommerlade, E., Pirsiavash, H., Gall, J.: Adaptive token sampling for efficient vision transformers. In: Proceedings of the IEEE/CVF European Conference on Computer Vision. pp. 396–414. Springer (2022) 2, 4, 6
9. Fu, C., Chen, P., Shen, Y., Qin, Y., Zhang, M., Lin, X., Yang, J., Zheng, X., Li, K., Sun, X., et al.: Mme: A comprehensive evaluation benchmark for multimodal large language models. arXiv preprint arXiv:2306.13394 (2023) 9, 10
10. Goyal, S., Choudhury, A.R., Raje, S., Chakaravarthy, V., Sabharwal, Y., Verma, A.: Power-bert: Accelerating bert inference via progressive word-vector elimination. In: International Conference on Machine Learning. pp. 3690–3699. PMLR (2020) 3
11. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6904–6913 (2017) 10
12. Guo, Y., Zhang, H., Nie, L., Wong, Y., Kankanhalli, M.: Elip: Efficient language-image pre-training with fewer vision tokens. arXiv preprint arXiv:2309.16738 (2023) 4
13. Gurari, D., Li, Q., Stangl, A.J., Guo, A., Lin, C., Grauman, K., Luo, J., Bigham, J.P.: Vizwiz grand challenge: Answering visual questions from blind people. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3608–3617 (2018) 10
14. Hudson, D.A., Manning, C.D.: Gqa: A new dataset for real-world visual reasoning and compositional question answering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6700–6709 (2019) 10
15. Jiang, C., Xu, H., Ye, W., Ye, Q., Li, C., Yan, M., Bi, B., Zhang, S., Huang, F., Huang, S.: Bus: Efficient and effective vision-language pre-training with bottom-up patch summarization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2900–2910 (2023) 4

16. Kim, G., Cho, K.: Length-adaptive transformer: Train once with length drop, use anytime with search. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. pp. 6501–6511 (2021) 3
17. Kim, S., Shen, S., Thorsley, D., Gholami, A., Kwon, W., Hassoun, J., Keutzer, K.: Learned token pruning for transformers. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 784–794 (2022) 3
18. Kong, Z., Dong, P., Ma, X., Meng, X., Niu, W., Sun, M., Shen, X., Yuan, G., Ren, B., Tang, H., et al.: Spvit: Enabling faster vision transformers via latency-aware soft token pruning. In: Proceedings of the IEEE/CVF European Conference on Computer Vision. pp. 620–640. Springer (2022) 4, 6
19. Li, B., Wang, R., Wang, G., Ge, Y., Ge, Y., Shan, Y.: Seed-bench: Benchmarking multimodal llms with generative comprehension. arXiv preprint arXiv:2307.16125 (2023) 10
20. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. arXiv preprint arXiv:2301.12597 (2023) 1, 2, 3
21. Li, J., Li, D., Xiong, C., Hoi, S.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: International Conference on Machine Learning. pp. 12888–12900. PMLR (2022) 4, 9
22. Li, Y., Wang, C., Jia, J.: Llama-vid: An image is worth 2 tokens in large language models. arXiv preprint arXiv:2311.17043 (2023) 1, 2, 3, 9, 10
23. Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, W.X., Wen, J.R.: Evaluating object hallucination in large vision-language models. arXiv preprint arXiv:2305.10355 (2023) 10
24. Liang, Y., Chongjian, G., Tong, Z., Song, Y., Wang, J., Xie, P.: Evit: Expediting vision transformers via token reorganizations. In: International Conference on Learning Representations (2021) 2, 4, 9, 10
25. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. arXiv preprint arXiv:2310.03744 (2023) 3, 4, 9, 10, 11, 13
26. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. Advances in neural information processing systems **36** (2024) 1, 3, 10
27. Liu, Y., Duan, H., Zhang, Y., Li, B., Zhang, S., Zhao, W., Yuan, Y., Wang, J., He, C., Liu, Z., et al.: Mmbench: Is your multi-modal model an all-around player? arXiv preprint arXiv:2307.06281 (2023) 10
28. Lu, P., Mishra, S., Xia, T., Qiu, L., Chang, K.W., Zhu, S.C., Tafjord, O., Clark, P., Kalyan, A.: Learn to explain: Multimodal reasoning via thought chains for science question answering. Advances in Neural Information Processing Systems **35**, 2507–2521 (2022) 10
29. OpenAI: Chatgpt. https://openai.com/blog/chatgpt/, 2023. 1, 2 (2023) 3
30. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021) 2
31. Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., Hsieh, C.J.: Dynamicvit: Efficient vision transformers with dynamic token sparsification. Advances in neural information processing systems **34**, 13937–13949 (2021) 2, 4, 6
32. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. arXiv preprint arXiv:1508.07909 (2015) 7

33. Singh, A., Natarajan, V., Shah, M., Jiang, Y., Chen, X., Batra, D., Parikh, D., Rohrbach, M.: Towards vqa models that can read. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8317–8326 (2019) 10
34. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023) 3
35. Wei, S., Ye, T., Zhang, S., Tang, Y., Liang, J.: 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. pp. 2092–2101 (2023) 4
36. Wu, X., Zeng, F., Wang, X., Wang, Y., Chen, X.: Ppt: Token pruning and pooling for efficient vision transformers. arXiv preprint arXiv:2310.01812 (2023) 4
37. Xu, Y., Zhang, Z., Zhang, M., Sheng, K., Li, K., Dong, W., Zhang, L., Xu, C., Sun, X.: Evo-vit: Slow-fast token evolution for dynamic vision transformer. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 2964–2972 (2022) 4
38. Ye, D., Lin, Y., Huang, Y., Sun, M.: Tr-bert: Dynamic token reduction for accelerating bert inference. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 5798–5809 (2021) 3
39. Yin, H., Vahdat, A., Alvarez, J.M., Mallya, A., Kautz, J., Molchanov, P.: A-vit: Adaptive tokens for efficient vision transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10809–10818 (2022) 2, 4
40. Yu, W., Yang, Z., Li, L., Wang, J., Lin, K., Liu, Z., Wang, X., Wang, L.: Mm-vet: Evaluating large multimodal models for integrated capabilities. arXiv preprint arXiv:2308.02490 (2023) 10
41. Zhu, D., Chen, J., Shen, X., Li, X., Elhoseiny, M.: Minigpt-4: Enhancing vision-language understanding with advanced large language models. arXiv preprint arXiv:2304.10592 (2023) 1, 3