PACT: Pruning and Clustering-Based Token Reduction for Faster Visual Language Models

Mohamed Dhouib
LIX, École Polytechnique, IP Paris, France
mohamed.dhouib@polytechnique.edu

Davide Buscaldi
LIPN, Université Sorbonne Paris Nord, France
davide.buscaldi@lipn.univ-paris13.fr

Sonia Vanier
LIX, École Polytechnique, IP Paris, France
sonia.vanier@polytechnique.edu

Aymen Shabou
DataLab Groupe, Crédit Agricole S.A, France
aymen.shabou@credit-agricole-sa.fr

Abstract

*Visual Language Models require substantial computational resources for inference due to the additional input tokens needed to represent visual information. However, these visual tokens often contain redundant and unimportant information, resulting in an unnecessarily high number of tokens. To address this, we introduce **PACT**, a method that reduces inference time and memory usage by pruning irrelevant tokens and merging visually redundant ones at an early layer of the language model. Our approach uses a novel importance metric to identify unimportant tokens without relying on attention scores, making it compatible with FlashAttention. We also propose a novel clustering algorithm, called Distance Bounded Density Peak Clustering, which efficiently clusters visual tokens while constraining the distances between elements within a cluster by a predefined threshold. We demonstrate the effectiveness of **PACT** through extensive experiments.*

1. Introduction

Extending Large language models to modalities other than text [12, 19, 20, 54, 55] has seen success in recent years across various domains, especially in the visual domain with models like LLaVA [31] and Qwen-VL [4]. State-of-the-art Visual Language Models generally consist of three main components: a vision encoder, a connector, and a language model. The vision encoder converts input images into visual tokens, which are passed through the connector and then fed to the language model along with the input text. While this architecture has shown impressive performance across different tasks, it suffers from high computational cost due to the large number of visual tokens. In this paper, we introduce two complementary methods to optimize Visual Language Models by reducing inference time and memory requirements: a pruning module and a clustering algorithm. These methods can be used independently or combined, forming the **PACT** approach for greater effectiveness. Notably, our pruning and clustering modules, as well as **PACT**, are applied at inference time and thus require no additional training. The pruning module identifies unimportant visual tokens based on a novel importance metric that evaluates each token's relevance without relying on attention scores. This makes it compatible with FlashAttention [13], as FlashAttention does not support the calculation of attention scores. The second module introduces a novel clustering algorithm, **Distance Bounded Density Peak Clustering (DBDPC)**, which clusters visual tokens while ensuring that the distances between elements within a cluster are constrained by a predefined threshold. By combining these two methods, we develop **PACT**. First, the pruning module eliminates unimportant tokens, then the **DBDPC** algorithm clusters the remaining ones. Tokens that were initially pruned but are sufficiently close to the constructed clusters are reincorporated, ensuring that valuable information from the pruned tokens is recovered. Finally, the tokens within each cluster are merged into a single representative token, reducing the total token count.

By combining both pruning and clustering, **PACT** achieves an effective visual token reduction, addressing both irrelevant and redundant tokens. When applied to LLaVA-OneVision-7B, **PACT** achieves a 50% visual token reduction with negligible performance loss. Moreover, **PACT** exhibits significantly less performance degradation at higher reduction ratios compared to previous methods, achieving 71.3% visual token reduction ratio with only 1.4% performance drop, whereas previous state-of-the-art methods show at best a 4.4% performance drop at an equal reduction ratio. Our contributions are as follows:

- We propose a novel visual token pruning metric that does

not rely on attention scores, ensuring compatibility with FlashAttention, and empirically validate its effectiveness.
- We introduce a new clustering algorithm aimed at reducing visual redundancy and show its superiority over other clustering algorithms for visual token reduction.
- We show that combining pruning with clustering-based merging surpasses either technique alone for visual token reduction. By integrating our pruning and clustering algorithms, we propose a novel approach, **PACT**, and demonstrate that it outperforms previous and concurrent works [3, 7, 10, 30, 43]. The codebase used to obtain the results in this study is available at https://github.com/orailix/PACT/tree/main .

2. Related work

2.1. Visual language models

Since the introduction of BLIP-2 [28], the use of a visual encoder followed by a connector that feeds visual vectors to the language model has become the standard architecture for Visual Language Models (VLMs) [8, 18, 48]. Recent models [11, 27, 47] have enhanced VLM architecture with high-resolution handling, which is necessary for document understanding tasks [14, 24]. LLaVA-OneVision [27] divides images into 384×384 crops, encodes each part with SigLIP [53], and uses bilinear interpolation to reduce token count up to 8,748 tokens. InternVL2 [11] splits images into 448×448 tiles, processing up to 40 tiles per image with InternViT [11], and applies pixel shuffle to reduce the number of visual tokens, producing up to 10,240 tokens. Qwen-VL2 [47] uses 2D Rotary Positional Embeddings for dynamic resolution support and merges adjacent tokens via an MLP layer, yet still requires over 10,000 tokens for high resolution images. While these models apply token reduction by merging adjacent tokens to preserve structure, they do not address token irrelevance or redundancy, limiting efficiency.

2.2. Visual token reduction

Reducing the number of visual tokens in Vision Transformers (ViT) has been a key focus of the research community for several years. EViT [29] identifies and merges irrelevant tokens by relying on the attention scores between the class token ([CLS]) and visual tokens. ToME [7] proposed a simple yet effective approach that iteratively merges similar tokens throughout the ViT layers. Building on these ideas, recent efforts have extended visual token reduction techniques to VLMs. LaVIT [22] used the Gumbel-Softmax [21] to train a mask that selects tokens for retention, merging discarded tokens into retained ones via additional attention layers. LLaVA-PruMerge [43] accelerates LLAVA 1.5 [31] by leveraging the attention scores between the [CLS] token and visual tokens in the last layer of the ViT encoder to decide which tokens to retain, while HiRED [3] refines this approach by allocating token budgets based on attention from earlier layers. However, both these methods are only applicable to architectures where a ViT is used and a [CLS] token is added to the input visual sequence, making them incompatible with the majority state-of-the-art VLMs, which do not use a [CLS] token. Moreover, both methods attribute scores to tokens at the output of the visual encoder, but recent VLMs merge adjacent visual tokens before passing them to the language model. It is unclear how to attribute pre-merging scores to the resulting tokens, making LLaVA-PruMerge and HiRED inapplicable. We note that LLaVA-PruMerge mitigates information loss by merging pruned tokens with retained ones. However, it does not merge similar retained tokens; therefore, it does not address visual redundancy, a typical limitation of pruning-based methods. TRIM [45] prunes tokens based on similarity with pooled text from CLIP [41]. However, as TRIM relies on textual information for pruning, it is less suitable for multi-turn conversations where, in practice, visual tokens would be pruned solely based on the text information available during the image's forward pass, potentially losing crucial information required to answer subsequent prompts. FastV [10] evaluates token importance via average attention scores, which is not compatible with FlashAttention, adding computational overhead for recent VLMs. VTW [30] removes tokens in deeper layers. While this method shows promising results, its reduction of computational costs is limited as visual tokens are only withdrawn in later layers. These previous methods address only one of two issues: the presence of unimportant tokens or visual redundancy. In this work, we introduce **PACT**, a novel approach that tackles both issues simultaneously by pruning irrelevant tokens and merging visually redundant ones.

3. Method

In this section, we present **PACT**, a method that aims to reduce VLMs inference time and memory usage by pruning unimportant tokens and merging visually redundant ones at an early layer L of the language model. **PACT** consists of three steps: First, unimportant tokens are identified. Next, the remaining tokens are clustered. Finally, tokens in each cluster, along with sufficiently close tokens that were initially discarded, are merged. **PACT** operates within a selected layer L of the language model and is applicable in scenarios where visual tokens are fed into the language model, regardless of the architecture of the visual encoder or connector. We denote the hidden states at layer L by $\mathbf{H} \in \mathbb{R}^{n \times d}$, where n is the number of visual tokens and d is the dimensionality of the hidden states. We denote by $\mathbf{K}, \mathbf{Q} \in \mathbb{R}^{n \times n_h \times d_h}$ the key and query matrices for the visual tokens at layer L, where n_h represents the number of attention heads and d_h is the dimensionality of each attention head. For simplicity, we omit the layer index in the

Algorithm 1 EUTI

Input: Hidden states $\mathbf{H} \in \mathbb{R}^{n \times d}$; key and query matrices $\mathbf{K}, \mathbf{Q} \in \mathbb{R}^{n \times n_h \times d_h}$; pruning percentage $\lambda \in [0, 1]$

Output: Sets of important and unimportant visual tokens

Step 1: Calculate the global query vector

$\mathbf{Q}_{\text{global}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{Q}_i$

Step 2: Compute the importance score for each visual token

for all $i = 1, \ldots, n$ **do**

$s_i = \frac{1}{n_h} \sum_{j=1}^{n_h} Softmax(\mathbf{k}_i^{(j)} \cdot \mathbf{Q}_{\text{global}}^{(j)}) \cdot \|\mathbf{h}_i\|_2$

end for

Step 3: Define sets of important and unimportant tokens

$S_{\text{important}} = \{i \mid s_i \geq \text{Percentile}(s, \lambda)\}$

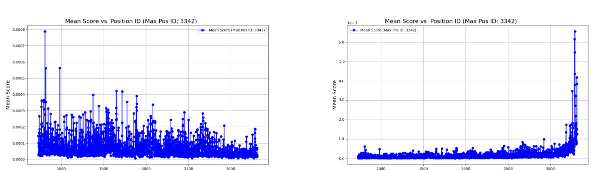
$S_{\text{unimportant}} = \{i \mid s_i < \text{Percentile}(s, \lambda)\}$

Return $S_{\text{important}}$, $S_{\text{unimportant}}$

notation. We denote the position index of a token by a subscript, while the attention head is indicated by a superscript. For instance, $\mathbf{k}_i^{(j)}$ represents the key vector corresponding to the i-th visual token and the j-th attention head.

3.1. Unimportant tokens identification

A straightforward approach to identifying unimportant tokens at a certain layer L of the used language model is to define the importance of each token as the total attention score that a given token receives from all other tokens [10]. However, this method has three main drawbacks. First, current VLMs utilize FlashAttention [13], which does not support outputting attention scores. Secondly, attention scores are computed with masking, which introduces biases. Tokens at the end of a sequence tend to receive lower average attention scores since fewer tokens attend to them. Calculating the average attention score for each token based solely on the tokens that attend to it can mitigate this masking effect but introduces a new bias: end-of-sequence tokens may exhibit higher scores as they receive attention mainly from nearby tokens. This leads to either earlier or later tokens being pruned more frequently, as shown in Fig. 1. Such positional bias should be avoided, as pruning should depend solely on the information that visual tokens hold, not their position. Finally, relying only on keys and queries at a single layer to determine an importance metric may fail to fully capture the significance of visual tokens across all layers of the language model, mainly because each self-attention layer focuses on different aspects of the visual tokens. To address this, we propose



(a) Average attention scores as a function of Position IDs.

(b) Average attention scores relative to non-masked tokens as a function of Position IDs.

Figure 1. **Illustration of the bias induced by the use of the average attention scores across visual tokens as a pruning metric.** In (a), averaging attention over all tokens favors earlier tokens, leading to pruning later tokens more frequently. In (b), averaging only over attending tokens reverses the bias, leading to earlier tokens being pruned more often.

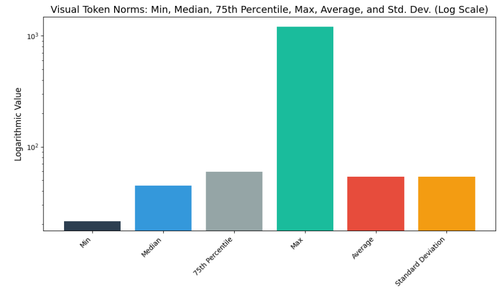


Figure 2. **Illustration of visual token norm statistics at the fourth layer of LLaVA-OneVision-7B.**

Table 1. **Throughput ratio, reduction ratio, and GPU memory usage for PACT, FastV, VTW, and ToME applied to LLaVA-OneVision-7B. Results are reported at a 98.6% Approach-to-Reference Metric Ratio.**

	No reduction	PACT (ours)	FastV	VTW	ToME
Reduction Ratio	0%	**71.3%**	50%	25%	40%
Throughput ratio	100%	**225%**	165%	160%	137%
GPU Maximum Memory consumption (GB)	27.4	**19.05**	30.4	19.2	21.4

an importance metric that incorporates the accumulated information from the hidden states and the layer-specific information from the keys and queries at an early layer L. We refer to this method as **E**fficient **U**nimportant **T**okens **I**dentification (**EUTI**). We speculate that the norm of hidden states can provide critical information about the importance of each visual token, as they reflect how much information a particular token carries through the network. Figure 2 presents statistics on the hidden state norms of visual tokens at the fourth layer of LLaVA-OneVision-7B, indicating a high variance. This variance suggests that certain visual tokens accumulate more information through residual connections and may therefore be more important for subsequent calculations. To leverage information from both hidden state norms and the key and query vectors, we first compute a global query vector $\mathbf{Q}_{\text{global}}$ as the average of all query vectors across visual tokens:

$$\mathbf{Q}_{\text{global}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{Q}_i \tag{1}$$

This vector represents the overall query information requested by visual tokens at layer L across all attention heads. The importance score for each visual token is then computed by first taking the dot product between its key and the global query for each attention head. A softmax is applied across visual tokens within each attention head, followed by averaging across attention heads. The final score is obtained by scaling the result with the hidden state norm:

$$s_i = \frac{1}{n_h} \sum_{j=1}^{n_h} \text{Softmax}\left(\mathbf{k}_i^{(j)} \cdot \mathbf{Q}_{\text{global}}^{(j)}\right) \cdot \|\mathbf{h}_i\|_2 \tag{2}$$

Then, we divide the visual tokens into important and unimportant tokens, using a parameter $\lambda \in [0, 1]$ to control the percentage of tokens deemed unimportant. The two sets are defined as follows:

$$S_{\text{important}} = \{i \mid s_i \geq \text{Percentile}(s, \lambda)\} \tag{3}$$

$$S_{\text{unimportant}} = \{i \mid s_i < \text{Percentile}(s, \lambda)\} \tag{4}$$

Unimportant tokens can be pruned, or the resulting sets can be combined with a clustering algorithm to further reduce the number of visual tokens, as we will show in the next section. The full **EUTI** algorithm is illustrated in Algorithm 1.

3.2. Clustering-based merging of visual tokens

Distance Bounded Density Peak Clustering Relying solely on the importance scores presented above to prune unimportant tokens can lead to a significant reduction in visual tokens, retaining only important ones. However, redundant information may still be present across retained visual tokens. Therefore, we propose merging the redundant visual tokens using a clustering algorithm. We desire our clustering algorithm to have the following characteristics:

(a) Low computational time.

(b) Avoid assigning points that are far from each other, in terms of feature similarity, into the same cluster.

Condition (b) ensures that outliers are not assigned to distant cluster centers, as we speculate that these outliers contain important information and should only be merged with

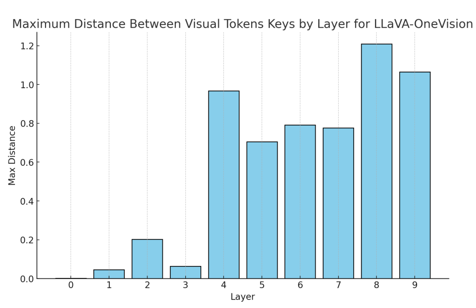


Figure 3. **Illustration of the maximum distance between the keys of visual tokens for the first 10 layers of LLaVA-OneVision-7B before the application of rotary embeddings.**

nearby outliers or remain as single points in separate clusters. Condition (b) also guarantees that points in each cluster will be relatively close to each other, which minimizes information loss when assigning a single vector as their representative. The Density Peaks Clustering (DPC) algorithm [5] is appealing in this context because it satisfies condition (a), unlike iterative clustering algorithms like k-means [2]. However, DPC does not satisfy condition (b) as it can form large clusters where boundary points may be distant from each other. The same issue arises with other algorithms such as DBSCAN [15]. Therefore, we propose a new clustering algorithm, which we call **D**istance **B**ounded **D**ensity **P**eaks **C**lustering (**DBDPC**).

DBDPC takes as input a set of vectors $\{\mathbf{u}_i \in \mathbb{R}^{d_1}\}_{i=1}^{q}$, where $q, d_1 \in \mathbb{N}^+$, and outputs a set of clusters. Our algorithm's output depends on two parameters, the cutoff distance $d_c \in \mathbb{R}^+$ and a normalization factor $d_n \in \mathbb{R}^+$, as well as a distance function $d : \mathbb{R}^{d_1} \times \mathbb{R}^{d_1} \to \mathbb{R}^+$. We define the distance between two vectors $\mathbf{u}_i$ and $\mathbf{u}_j$ as:

$$d_{ij} = d(\mathbf{u}_i, \mathbf{u}_j) = 1 - \frac{\mathbf{u}_i \cdot \mathbf{u}_j}{\|\mathbf{u}_i\|_2 \|\mathbf{u}_j\|_2} \tag{5}$$

Then the local density ρ_i is calculated as:

$$\rho_i = \sum_j e^{-d_{ij}/d_n} \tag{6}$$

We process the $\mathbf{u}_i$ vectors from highest to lowest ρ values and designate a vector as a cluster center if its minimum distance from already selected centers is greater than d_c. Each vector $\mathbf{u}_i$ is then assigned to the cluster of the closest center. Our algorithm guarantees that the distance from each vector to its cluster center is less than d_c, thereby satisfying condition (b) stated above. The full **DBDPC** algorithm is detailed in Algorithm 2. The center identification process in **DBDPC** ensures that inter-cluster distances are upper-bounded by $2d_c \times (2 - d_c)$ while distances between cluster centers are lower-bounded by d_c, which we formally prove in Appendix B. We note that several parts of our algorithm are presented as for-loops for clarity. However, all computations are parallelizable on GPU, as there are no dependencies between the elements of each loop, except for the part where we select cluster centers. For this part, we use a recursive algorithm that efficiently identifies an initial set of centers and discarded vectors, thereby reducing the number of vectors to be processed. We explain this in detail in Appendix D. For a comparison between **DBDPC** and DPC, as well as a qualitative comparison with other clustering algorithms, refer to Appendix C.

Which vectors should be used for distance calculation? As previously discussed, the **DBDPC** algorithm operates on a set of vectors that are used for distance calculation. To achieve effective clustering, the dot product between these vectors needs to accurately reflect the similarity between the

Algorithm 2 DBDPC

Input: Cutoff distance $d_c \in \mathbb{R}^+$, normalization factor $d_n \in \mathbb{R}^+$, set of vectors $\{\mathbf{u}_i \in \mathbb{R}^{d_1}\}_{i=1}^q$

Output: Cluster center indices C_{centers}, element indices in each cluster C_{elements}

for all pairs $(\mathbf{u}_i, \mathbf{u}_j)$ **do**
  $d_{ij} = 1 - \frac{\mathbf{u}_i \cdot \mathbf{u}_j}{\|\mathbf{u}_i\|_2 \|\mathbf{u}_j\|_2}$
end for
for all vectors $\mathbf{u}_i$ **do**
  $\rho_i = \sum_{j=1}^q e^{-d_{ij}/d_n}$
end for
Sort vectors by ρ_i in descending order, obtaining indices $[i_1, i_2, \ldots, i_q]$
Initialize $C_{\text{centers}} = \{i_1\}$, $C_{\text{elements}} = \{i_1 : \emptyset\}$
for all indices i_k in sorted order **do**
  if $\min_{s \in C_{\text{centers}}} d_{i_k s} > d_c$ **then**
    $C_{\text{centers}} = C_{\text{centers}} \cup \{i_k\}$
    $C_{\text{elements}}[i_k] = \emptyset$
  end if
end for
for all indices i **do**
  $s_i = argmin_{s \in C_{\text{centers}}} d_{is}$
  $C_{\text{elements}}[s_i] = C_{\text{elements}}[s_i] \cup \{i\}$
end for
Return C_{centers}, C_{elements}

corresponding visual tokens. Fortunately, transformers address this issue through the QKV self-attention mechanism. Specifically, the key vectors K provide a meaningful representation of each token, tailored for dot product similarity. Therefore, we will use the key vectors in the **DBDPC** algorithm. Formally, we have:

$$C_{\text{centers}}, C_{\text{elements}} = \text{DBDPC}(K') \tag{7}$$

where $K' = \{\mathbf{u}_i \in K \mid i \in S_{\text{important}}\}$ is the subset of keys consisting of elements with indices in $S_{\text{important}}$.

What about unimportant tokens near cluster centers? Tokens initially deemed unimportant but close enough to cluster centers have a high probability of being mislabeled. We add these tokens to the corresponding cluster to limit information loss. Formally, we define a threshold based on a coefficient α, where any token $\mathbf{u}_i$, initially excluded, is added to the cluster of the closest center $s \in C_{\text{centers}}$ if its distance to the center satisfies $d_{is} < \alpha \cdot d_c$. Specifically, the new cluster elements set $C^{(s)}_{\text{elements}}$ is updated as follows:

$$S^{(s)}_{\text{added}} = \{i \in S_{\text{unimportant}} \mid s = argmin_{s' \in C_{\text{centers}}} d_{is'} \text{ and } d_{is} < \alpha \cdot d_c\} \tag{8}$$

$$C^{(s)}_{\text{elements}} \leftarrow C^{(s)}_{\text{elements}} \cup S^{(s)}_{\text{added}} \tag{9}$$

Algorithm 3 PACT

Input: Hidden states $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_n] \in \mathbb{R}^{n \times d}$; key and query matrices $\mathbf{K}, \mathbf{Q} \in \mathbb{R}^{n \times n_h \times d_h}$; position IDs $\mathbf{P} = [p_1, \ldots, p_n]$; pruning percentage $\lambda \in [0,1]$; cutoff distance $d_c > 0$; tolerance coefficient $\alpha > 0$

Output: Merged hidden states $\mathbf{H}'$; new position IDs $\mathbf{P}'$

Step 1: Identify important and unimportant tokens
$S_{\text{important}}, S_{\text{unimportant}} \leftarrow \text{EUTI}(\mathbf{H}, \mathbf{K}, \mathbf{Q}, p)$
Step 2: Cluster important tokens with DBDPC
$\mathbf{K}' \leftarrow \{\mathbf{k}_i \in \mathbf{K} \mid i \in S_{\text{important}}\}$
$C_{\text{centers}}, C_{\text{elements}} \leftarrow \text{DBDPC}(\mathbf{K}', d_c)$
Step 3: Assign unimportant tokens to sufficiently close clusters.
for all $i \in S_{\text{unimportant}}$ **do**
  $s_i \leftarrow argmin_s d_{is}$
  if $d_{is_i} < \alpha . d_c$ **then**
    $C^{(s_i)}_{\text{elements}} \leftarrow C^{(s_i)}_{\text{elements}} \cup \{i\}$
  end if
end for
Step 4: Merge hidden states and assign position IDs
for all $s \in C_{\text{centers}}$ **do**
  $\mathbf{h}'_s \leftarrow \frac{1}{|C^{(s)}_{\text{elements}}|} \sum_{i \in C^{(s)}_{\text{elements}}} \mathbf{h}_i$
  $p'_s \leftarrow p_s$
end for
Return $\mathbf{H}'$, $\mathbf{P}'$

Merging Finally, the hidden states corresponding to the elements in each cluster are merged. Formally, the merged hidden states are computed as:

$$\mathbf{H}' = \left\{ \frac{1}{|C^{(j)}_{\text{elements}}|} \sum_{i \in C^{(j)}_{\text{elements}}} \mathbf{h}_i \;\middle|\; C^{(j)}_{\text{elements}} \in C_{\text{elements}} \right\} \tag{10}$$

Defining the position IDs It is important to accurately assign position IDs to each vector in the new hidden states $\mathbf{H}'$, especially for models using Rotary embeddings, as these IDs determine the input image structure or the temporal dependencies of the input video. In order to achieve a low statistical discrepancy compared to regular inference, we assign the position ID for each vector from H' as its corresponding cluster center. The full **PACT** pipeline is illustrated in Algorithm 3. We note that both **DBDPC** and **EUTI**, as well as **PACT**, do not utilize textual tokens. Therefore, the reduction of visual tokens is performed independently of the textual context, making our method well-suited for multi-turn conversations.

Proportional attention Merging tokens reduces their influence in the attention mechanism and can therefore deteriorate performance if many important tokens are merged together. To address this, we use proportional attention [7], which we explain in Appendix E.

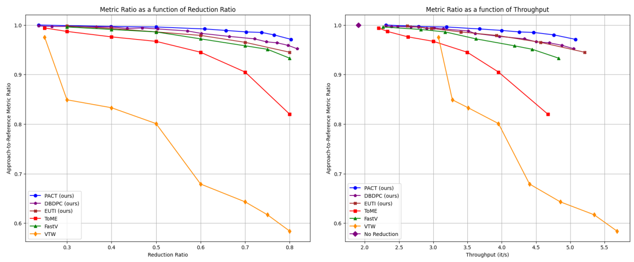


Figure 4. **Comparison between PACT, DBDPC, and EUTI against other visual token reduction methods across various reduction ratios applied on LLaVA-OneVision-7B.**

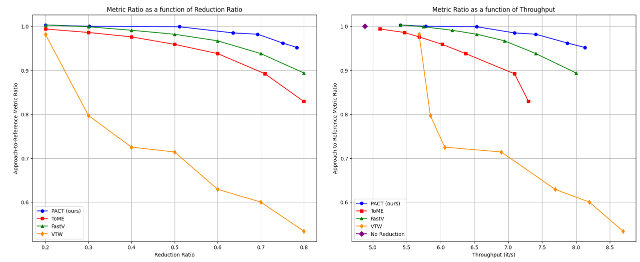


Figure 6. **Comparison between PACT and other visual token reduction methods across various reduction ratios applied on Qwen2-VL-7B-Instruct.**

Selecting the layer L for token reduction: To ensure maximum computational gain, we must choose an early layer L for visual token reduction. However, we also require that the keys at the selected layer are not too similar, allowing for effective clustering and pruning. Thus, we select the earliest layer where the maximum distance between keys is sufficiently high. Figure 3 shows that in the initial layers of LLaVA-OneVision-7B, the keys corresponding to visual tokens are quite similar, indicating a lack of distinctive features necessary for effective pruning and clustering.

4. Experiments

4.1. Evaluation datasets

We evaluate the effectiveness of **PACT** using diverse benchmarks, similar to those used for LLaVA-OneVision-7B, covering single-image, multi-image, and video tasks. We use AI2D [23], TextVQA [44], ChartQA [36], DocVQA [37], and InfographicVQA [38] to assess **PACT**'s ability to reduce visual tokens while maintaining performance in text-rich documents. To test reasoning across multiple disciplines, we use MME [16], MMBench [51], MMVet [49], MathVerse [56], MathVista [33], MMMU [52], MMStar [9], and ScienceQA [32]. Additionally, Vibe-Eval [39], MM-LiveBench [6], and LLaVA-Bench-Wilder [26] evaluate its robustness in real-world scenarios and visual chat contexts. We use LLaVA-Interleave Bench [26] and MuirBench [46] to examine **PACT**'s efficiency in token reduction while preserving inter-image reasoning. To assess performance in video comprehension tasks, we use ActivityNet-QA [50], MLVU [57], VideoMME [17], EgoSchema [35], and PerceptionTest [40]. Finally, Video-ChatGPT [34] evaluates the method's effectiveness in dialogue-based video interaction.

4.2. Evaluation setup

In our comparison, we include approaches where the reduction is applied at a single layer, similar to **PACT**, such as FastV and clustering-based visual token reduction. For these approaches, we refer to the reduction ratio as the relative reduction in the number of visual tokens, defined as $1 - \frac{\text{number of visual tokens after reduction}}{\text{number of visual tokens before reduction}}$. For all these approaches, we use the same value of L and vary hyperparameters to test across different reduction ratios. For methods that use progressive token reduction, like ToME [7], or apply reduction after the visual encoder, as PruMerge and HiReD, or when the reduction ratio cannot be controlled at a fixed layer, such as VTW, we adjust the parameters of these approaches to achieve the same average number of visual tokens across all layers as the one-layer reduction methods for a given reduction ratio. When evaluating clustering algorithms for visual token reduction, we apply proportional attention, as it consistently improves performance across all clustering algorithms, especially at high reduction ratios. Additionally, it is crucial to correctly assign position IDs to the resulting reduced set of visual tokens. Details on the assignment strategy are presented in Appendix F.

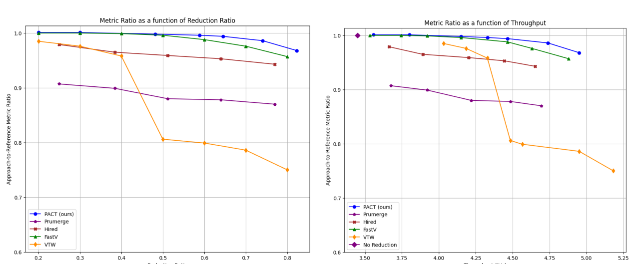


Figure 5. **Comparison between PACT and other visual token reduction methods across various reduction ratios applied on LLaVA-1.6-Mistral-7B.**

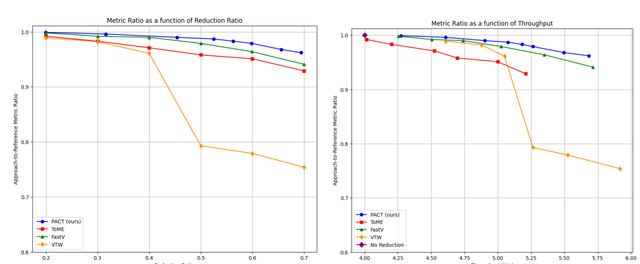


Figure 7. **Comparison between PACT and other visual token reduction methods across various reduction ratios applied on InternVL2-8B.**

Table 2. **Comparison of PACT with FastV, VTW, and ToME on LLaVA-OneVision-7B. Algo. Time** refers to the average time the algorithm takes per input element, measured in seconds. **Proc. Time** refers to the average time taken by both the language model and the reduction algorithm per input element. **Red. Ratio** stands for average Reduction Ratio. The Algo. Time for VTW is nearly zero, and thus omitted. The different visual token reduction methods are evaluated at the same reduction ratio as **PACT**.

Dataset	No reduction		PACT (Ours)				FastV			VTW		ToME		
	Metric	Proc. Time	Metric	Red. Ratio	Proc. Time	Algo. Time	Metric	Proc. Time	Algo. Time	Metric	Proc. Time	Metric	Proc. Time	Algo. Time
VideoMME	58.5	0.792	**57.7**	69.2%	0.321	0.021	56.5	0.323	0.040	46.9	0.269	56.2	0.375	0.091
MME	1579	0.554	1571.0	72.1%	0.226	0.017	**1589.9**	0.232	0.016	944.0	0.208	1520.8	0.299	0.084
DocVQA	87.2	1.088	**85.4**	71.1%	0.467	0.026	84.0	0.495	0.051	14.6	0.402	59.1	0.542	0.099
MLVU	65.2	0.795	**64.8**	69.2%	0.322	0.022	62.3	0.322	0.040	54.4	0.267	63.7	0.383	0.092
LLaVA-Interleave	64.1	0.249	**62.2**	72.2%	0.133	0.010	58.5	0.139	0.007	32.6	0.123	49.3	0.192	0.068
ChartQA	79.9	0.671	**77.3**	71.4%	0.309	0.019	76.6	0.312	0.016	16.7	0.277	58.5	0.376	0.082
MMBench	80.6	0.249	**79.9**	72.0%	0.134	0.010	79.2	0.139	0.005	58.9	0.126	79.7	0.191	0.066
MuirBench	42.0	0.384	**42.4**	70.9%	0.175	0.013	40.7	0.177	0.009	35.1	0.158	40.5	0.232	0.072
ScienceQA	95.9	0.238	93.5	72.0%	0.130	0.010	91.6	0.134	0.006	80.7	0.121	**94.6**	0.186	0.066
MMMU	49.2	0.139	**48.8**	72.6%	0.103	0.007	47.7	0.106	0.003	44.2	0.093	47.4	0.161	0.062
AI2D	81.5	0.382	**81.2**	72.5%	0.173	0.013	79.4	0.181	0.014	69.5	0.157	79.8	0.234	0.073
InfographicVQA	66.0	0.895	**61.5**	70.0%	0.403	0.023	55.9	0.403	0.040	30.1	0.336	44.2	0.547	0.130
MMStar	62.0	0.297	59.5	72.3%	0.147	0.011	56.7	0.152	0.007	39.1	0.165	**60.4**	0.229	0.069
ActivityNetQA	54.5	0.921	**55.1**	70.0%	0.409	0.029	53.4	0.420	0.042	36.6	0.391	54.5	0.510	0.203
MM-LiveBench	73.1	4.434	**71.4**	71.5%	3.203	0.047	61.3	3.201	0.044	41.0	3.072	69.4	3.604	0.102
LLaVA-Wilder	71.0	10.10	**72.0**	71.5%	8.260	0.035	71.3	8.260	0.025	51.0	7.512	69.4	7.920	0.085
MathVerse	16.8	0.831	**17.1**	76.0%	0.350	0.021	15.8	0.372	0.036	17.0	0.289	17.0	0.543	0.150
MathVista	63.3	0.440	**62.1**	73.0%	0.260	0.015	59.9	0.262	0.016	39.6	0.249	56.8	0.329	0.071
MMVet	58.0	4.602	**58.3**	72.3%	3.750	0.035	52.2	3.752	0.036	15.7	3.601	47.2	4.110	0.212
Vibe-Eval	41.6	5.153	**39.2**	72.5%	3.702	0.032	38.0	3.710	0.047	38.0	3.551	31.7	4.315	0.095
VideoChatGPT	3.25	2.972	**3.25**	70.0%	1.850	0.029	3.22	1.862	0.040	1.88	1.320	3.14	1.971	0.205
EgoSchema	60.1	0.811	**60.0**	69.1%	0.320	0.021	58.5	0.326	0.044	44.9	0.277	59.5	0.389	0.091
PerceptionTest	52.1	0.801	**52.3**	70.0%	0.301	0.023	51.7	0.316	0.040	45.0	0.267	51.5	0.373	0.090
TextVQA	75.8	0.690	**75.5**	69.2%	0.320	0.023	**75.5**	0.331	0.029	11.2	0.277	59.7	0.379	0.087

When reporting processing time or throughput, we take into account the total time required by both the language model and the reduction algorithm per input element. In the next section, we base our comparison on a metric called the Approach-to-Reference Metric Ratio, defined as the average of the ratio of the metric of the tested approach to the metric obtained without visual token reduction across all test datasets. Formally we have *Approach-to-Reference Metric Ratio* $= \frac{1}{N}\sum_{i=1}^{N} \frac{\text{Metric}_{\text{with reduction}}(i)}{\text{Metric}_{\text{no reduction}}(i)}$ where N is the total number of test datasets. This metric indicates how much of the original model capacity is retained. It is important to note that when using ToME for visual token reduction, a reduction ratio greater than 50% can't be achieved if the number of visual tokens is reduced by a fixed amount in each layer, as suggested in [7]. Instead, we use a scheduler to achieve higher reduction ratios, which we explain in Appendix G. More details on the hyperparameters used for evaluating **PACT** are provided in Appendix H. We follow the same dataset splits and metrics used for evaluating LLaVA-OneVision wherever feasible. More details are provided in Appendix I. Note that all experiments were conducted on a single A100 GPU.

4.3. Results

We compare **PACT** with FastV [10], VTW [30], ToME [7], PruMerge [43] and HiRED [3] on LLaVA-OneVision-7B, InternVL2-8B, Qwen2-VL-7B-Instruct and LLaVA-1.6-Mistral-7B. Since HiRED and PruMerge are only applicable to LLaVA-1.6, we exclude them from other comparisons. As shown in figures 4, 5, 6, and 7 **PACT** consistently outperforms other methods at both equal reduction ratios and equal throughput across all four models. VTW experiences a significant performance drop for reduction ratios above 40%, indicating that removing all visual tokens is only effective when done in later layers. FastV and ToME struggle at high reduction ratios, while PruMerge and HiRED exhibit degradation even at low reduction ratios. Meanwhile, **PACT** maintains acceptable performance even at high reduction ratios. Tab. 2 shows that **PACT** outperforms other approaches on most test datasets when applied on LLaVA-OneVision-7B. The same conclusion applies to other models, with detailed results provided in Appendix J. In Tab. 1, we report the reduction ratio, throughput, and maximum GPU memory consumption of the different approaches at an equal Approach-to-Reference Metric Ratio of 98.6% on LLaVA-OneVision-7B. **PACT** significantly outperforms the other methods, achieving a re-

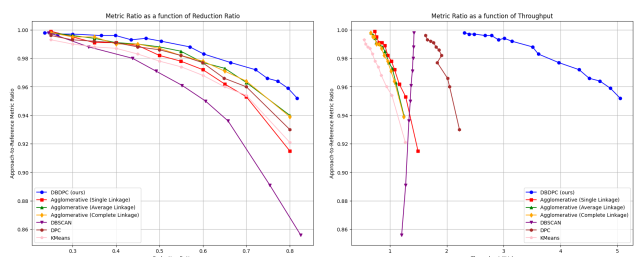


Figure 8. **Comparison of DBDPC and other clustering algorithms for visual token reduction at different reduction ratios on LLaVA-OneVision-7B.**

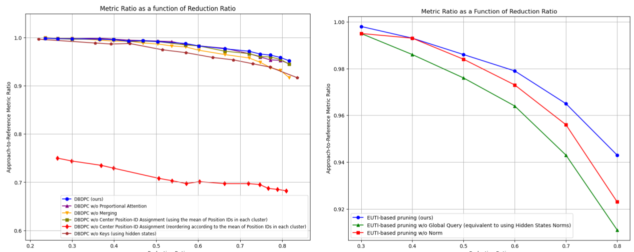


Figure 9. **Ablation study of DBDPC and EUTI on LLaVA-OneVision-7B.**

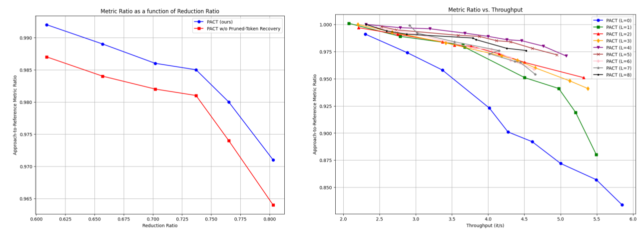


Figure 10. **Ablation study of PACT on LLaVA-OneVision-7B.**

duction ratio of 71.3%, a GPU memory reduction of 31%, and a speedup of 225%. Tab. 1 also indicates that when using FastV, the maximum GPU memory consumption is relatively high due to the costly computation of attention scores. We further compare **DBDPC** against agglomerative clustering [1], k-means [2], Density Peaks Clustering (DPC) [5], and DBSCAN [15], with results presented in Fig. 8. The graphs reveal that **DBDPC** consistently outperforms other clustering algorithms for visual token reduction, exhibiting less performance degradation at equal reduction ratios and demonstrating improved computational efficiency, leading to better throughput. These results validate our hypothesis that, for an effective visual token reduction, it is necessary to ensure that the distances between elements within each cluster do not exceed a predefined threshold. Fig. 4 also shows that **EUTI** consistently outperforms FastV at equal reduction ratios and is less costly, as it does not require the computation of attention scores. In addition, unlike FastV, **EUTI** does not introduce a GPU memory overhead[1]. We provide additional numerical results in Appendix J.

4.4. Ablation study

Fig. 4 shows that **PACT** consistently outperforms both **DBDPC** and **EUTI** across various reduction ratios. This confirms that combining clustering and pruning techniques yields better performance than using each approach independently, as the combined method addresses both visual tokens irrelevance and redundancy. We ablate several components of the **DBDPC** algorithm and present the results in Fig. 9. First, we ablate token merging by selecting the center of each cluster as the representative token instead of merging tokens within each cluster. We also ablate the use of proportional attention. Additionally, we ablate the assignment of position IDs to the reduced set of tokens and experiment with two alternatives: using the mean of position IDs of all elements in each cluster and assigning position IDs sequentially after reordering the reduced set according to the mean of position IDs. Finally, we ablate the use of key vectors in the clustering process and instead use hidden states. Our results show that each ablated component contributes positively to the performance of the **DBDPC** algorithm. Notably, correctly assigning position IDs to the reduced set is crucial, as these position IDs reflect the structure of input images and the temporal order of input videos. Additionally, proportional attention proves effective at higher reduction ratios, while token merging enhances performance once the reduction ratio exceeds 50%. The figure also confirms that keys are better suited for cosine similarity-based distance calculations, as they are naturally used in dot products within the attention mechanism. We perform two separate ablations on Eq. (2) of the **EUTI** algorithm. The first ablation removes the use of hidden state norms, while the second ablates the use of the global query, which corresponds to using only the hidden state norms. The results in Fig. 9 show that combining both the global query-based score and the norm of hidden states consistently leads to better results than using either metric alone, suggesting that they provide complementary information about the importance of each visual token. Finally, we ablate the pruned token recovery module in **PACT** by setting α to zero, with results presented in Fig. 10. The plot shows that reintegrating visual tokens initially deemed unimportant but close enough to a cluster center consistently enhances performance across different reduction ratios, supporting our hypothesis that these tokens were likely mislabeled by the **EUTI** module. Figure 10 also shows the effect of the choice of the reduction layer on **PACT**'s performance, demonstrating the effectiveness of our reduction layer identification approach. We provide additional numerical results in Appendix K.

5. Conclusion

In this work, we presented **PACT**, a method that addresses both visual token irrelevance and redundancy. **PACT** is a plug-and-play solution that does not require additional training. It does not rely on textual tokens for visual token reduction, making it well-suited for multi-turn conversations. Additionally, it operates independently of the visual encoder and connector architecture, making it broadly applicable across various Visual Language Models. Our results confirm that the number of visual tokens in Visual Language Models is unnecessarily large and provide valuable insights for effective token reduction. This opens the door for future work in designing more efficient connectors and architectures for VLMs.

[1] EUTI achieves roughly the same memory reduction as PACT.

6. Acknowledgments

This work received financial support from Crédit Agricole S.A. through the research chair with Ecole Polytechnique on Trustworthy and Responsible AI. This work was granted access to the HPC resources of IDRIS under the allocation 2024-AD011014793R1 made by GENCI.

References

[1] Marcel R Ackermann, Johannes Blömer, Daniel Kuntze, and Christian Sohler. Analysis of agglomerative clustering. *Algorithmica*, 69:184–215, 2014.

[2] Mohiuddin Ahmed, Raihan Seraj, and Syed Mohammed Shamsul Islam. The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics*, 9(8):1295, 2020.

[3] Kazi Hasan Ibn Arif, JinYi Yoon, Dimitrios S Nikolopoulos, Hans Vandierendonck, Deepu John, and Bo Ji. Hired: Attention-guided token dropping for efficient inference of high-resolution vision-language models in resource-constrained environments. *arXiv preprint arXiv:2408.10945*, 2024.

[4] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.

[5] Panthadeep Bhattacharjee and Pinaki Mitra. A survey of density based clustering algorithms. *Frontiers of Computer Science*, 15:1–27, 2021.

[6] Kaichen Zhang* Fanyi Pu* Xinrun Du Yuhao Dong Haotian Liu Yuanhan Zhang Ge Zhang Chunyuan Li Bo Li*, Peiyuan Zhang* and Ziwei Liu. Lmms-eval: Accelerating the development of large multimoal models, 2024.

[7] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

[8] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023.

[9] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*, 2024.

[10] 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. *arXiv preprint arXiv:2403.06764*, 2024.

[11] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024.

[12] Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models. *arXiv preprint arXiv:2311.07919*, 2023.

[13] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness, 2022.

[14] Mohamed Dhouib, Ghassen Bettaieb, and Aymen Shabou. Docparser: End-to-end ocr-free information extraction from visually rich documents. In *International Conference on Document Analysis and Recognition*, pages 155–172. Springer, 2023.

[15] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, pages 226–231, 1996.

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

[17] Chaoyou Fu, Yuhan Dai, Yondong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*, 2024.

[18] Tao Gong, Chengqi Lyu, Shilong Zhang, Yudong Wang, Miao Zheng, Qian Zhao, Kuikun Liu, Wenwei Zhang, Ping Luo, and Kai Chen. Multimodal-gpt: A vision and language model for dialogue with humans. *arXiv preprint arXiv:2305.04790*, 2023.

[19] Jiaming Han, Kaixiong Gong, Yiyuan Zhang, Jiaqi Wang, Kaipeng Zhang, Dahua Lin, Yu Qiao, Peng Gao, and Xiangyu Yue. Onellm: One framework to align all modalities with language. *arXiv preprint arXiv:2312.03700*, 2023.

[20] Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Qiang Liu, et al. Language is not all you need: Aligning perception with language models. *arXiv preprint arXiv:2302.14045*, 2023.

[21] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[22] Yang Jin, Kun Xu, Liwei Chen, Chao Liao, Jianchao Tan, Bin Chen, Chenyi Lei, An Liu, Chengru Song, Xiaoqiang Lei, et al. Unified language-vision pretraining with dynamic discrete visual tokenization. *arXiv preprint arXiv:2309.04669*, 2023.

[23] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *European conference on computer vision*, pages 235–251. Springer, 2016.

[24] Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. Ocr-free document understanding transformer. In *Computer Vision –*

ECCV 2022, pages 498–517, Cham, 2022. Springer Nature Switzerland.

[25] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023.

[26] Bo Li, Kaichen Zhang, Hao Zhang, Dong Guo, Renrui Zhang, Feng Li, Yuanhan Zhang, Ziwei Liu, and Chunyuan Li. Llava-next: Stronger llms supercharge multimodal capabilities in the wild, 2024.

[27] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

[28] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

[29] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. *arXiv preprint arXiv:2202.07800*, 2022.

[30] Zhihang Lin, Mingbao Lin, Luxi Lin, and Rongrong Ji. Boosting multimodal large language models with visual tokens withdrawal for rapid inference. *arXiv preprint arXiv:2405.05803*, 2024.

[31] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023.

[32] Pan Lu, Swaroop Mishra, Tony 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, 2022.

[33] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations (ICLR)*, 2024.

[34] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*, 2024.

[35] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems*, 36:46212–46244, 2023.

[36] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022.

[37] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209, 2021.

[38] Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706, 2022.

[39] Piotr Padlewski, Max Bain, Matthew Henderson, Zhongkai Zhu, Nishant Relan, Hai Pham, Donovan Ong, Kaloyan Aleksiev, Aitor Ormazabal, Samuel Phua, Ethan Yeo, Eugenie Lamprecht, Qi Liu, Yuqi Wang, Eric Chen, Deyu Fu, Lei Li, Che Zheng, Cyprien de Masson d'Autume, Dani Yogatama, Mikel Artetxe, and Yi Tay. Vibe-eval: A hard evaluation suite for measuring progress of multimodal language models. *arXiv preprint arXiv:2405.02287*, 2024.

[40] Viorica Patraucean, Lucas Smaira, Ankush Gupta, Adria Recasens, Larisa Markeeva, Dylan Banarse, Skanda Koppula, Mateusz Malinowski, Yi Yang, Carl Doersch, et al. Perception test: A diagnostic benchmark for multimodal video models. *Advances in Neural Information Processing Systems*, 36, 2024.

[41] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[42] Erich Schubert. *A Triangle Inequality for Cosine Similarity*, page 32–44. Springer International Publishing, 2021.

[43] Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. *arXiv preprint arXiv:2403.15388*, 2024.

[44] 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.

[45] Dingjie Song, Wenjun Wang, Shunian Chen, Xidong Wang, Michael Guan, and Benyou Wang. Less is more: A simple yet effective token reduction method for efficient multimodal llms. *arXiv preprint arXiv:2409.10994*, 2024.

[46] Fei Wang, Xingyu Fu, James Y Huang, Zekun Li, Qin Liu, Xiaogeng Liu, Mingyu Derek Ma, Nan Xu, Wenxuan Zhou, Kai Zhang, et al. Muirbench: A comprehensive benchmark for robust multi-image understanding. *arXiv preprint arXiv:2406.09411*, 2024.

[47] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution, 2024.

[48] 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.

[49] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities, 2023.

[50] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for

understanding complex web videos via question answering. In *AAAI*, pages 9127–9134, 2019.

[51] Yuanhan Zhang Bo Li Songyang Zhang Wangbo Zhao Yike Yuan Jiaqi Wang Conghui He Ziwei Liu Kai Chen Dahua Lin Yuan Liu, Haodong Duan. Mmbench: Is your multi-modal model an all-around player? *arXiv:2307.06281*, 2023.

[52] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of CVPR*, 2024.

[53] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training, 2023.

[54] Duzhen Zhang, Yahan Yu, Chenxing Li, Jiahua Dong, Dan Su, Chenhui Chu, and Dong Yu. Mm-llms: Recent advances in multimodal large language models. *arXiv preprint arXiv:2401.13601*, 2024.

[55] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023.

[56] Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Peng Gao, and Hongsheng Li. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems?, 2024.

[57] Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. Mlvu: A comprehensive benchmark for multi-task long video understanding. *arXiv preprint arXiv:2406.04264*, 2024.