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

A. On the density peaks clustering algorithm

Density Peak Clustering (DPC) is a clustering algorithm that identifies cluster centers based on local density and the distance to points with higher density, denoted as δ_i. The density, ρ_i, can be measured by counting the number of points within a cutoff distance d_c from $\mathbf{u}_i$, or by using a Gaussian function where nearby points contribute more to the density, $\rho_i = \sum_j \exp\left(-\left(\frac{d_{ij}}{d_c}\right)^2\right)$. Points with high ρ_i and δ_i values are selected as cluster centers. This selection can be done by defining a threshold t and designating points as cluster centers where $\rho_i \cdot \delta_i \geq t \times \max(\rho_i \cdot \delta_i)$, or by selecting a fixed percentage. Other points are then assigned to the cluster of the nearest higher-density point, iterating from the highest to the lowest density. This process can create clusters of varying shapes, where the maximum distance between elements within a cluster can be extremely large. In extreme cases, the two farthest points in the input data can end up in the same cluster.

B. DBDPC Characteristics

This section aims to prove that **DBDPC** guarantees that: Each element's distance to its assigned cluster center is at most d_c and that all cluster centers are at least d_c apart.
Assume, for contradiction, that at least one of the following statements is false:

1. There exists an element i assigned to a cluster such that its distance to the cluster center is greater than d_c, i.e., $d_{is} > d_c$.
2. There exist two cluster centers s_1, s_2 such that their pairwise distance is at most d_c, i.e., $d_{s_1 s_2} \leq d_c$.

Contradiction for Assumption 1 In **DBDPC**, each element i is assigned to its closest cluster center:

$$s_i = \arg\min_{s \in C_{\text{centers}}} d_{is}.$$

If $d_{is} > d_c$ for a given center s, then we have $d_{is'} > d_c$ for all centers. However, in the **DBDPC** selection process, an element is assigned as a cluster center if its minimum distance to already selected centers is over d_c. Thus, i should have been selected as a new cluster center, and its distance to the closest cluster center would be zero, which leads to a contradiction, proving that every element satisfies $d_{is} \leq d_c$.

Contradiction for Assumption 2 Assume, without loss of generality, that s_2 is chosen after s_1. By the center selection criterion, a new center s_2 is added only if:

$$\min_{s \in C_{\text{centers}}} d_{s_2 s} > d_c.$$

If $d_{s_1 s_2} \leq d_c$, then s_2 shouldn't be selected as a cluster center, which leads to a contradiction. Thus, no two centers can be closer than d_c.
Inter-cluster distance upper-bound : Here we will refer to cosine similarity by sim. Let's x and y be two points in the same cluster, and s their cluster center. Since each point $\mathbf{x}$ is within d_c of its cluster center $\mathbf{s}$ and the distance used in the **DBDPC** algorithm is $1 - \text{sim}$, we have $\text{sim}(\mathbf{x}, \mathbf{s}) \geq 1 - d_c$. We have from [42]:

$$\text{sim}(\mathbf{x}, \mathbf{y}) \;\geq\; \text{sim}(\mathbf{x}, \mathbf{s}) \cdot \text{sim}(\mathbf{s}, \mathbf{y}) \;+\; m - 1,$$

$$\text{where } m = \min\Big\{\text{sim}(\mathbf{x}, \mathbf{s})^2,\; \text{sim}(\mathbf{s}, \mathbf{y})^2\Big\}.$$

Using $\text{sim}(\mathbf{x}, \mathbf{s}), \text{sim}(\mathbf{s}, \mathbf{y}) \geq 1 - d_c$ we get

$$\text{sim}(\mathbf{x}, \mathbf{y}) \;\geq\; (1-d_c)^2 + (1-d_c)^2 - 1 \;=\; 1 - 2\,d_c\,(2-d_c).$$

Finally, converting this back to the distance $d(\mathbf{x}, \mathbf{y}) = 1 - \text{sim}(\mathbf{x}, \mathbf{y})$, we obtain:

$$d(\mathbf{x}, \mathbf{y}) \;\leq\; 2\,d_c\,(2 - d_c).$$

Therefore, the intra-cluster distance in the **DBDPC** algorithm is bounded by $2\,d_c\,(2 - d_c)$.

C. A comparison between DBDPC and other clustering algorithms

Comparison between DBDPC and DPC: We note that, aside from using densities, **DBDPC** is fundamentally different from DPC. Please refer to Appendix A for a detailed explanation of the **DPC** algorithm. The center identification process in **DBDPC** results in two main characteristics with formal proof detailed in Appendix B. First, the distance between each element and its cluster center is below d_c, which leads to inter-cluster distances being upper-bounded by $2d_c \times (2 - d_c)$. Additionally, the distance between cluster centers is lower-bounded by d_c. These guarantees do not hold for DPC, leading to two drawbacks. Since inter-cluster distances are not controlled, merging these vectors may result in merging highly dissimilar vectors, leading to information loss. Also, in high-density regions, the distance between cluster centers becomes too small, making DPC ineffective in addressing information redundancy.
A Qualitative comparison Figure 11 presents the clustering results for **DBDPC**, DPC, DBSCAN, and K-Means on a

Algorithm 4 Recursive Center Identification for DBDPC with Iterative Center Identification

Input: Cutoff distance $d_c \in \mathbb{R}^+$, set of vectors $\mathbf{U} = \{\mathbf{u}_i \in \mathbb{R}^{d_l}\}_{i=1}^n$, density values $\{\rho_i\}_{i=1}^n$, distance matrix $D = [d_{ij}]$, fallback threshold $T > 0$

Output: Cluster center indices C_{centers}

Initialize cluster center set $C_{\text{centers}} = \emptyset$

Set the density of each point :

$$\rho_i = \text{argsort}(\{-\rho_j\}_{j=1}^n)[i]$$

while $\mathbf{U} \neq \emptyset$ **do**

Compute δ_i for all vectors $\mathbf{u}_i \in \mathbf{U}$:

$$\delta_i = \min_{\rho_j > \rho_i} d_{ij}$$

Select cluster candidates:

$$\mathbf{C}_{\text{new}} = \{\mathbf{u}_i \in \mathbf{U} \mid \delta_i > d_c\}$$

$C_{\text{centers}} \leftarrow C_{\text{centers}} \cup \mathbf{C}_{\text{new}}$

Update remaining vectors:

$$\mathbf{U} \leftarrow \mathbf{U} \setminus \left(\mathbf{C}_{\text{new}} \cup \left\{ \mathbf{u}_k \in \mathbf{U} \mid \begin{array}{c} \exists \mathbf{u}_i \in \mathbf{C}_{\text{new}} \\ \text{such that } d_{ik} \leq d_c \end{array} \right\} \right)$$

if $|\mathbf{C}_{\text{new}}| < T$ **then**

Order remaining vectors $\mathbf{U}$ by decreasing ρ_i:

$\mathbf{U} \leftarrow \text{Sort}(\mathbf{U}, \text{key} = \rho_i, \text{order} = \text{descending})$

Call Iterative Center Identification:

$C_{\text{centers}} \leftarrow \text{IterativeCenterIdentification}(C_{\text{centers}}, \mathbf{U}, d_c)$

return C_{centers}

end if

end while

return C_{centers}

Function: Iterative Center Identification

Inputs: Remaining vectors $\mathbf{U}$ (ordered by ρ_i), current cluster center set C_{centers}, cutoff distance d_c

Outputs: Updated cluster center indices C_{centers}

for all $\mathbf{u}_i \in \mathbf{U}$ **do**

if $\min_{\mathbf{u}_s \in C_{\text{centers}}} d_{is} > d_c$ **then**

$C_{\text{centers}} \leftarrow C_{\text{centers}} \cup \{\mathbf{u}_i\}$

end if

end for

return C_{centers}

predefined set of two-dimensional points. The figure shows that only **DBDPC** and DBSCAN identify isolated points as distinct clusters, a crucial feature for visual token reduction, as these points contain unique and thus potentially valuable information. We note that, for DBSCAN, these isolated points may be identified as noise, depending on the chosen hyperparameters. Moreover, **DBDPC** partitions both the left and right groups of points into the same number of clusters, maintaining consistency despite the higher density on the left side. In contrast, DPC tends to form a greater number of clusters in high-density regions while creating large clusters in low-density areas, whereas DBSCAN follows the opposite pattern, producing large clusters in high-density regions. In the context of visual token reduction, merging points within these large clusters can result in information loss, leading to performance degradation and making DPC and DBSCAN less suitable than **DBDPC** for this task. We note that the results presented in Fig. 11 for DPC and DBSCAN may change when modifying the hyperparameters; however, the characteristics discussed above persist across different hyperparameter choices.

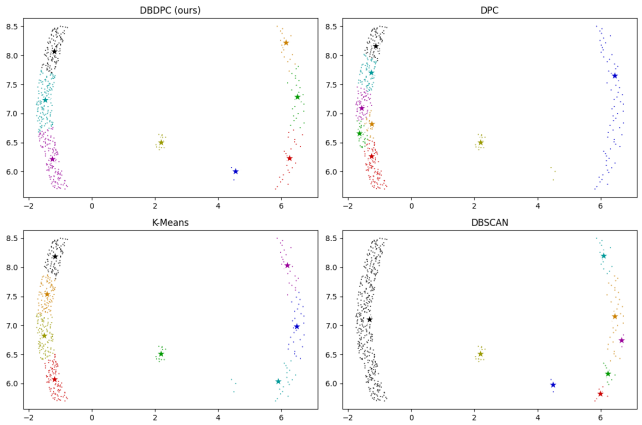

Figure 11. **An illustrative example of the difference in clustering characteristics between DBDPC and other clustering algorithms.** Two-dimensional points and the Euclidean distance were used for illustration purposes.

D. Efficient center identification in DBDPC

D.1. A recursive approach

To enhance the efficiency of the **DBDPC** algorithm, we introduce a recursive center identification method that reduces computational overhead while maintaining clustering accuracy. In the **DBDPC** algorithm, vectors are processed in descending order of their local densities ρ_i, and a vector $\mathbf{u}_i$ is selected as a cluster center if it is farther than the cutoff distance d_c from all previously selected centers. Implementing this as described in the algorithm requires sequentially iterating through all the vectors and checking distances to all previously selected centers, which does not fully leverage GPU parallelization capabilities. In the **DBDPC** algorithm, when two points have the same density, one is treated as if it has a higher density than the other, depending on the order of their processing. To replicate this behavior, we assign the

density of each point to its rank as:

$$\rho_i = \text{rank}_i = \text{argsort}(\{-\rho_j\}_{j=1}^{n})[i]$$

Our accelerated method leverages the quantity δ_i, representing the minimum distance from vector $\mathbf{u}_i$ to any higher-density vector:

$$\delta_i = \min_{\rho_j > \rho_i} d_{ij} \tag{11}$$

If $\delta_i > d_c$, then $\mathbf{u}_i$ is selected as a cluster center because it is not within d_c of any higher-density vector, which are the only potential cluster centers that can be selected before d_{ij} in the **DBDPC** algorithm. In addition, any vector within d_c of a cluster center identified using δ_i has a lower density than that center, as cluster centers identified using δ_i are not within d_c of any higher-density vector. In the **DBDPC** algorithm, such a vector would not be chosen as a cluster center because it violates the distance condition relative to already selected centers. By identifying these vectors early, we can exclude them from further consideration as potential centers. We repeat this process recursively: after selecting cluster centers where $\delta_i > d_c$ and excluding vectors within d_c of these centers, we process the remaining vectors. This recursion continues until the number of newly discovered cluster centers becomes small (e.g., less than 10). At that point, we fall back to the **DBDPC** method, processing the remaining vectors iteratively to ensure all potential centers are considered. This recursive approach reduces the number of iterations in the main loop and enhances parallelization, particularly on GPUs, by minimizing sequential computation. By leveraging δ_i and incorporating an early exclusion mechanism, the recursive center identification method reduces computational time while ensuring the same clustering results as the **DBDPC** algorithm. The recursive approach decreases the number of iterations and enhances GPU parallelization by minimizing sequential computation, making the algorithm more efficient for large datasets. The recursive center identification method is presented in Algorithm 4. We note that in practice this recursive approach reduce the computational time of the **DBDPC** algorithm by around 3 times.

D.2. Proof of correctness of the recursive approach

To validate the correctness of the accelerated method, we demonstrate the following key points: selected centers are valid cluster centers, excluded vectors are not cluster centers and identifying remaining cluster centers is equivalent to identifying cluster centers on the reduced set. Proving these points suffices to establish correctness, as the remaining vectors after the recursive steps are treated the same as in the **DBDPC** algorithm.

Selected Centers Are Valid Cluster Centers In the **DBDPC** algorithm, for any vector $\mathbf{u}_i$, only vectors with higher densities are considered for selection as cluster centers before $\mathbf{u}_i$. If $\mathbf{u}_i$ is not within d_c of any higher-density vector (i.e., $\delta_i > d_c$) then the distance of $\mathbf{u}_i$ from any previously selected center cannot exceed the cutoff distance d_c. Consequently, $\mathbf{u}_i$ satisfies the condition for being a cluster center in the **DBDPC** algorithm, as it is farther than d_c from all centers processed earlier.

Excluded Vectors Are Not Cluster Centers Vectors within d_c of a cluster center identified using δ_i have lower densities than that center, as these centers are not within d_c to any higher density point. In the **DBDPC** algorithm, such vectors would not be selected as cluster centers because they are within d_c to an already selected center, violating the distance condition. Therefore, excluding these vectors early does not affect the selection of valid cluster centers.

Identifying Remaining Cluster Centers is Equivalent to Identifying Cluster Centers on the Reduced Set After selecting cluster centers where $\delta_i > d_c$ and excluding vectors within d_c of these centers, we focus on the reduced set of remaining vectors for further processing. The critical observation is that the previously selected cluster centers are not within d_c of any vector in the reduced set. This is ensured by the exclusion step, where all vectors within d_c of these centers have been removed. Consequently, when identifying new cluster centers within the reduced set, we do not need to consider distances to the previously selected centers, as they cannot influence the selection due to their distance. Moreover, the vectors that have been excluded are not potential cluster centers themselves. Meaning that they can not influence the center selection process. This means that any vector satisfying $\delta > d_c$ in the reduced set, is actually not within d_c to any higher density potential cluster center form the initial set, making it a cluster center.

E. Proportional attention

Token merging reduces their impact within the attention mechanism, potentially degrading performance. To mitigate this, we employ proportional attention. Let K, Q, and V denote the keys, queries, and values at a layer L', where $L' \geq L$. For each attention head j, the attention scores are calculated as follows:

$$A^{(j)} = \text{softmax}\left(\frac{Q^{(j)}K^{(j)\top}}{\sqrt{d_{l'}}} + \log \mathbf{W} + \mathbf{B}\right) \tag{12}$$

where $d_{l'}$ is the dimensionality of the query for each attention head. Here, $\mathbf{W}$ is a matrix representing the weight of each token, and $\mathbf{B}$ is the attention mask. Specifically, for visual tokens, w_{i_0,i_1} represents the size of the cluster corresponding to token i_1, for any value of i_0. For each textual token at position t, $w_{i_0,t} = 1$, as they remain unmerged, retaining a weight of one. By scaling the attention scores based on $\mathbf{W}$, the model effectively treats each visual token as if it represents multiple tokens. We note that when

using proportional attention, we use PyTorch's scaled dot-product attention[2], which produces similar results to the official FlashAttention implementation while supporting custom masks.

F. On the choice of Positional IDs for clustering algorithms

In our work, we benchmark four clustering algorithms: agglomerative clustering [1], k-means [2], Density Peaks Clustering (DPC) [5], and DBSCAN [15]. For each algorithm, we use the key vectors for clustering, apply a cosine similarity-based distance (as in **DBDPC**), and evaluate two strategies: merging the hidden states within each cluster or selecting the cluster center as a representative token. We report the best-performing approach for each algorithm. Similar to **DBDPC**, we assign the position ID of the cluster center to the resulting vectors. However, apart from DPC, the other clustering algorithms do not explicitly provide a cluster center. For k-means and agglomerative clustering, we select the cluster center as the point closest to the average of all points in the cluster, using keys and cosine similarity. For DBSCAN, we experimented with choosing the point connected to the most other points within the cluster and found this approach to yield slightly better results, aligning better with the principles of DBSCAN. Thus, we adopted this strategy in our tests.

G. More about applying ToME to Visual Language Models

ToMe reduces the number of visual tokens at each layer of the transformer. For a given layer i, the process starts by splitting the tokens into two distinct sets, A and B. Each token in set A is matched with its most similar counterpart in set B, using cosine similarity based on key vectors to determine the closest pairs. The top r_i pairs with the highest similarity are then selected for merging. Connected components from the matched pairs are combined into single vectors, where hidden states are averaged. It is important to note that each connected component contains exactly one element from set B, and when applying ToME to Visual Language Models, this element's position ID is assigned to the merged token. In [7], the number of visual tokens was reduced by a fixed quantity ($r_i = r$). However, this fixed reduction scheme cannot achieve more than a 50% reduction unless no reduction is done at later layers when the number of tokens drops below r, which goes against the gradual reduction strategy proposed in ToMe. To enable higher reduction ratios, we adopt a linearly decreasing scheduler, where the reduction is higher in early layers and decreases in later layers. This approach achieves a smaller average number of visual tokens across the network while still reducing the token count at each layer, allowing us to reach high reduction ratios effectively.

[2] https://pytorch.org/docs/stable/generated/torch.nn.functional.scaled_dot_product_attention.html

H. Implementation details and hyperparameters for PACT

For all experiments on LLaVA-OneVision-7B, we set $d_n = 2$, $\alpha = 1.5$, and $L = 4$. While the optimal values of each parameter may vary depending on the dataset, we aim to evaluate the real-world effectiveness of our approach by using consistent values across all testing datasets. The results in Tab. 2 were obtained using $d_c = 0.17$ and $\lambda = 0.4$. Additionally, to demonstrate the performance of our approach at different reduction ratios, we vary d_c and λ and report the results. The values of the fixed parameters d_n and α were chosen by performing a grid search on SeedBench [25], which is why we do not include SeedBench in the testing datasets. It is important to note that finding the optimal parameters for all testing datasets is not the focus of this study, as this would require extensive testing of different values for d_c, λ, L, α, and d_n on all test sets. Such an approach would not accurately reflect the real-world performance of our method. Instead, we chose to only vary d_c and λ to evaluate the effectiveness of our approach at different reduction ratios. When testing on SeedBench, we found that a pruning ratio higher than 60% harms performance. Therefore, we vary the pruning ratio between 10% and 60% and test across different values of d_c. When testing **PACT** on LLaVA-1.6-Mistral-7B, Qwen2-VL-7B-Instruct and InternVL2-8B. We use the same values of d_n and α as when testing on LLaVA-OneVision-7B. We note that these hyperparameters may not be optimal; however, as we aim to test the generalizability of our approach, we opt to use the same hyperparameters across models. Figure 12, Figure 13 and Figure 14 show the maximum distance between the keys at several layers of the language model for LLaVA-1.6-Mistral-7B, Qwen2-VL-7B-Instruct and InternVL2-8B. Following the same approach for LLaVA-OneVision-7B, we choose $L = 4$ for Qwen2-VL-7B-Instruct and $L = 7$ for InternVL2-8B. We note that the choice of the reduction layer for InternVL2-8B is not as evident as for LLaVA-OneVision-7B and Qwen2-VL-7B-Instruct, as the increase in maximum distance from one layer to the next is sometimes minimal, making it unclear which layer offers the best balance between accuracy and computational efficiency. However, since we do not aim to experimentally determine the optimal reduction layer, we end up choosing $L = 7$, as the maximum distance between keys is increased by an acceptable amount between the seventh and eighth layer. Following the same approach we use $L = 7$ for LLaVA-1.6-Mistral-7B.

I. More about test datasets and used metrics

For evaluating the different approaches, we use LMMs-Eval [6] and aim to follow the same dataset splits and metrics as used in [27]. We detail the used splits and metrics in Tab. 3. Some datasets require evaluation using a GPT model through the OPENAI API or other closed-source models. However, for many datasets the version of the closed-source model used in evaluating LLaVA-OneVision in [27] is no longer available. So we use the latest version of GPT-4 for our assessments at the time of publication (gpt-4o-2024-08-06). We also observed that when calling a closed-source model like GPT-4 via an API, the responses are not fully deterministic, even with a temperature set to zero, introducing some noise into the evaluation metrics. To reduce this noise, we exclude all these datasets when testing across different reduction ratios. On the other hand, for Tab. 1, we exclude MMVet, Vibe-Eval, VideoChatGPT, MM-LiveBench, and LLaVA-Wilder as they have high inference times, which would dominate the throughput calculation.

For certain datasets, such as DocVQA, InfoVQA, and TextVQA, we use the validation split contrary to [27]. This choice allows us to test various reduction ratios and approaches without requiring submission to the test server, which would be impractical for extensive testing. For datasets requiring a test set submission (EgoSchema and PerceptionTest), where either the validation set is typically not used for evaluation or does not exist, we report the submission-based metrics evaluated directly on the test set. As explained above, for some datasets our evaluation setup differs from the one used for evaluating LLaVA-OneVision in [27], which may result in variations in the reported results for this model on certain datasets. This is primarily due to the use of validation splits for DocVQA, InfoVQA, and TextVQA, as well as the reliance on GPT-based metrics for some datasets (a common practice for these benchmarks, making alternative evaluation difficult). Nevertheless, our comparisons remain fair, as the same evaluation procedure is consistently applied across all approaches and reduction ratios. Notably, when testing on Qwen2-VL-7B-Instruct without reduction, some datasets encountered GPU out-of-memory errors (MLVU, VideoMME, and ActivityNet Perception) which we excluded from the test set. Additionally, results on ScienceQA were quite low when tested without reduction (0.132), leading to its exclusion from testing as well. We note that, as we use LMM-Eval [6] for evaluation, results differ for some datasets from the officially reported results, as prompts are sometimes not formatted in the same manner. This observation also applies to InternVL2-8B.

J. Additional numerical results

Table 7 and Tab. 8 show a comparison of **DBDPC** and various clustering algorithms for a reduction ratio of approximately 60% on LLaVA-OneVision-7B across multiple datasets. The results demonstrate that **DBDPC** outperforms other clustering algorithms in visual token reduction for the majority of the datasets. Additionally, the tables show that the clustering process for **DBDPC** is significantly faster than that of other clustering algorithms. Table 9 presents a comparison of **EUTI**-based visual token pruning and FastV for a reduction ratio of approximately 60% on LLaVA-OneVision-7B across various datasets. The results indicate that **EUTI** outperforms FastV on most datasets while also being more computationally efficient. Table 14 shows that using keys for distance calculations in DBDPC outperforms hidden states across the majority of the test datasets. Also, we present a comparison between **PACT** and other visual reduction techniques for Qwen2-VL-7B-Instruct, InternVL2-8B, and LLaVA-1.6-Mistral-7B across different datasets in Tab. 5, Tab. 4, and Tab. 6.

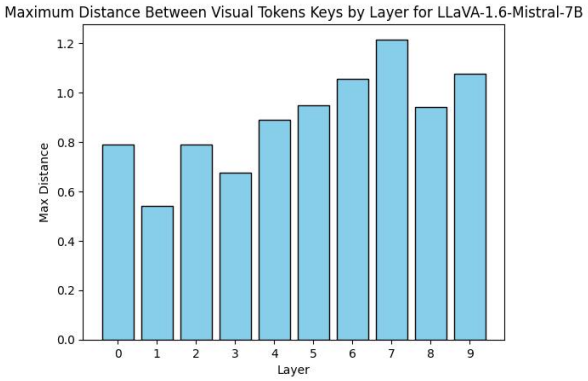

Figure 12. **Illustration of the maximum distance between the keys of visual tokens for the first 10 layers of LLaVA-1.6-Mistral-7B before the application of rotary embeddings.**

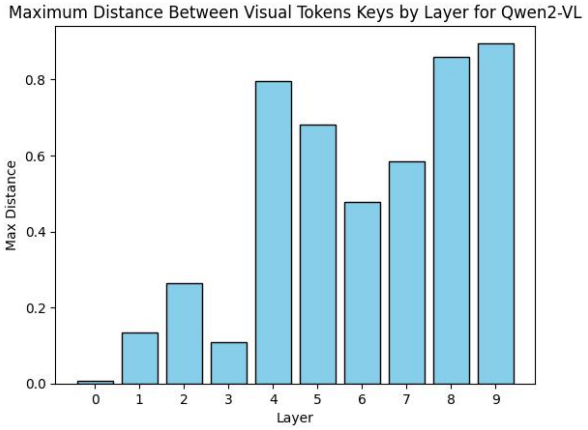

Figure 13. **Illustration of the maximum distance between the keys of visual tokens for the first 10 layers of Qwen2-VL-7B-Instruct before the application of rotary embeddings.**

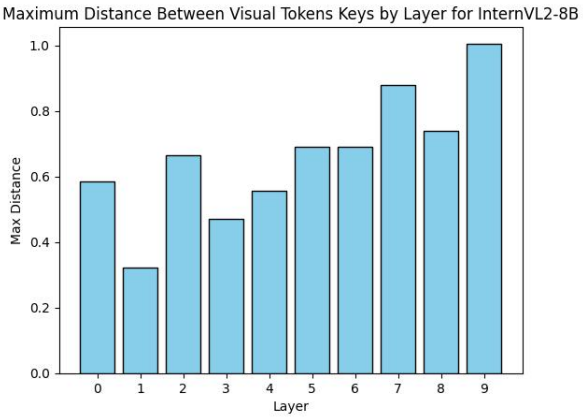

Figure 14. **Illustration of the maximum distance between the keys of visual tokens for the first 10 layers of InternVL2-8B before the application of rotary embeddings.**

K. Ablation study : Additional numerical results

Table 10 shows a comparison between **PACT**, **DBDPC**, and **EUTI** for a reduction ratio of approximately 70%, applied on LLaVA-OneVision-7B. The results demonstrate that **PACT**, which combines both clustering and pruning, outperforms the other two methods that are either clustering-based or pruning-based across various datasets. More importantly, **DBDPC** and **EUTI** exhibit a significant drop in performance on some of the datasets, which is not the case for **PACT**. We note that numerical results for the ablation studies conducted on **DBDPC**, **EUTI**, and **PACT** can be found in Tab. 11, Tab. 12 and Tab. 13.

Table 3. **Dataset Splits, Subsets, and Evaluation Metrics Used in Our Experiments.** Default indicates the use of the standard test split or cases where only one split/subset is available. The evaluation metrics employed are those commonly used for the respective datasets and generally the ones proposed in the official papers. For GPT-based scores (or any model-based scores), this means that a GPT model was used during evaluation, typically to extract answers from the generated output text, which are then matched with the ground truth to calculate accuracy using exact matches. When accuracy is reported, it generally implies that only an exact match is considered a correct answer.

Dataset	Split	Subset	Evaluation Metric
VideoMME	Default	No subtitles	Accuracy
MME	Default	Default	MME Perception Score
DocVQA	Validation	Default	ANLS
MLVU	Default	Default	Accuracy
LLaVA-Interleave	Default	Out-domain	Accuracy
ChartQA	Validation	Default	Relaxed Accuracy
MMBench	Validation	English	GPT-based Score
MuirBench	Default	Default	Accuracy
ScienceQA	Default	Vision only	Accuracy
MMMU	Validation	Default	Accuracy
AI2D	Default	Default	Accuracy
InfographicVQA	Validation	Default	ANLS
MMStar	Default	Default	Accuracy
ActivityNetQA	Default	Default	GPT-based Score
MM-LiveBench	Default	2406	GPT-based Score
LLaVA-Wilder	Default	Small	GPT-based Score
MathVerse	Default	Vision mini	GPT-based Score
MathVista	Default	Testmini	GPT-based Score
MMVet	Default	Default	GPT-based Score
Vibe-Eval	Default	Default	REKA-based Score
VideoChatGPT	Default	Default	GPT-based Score
EgoSchema	Default	Default	Submission
PerceptionTest	Default	Multiple Choice QA	Submission
TextVQA	Validation	Default	Official metric

Table 4. **Comparison of PACT with FastV, VTW, and ToME applied on Qwen2-VL-7B-Instruct across Various Datasets.**

Dataset	No Reduction		PACT (Ours)			FastV		VTW		ToME	
	Metric	Proc. Time	Metric	Red. Ratio	Proc. Time	Metric	Proc. Time	Metric	Proc. Time	Metric	Proc. Time
MME	1654.5	0.238	**1664.7**	86.3%	0.110	1594.3	0.111	1218.5	0.120	1607.5	0.140
DocVQA	93.9	0.516	**90.5**	77.5%	0.294	84.3	0.298	8.7	0.249	67.1	0.350
TextVQA	81.8	0.155	**80.4**	67.5%	0.132	79.6	0.135	14.2	0.118	63.9	0.151
InfographicVQA	74.6	0.478	**70.5**	69.7%	0.278	63.3	0.273	21.5	0.225	43.9	0.299
ChartQA	80.8	0.145	**76.2**	61.1%	0.135	69.4	0.134	16.1	0.123	57.0	0.155
MMBench	77.6	0.074	**77.1**	51.5%	0.077	76.9	0.074	76.9	0.073	76.1	0.080
MuirBench	40.7	0.159	**41.4**	76.9%	0.113	40.5	0.112	38.0	0.111	41.0	0.125
MMMU	51.4	0.109	**51.2**	72.6%	0.093	49.3	0.092	46.7	0.088	48.6	0.105
AI2D	79.9	0.105	**78.4**	64.2%	0.096	76.7	0.097	69.0	0.087	76.7	0.115
MMStar	56.0	0.072	**54.5**	61.3%	0.072	52.6	0.067	40.8	0.065	52.7	0.077
EgoSchema	62.1	0.360	**61.6**	60.0%	0.207	60.2	0.212	46.3	0.190	61.2	0.230
MathVerse	25.3	0.620	**24.5**	82.2%	0.393	23.7	0.396	15.4	0.296	18.1	0.651
MathVista	59.2	0.249	**57.7**	73.3%	0.195	56.4	0.194	35.6	0.165	53.5	0.275
MMVet	24.9	4.700	**25.1**	80.3%	3.820	22.3	3.830	14.5	3.650	16.7	4.780
Vibe-Eval	47.5	3.200	**46.1**	85.0%	2.310	44.3	2.375	28.3	1.993	29.6	3.620
LLaVA-Interleave	35.3	0.120	**35.6**	73.7%	0.100	34.8	0.101	33.4	0.096	33.6	0.125
MM-LiveBench	72.6	3.970	**70.7**	77.1%	3.040	63.0	3.120	43.8	2.970	57.6	4.450

Table 5. **Comparison of PACT with FastV, VTW, and ToME applied on InternVL2-8B on Various Datasets.**

Dataset	No Reduction		PACT (Ours)			FastV		VTW		ToME	
	Metric	Proc. Time	Metric	Red. Ratio	Proc. Time	Metric	Proc. Time	Metric	Proc. Time	Metric	Proc. Time
VideoMME	52.2	0.247	51.1	68.4%	0.151	**51.3**	0.155	51.0	0.142	50.2	0.190
MME	1621.0	0.171	1591.9	69.9%	0.121	1588.7	0.118	**1627.0**	0.111	1533.3	0.155
DocVQA	90.0	0.301	**87.0**	52.1%	0.251	86.2	0.254	52.2	0.229	83.4	0.248
MLVU	50.6	0.439	**49.7**	68.8%	0.326	48.8	0.325	49.5	0.333	29.3	0.343
LLaVA-Interleave	40.0	0.390	39.0	71.2%	0.265	**39.7**	0.263	39.6	0.230	36.7	0.316
ChartQA	82.7	0.221	**81.2**	59.2%	0.184	**81.2**	0.182	47.5	0.175	71.4	0.202
MMBench	81.9	0.161	**80.4**	70.4%	0.118	80.2	0.116	80.2	0.109	70.8	0.165
MuirBench	35.7	0.432	34.4	70.3%	0.249	**35.6**	0.258	33.7	0.210	32.7	0.296
ScienceQA	97.1	0.165	**97.1**	70.8%	0.118	95.8	0.116	95.7	0.109	89.9	0.151
MMMU	48.5	0.167	**48.0**	70.6%	0.126	47.7	0.126	47.8	0.119	47.5	0.156
AI2D	82.5	0.146	**81.4**	70.7%	0.112	78.5	0.110	79.6	0.105	74.4	0.142
InfographicVQA	66.0	0.206	**63.4**	50.7%	0.168	49.8	0.167	25.6	0.157	55.4	0.199
MMStar	59.0	0.179	**56.7**	70.4%	0.186	54.2	0.184	53.4	0.352	55.1	0.156
TextVQA	76.9	0.221	**75.0**	54.5%	0.186	73.9	0.199	61.6	0.194	71.6	0.189
PerceptionTest	57.7	0.300	**56.8**	66.0%	0.203	56.2	0.213	34.1	0.192	55.2	0.228
EgoSchema	54.0	0.240	**53.7**	67.0%	0.155	53.1	0.163	32.2	0.146	52.9	0.172
ActivityNet	51.7	0.240	**51.3**	66.0%	0.153	51.0	0.161	30.8	0.143	50.4	0.171
MM-LiveBench	68.0	3.075	**67.3**	68.0%	2.140	67.0	2.247	40.4	2.003	66.6	2.354

Table 6. **Comparison of PACT with FastV, Prumerge, and Hired applied on LLaVA-1.6-Mistral-7B across multiple datasets.**

Dataset	No Reduction		PACT (Ours)			FastV		Prumerge		Hired	
	Metric	Proc. Time	Metric	Red. Ratio	Proc. Time	Metric	Proc. Time	Metric	Proc. Time	Metric	Proc. Time
MME	1500.0	0.237	**1507.1**	70.3%	0.159	1503.9	0.158	1485.4	0.166	1497.0	0.168
DocVQA	70.0	0.363	**67.1**	67.1%	0.284	64.5	0.281	48.8	0.293	65.8	0.295
ChartQA	52.9	0.332	**49.3**	70.1%	0.259	48.9	0.261	36.0	0.264	46.1	0.266
MMBench	68.2	0.226	**68.0**	71.9%	0.155	67.9	0.154	66.2	0.160	67.6	0.164
ScienceQA	73.0	0.197	72.7	71.5%	0.144	**73.2**	0.145	71.7	0.148	72.9	0.149
MMMU	34.2	0.239	**34.9**	71.5%	0.171	34.7	0.169	33.9	0.180	33.9	0.180
AI2D	67.5	0.233	**67.5**	70.9%	0.160	67.0	0.158	64.5	0.165	65.9	0.166
InfographicVQA	36.9	0.294	**35.6**	66.2%	0.226	33.4	0.229	31.9	0.236	31.6	0.236
MMStar	36.2	0.375	**36.7**	71.9%	0.350	36.6	0.400	35.1	0.345	35.9	0.345

Table 7. **Comparison of DBDPC and Agglomerative Clustering Methods for a Reduction Ratio of approximately 60% on LLaVA-OneVision-7B.**

Dataset	DBDPC (ours)			Agg. (Single Linkage)			Agg. (Average Linkage)			Agg. (Complete Linkage)		
	Metric	Proc. Time	Algo. Time	Metric	Proc. Time	Algo. Time	Metric	Proc. Time	Algo. Time	Metric	Proc. Time	Algo. Time
VideoMME	57.4	0.389	0.040	57.6	1.504	1.148	57.0	1.657	1.316	57.9	1.690	1.350
MME	1563.8	0.255	0.028	1554.1	0.994	0.738	1559.2	1.123	0.868	1563.0	1.151	0.897
DocVQA	84.7	0.530	0.044	83.6	1.899	1.379	84.4	2.185	1.662	84.3	2.308	1.777
MLVU	64.2	0.384	0.039	64.0	1.574	1.229	65.2	1.675	1.329	64.8	1.700	1.355
LLaVA-Interleave	62.1	0.151	0.016	62.0	0.425	0.277	61.5	0.446	0.298	61.4	0.446	0.298
ChartQA	76.0	0.366	0.031	74.5	1.151	0.798	75.8	1.253	0.910	75.8	1.277	0.930
MMBench	80.1	0.151	0.016	79.5	0.427	0.277	79.7	0.437	0.291	79.8	0.449	0.299
MuirBench	43.2	0.215	0.023	41.4	0.667	0.474	42.0	0.727	0.534	42.0	0.738	0.544
ScienceQA	94.7	0.147	0.015	94.8	0.394	0.250	94.7	0.416	0.271	94.7	0.413	0.269
MMMU	48.3	0.110	0.009	48.4	0.218	0.110	49.3	0.232	0.121	48.2	0.225	0.117
AI2D	80.7	0.202	0.022	80.8	0.667	0.472	80.6	0.748	0.551	80.1	0.753	0.557
InfographicVQA	61.6	0.528	0.046	57.1	1.608	1.181	59.8	1.818	1.394	59.8	1.870	1.436
MMStar	60.5	0.167	0.018	60.2	0.507	0.344	59.8	0.556	0.390	60.5	0.560	0.395

Table 8. **Comparison of DBDPC, DBSCAN, DPC, and KMeans Clustering Methods for a Reduction Ratio of approximately 60% on LLaVA-OneVision-7B.**

Dataset	DBDPC (ours)			DBSCAN			DPC			KMeans		
	Metric	Proc. Time	Algo. Time	Metric	Proc. Time	Algo. Time	Metric	Proc. Time	Algo. Time	Metric	Proc. Time	Algo. Time
VideoMME	57.4	0.389	0.040	56.7	2.090	1.731	57.2	0.729	0.392	57.3	1.725	1.383
MME	1563.8	0.255	0.028	1531.3	1.577	1.304	1556.7	0.637	0.380	1549.9	1.254	0.999
DocVQA	84.7	0.530	0.044	83.5	4.127	3.607	83.8	0.950	0.442	79.6	2.059	1.544
MLVU	64.2	0.384	0.039	62.9	2.041	1.700	64.2	0.727	0.382	64.6	1.725	1.377
LLaVA-Interleave	62.1	0.151	0.016	63.9	0.697	0.547	62.3	0.258	0.121	62.3	0.711	0.566
ChartQA	76.0	0.366	0.031	74.6	1.851	1.507	74.9	0.758	0.415	74.2	1.399	1.059
MMBench	80.1	0.151	0.016	78.3	0.725	0.580	79.4	0.326	0.179	79.9	0.702	0.552
MuirBench	43.2	0.215	0.023	40.4	1.143	0.943	43.9	0.466	0.273	42.9	0.955	0.763
ScienceQA	94.7	0.147	0.015	91.5	0.665	0.518	94.2	0.251	0.117	93.4	0.661	0.518
MMMU	48.3	0.110	0.009	47.5	0.373	0.263	48.3	0.187	0.078	48.2	0.500	0.391
AI2D	80.7	0.202	0.022	80.2	1.165	0.971	81.0	0.455	0.264	81.1	1.062	0.860
InfographicVQA	61.6	0.528	0.046	56.6	2.584	2.147	59.2	0.975	0.547	57.8	1.780	1.357
MMStar	60.5	0.167	0.018	58.0	0.870	0.704	60.6	0.376	0.213	60.2	0.828	0.661

Table 9. **Comparison of EUTI-based visual tokens pruning and FastV for a Reduction Ratio of approximately 60% on LLaVA-OneVision-7B.**

Dataset	EUTI (Ours)			FastV		
	Metric	Proc. Time	Algo. Time	Metric	Proc. Time	Algo. Time
VideoMME	58.4	0.351	0.005	57.6	0.381	0.040
MME	1560.0	0.256	0.004	1570.7	0.283	0.025
DocVQA	86.5	0.521	0.005	85.3	0.559	0.032
MLVU	64.3	0.355	0.004	63.1	0.391	0.040
LLaVA-Interleave	58.9	0.140	0.003	59.7	0.152	0.007
ChartQA	78.6	0.344	0.004	78.0	0.363	0.016
MMBench	80.2	0.142	0.003	79.2	0.151	0.005
MuirBench	40.0	0.191	0.003	40.8	0.204	0.009
ScienceQA	93.6	0.137	0.003	92.3	0.149	0.006
MMMU	48.8	0.101	0.002	47.3	0.110	0.003
AI2D	81.1	0.191	0.003	80.3	0.202	0.009
InfographicVQA	63.0	0.425	0.005	60.3	0.473	0.040
MMStar	59.6	0.159	0.003	59.6	0.170	0.007

Table 10. **Comparison of PACT with Standalone Methods: EUTI-based Visual Token Pruning and DBDPC Clustering Algorithm for a Reduction Ratio of approximately 70%, applied on LLaVA-OneVision-7B.**

Dataset	PACT			DBDPC			EUTI		
	Metric	Proc. Time	Algo. Time	Metric	Proc. Time	Algo. Time	Metric	Proc. Time	Algo. Time
VideoMME	57.5	0.321	0.021	57.3	0.342	0.040	58.4	0.305	0.005
MME	1558.7	0.226	0.017	1543.7	0.243	0.028	1595.9	0.213	0.004
DocVQA	84.3	0.467	0.026	82.5	0.500	0.044	85.3	0.456	0.005
MLVU	64.6	0.322	0.022	63.9	0.358	0.039	64.4	0.291	0.004
LLaVA-Interleave	63.9	0.133	0.010	62.6	0.149	0.016	57.1	0.127	0.003
ChartQA	77.2	0.311	0.019	75.1	0.333	0.031	78.2	0.292	0.004
MMBench	80.2	0.134	0.010	79.7	0.147	0.016	79.6	0.128	0.003
MuirBench	42.8	0.175	0.013	43.2	0.211	0.023	39.9	0.164	0.003
ScienceQA	93.6	0.130	0.010	93.8	0.142	0.015	92.2	0.123	0.003
MMMU	48.9	0.103	0.007	47.2	0.109	0.009	48.9	0.096	0.002
AI2D	80.6	0.173	0.013	80.5	0.191	0.022	79.9	0.164	0.003
InfographicVQA	61.9	0.403	0.023	58.8	0.465	0.046	60.4	0.360	0.005
MMStar	59.5	0.147	0.011	59.5	0.163	0.018	59.2	0.140	0.003

Table 11. **Ablation Studies on DBDPC-based visual token reduction for a Reduction Ratio of approximately 60% on LLaVA-OneVision-7B.** We report only the metrics, as processing time is similar across different approaches. When ablating the Center Position-IDs assignment, we reorder the hidden states based on the mean of the Position-IDs of the elements in each cluster and then assign position IDs sequentially.

	DBDPC	w/o Center Position-IDs assignment	w/o Proportional Attention	w/o Merging
VideoMME	57.4	58.0	57.9	57.5
MME	1563.8	1539.3	1523.8	1476.9
DocVQA	84.7	28.2	84.2	83.1
MLVU	64.2	61.2	63.9	63.5
LLaVA-Interleave	62.1	69.6	63.2	63.6
ChartQA	76.0	24.8	76.0	74.4
MMBench	80.1	76.1	80.1	79.6
MuirBench	43.2	26.5	43.2	44.0
ScienceQA	94.7	67.4	94.2	93.6
MMMU	48.3	34.5	47.6	48.2
AI2D	80.7	43.0	80.4	79.9
InfographicVQA	61.6	17.8	59.8	58.7
MMStar	60.5	58.9	59.6	59.1

Table 12. **Ablation Studies on the EUTI-based Visual Token Pruning for a Reduction Ratio of approximately 70%, applied on LLaVA-OneVision-7B.** We report only the metrics, as processing time is similar across different approaches.

Dataset	EUTI	EUTI w/o Norm	Norm (EUTI w/o Global Query)
VideoMME	58.4	57.6	56.6
MME	1595.9	1573.4	1576.5
DocVQA	85.3	85.1	79.7
MLVU	64.3	63.0	63.1
LLaVA-Interleave	57.1	57.9	52.9
ChartQA	78.2	76.4	76.7
MMBench	79.6	79.4	79.4
MuirBench	40.0	40.5	39.6
ScienceQA	92.2	91.8	93.5
MMMU	48.9	49.3	49.2
AI2D	79.9	79.9	79.7
InfographicVQA	60.4	60.1	49.3
MMStar	59.2	57.4	59.2

Table 13. **Ablation Study on Pruned Tokens Recovery for a Reduction Ratio of approximately 70%.** We remove the token recovery step, which is equivalent to Setting α to Zero. We report only the metrics, as processing time is similar across both approaches.

Dataset	PACT	PACT w/o Pruned-Token Recovery
VideoMME	57.6	57.4
MME	1556.7	1576.3
DocVQA	84.3	84.3
MLVU	64.6	64.2
LLaVA-Interleave	63.9	59.6
ChartQA	76.4	76.4
MMBench	79.9	79.8
MuirBench	42.8	42.2
ScienceQA	93.3	93.6
MMMU	48.5	48.5
AI2D	80.6	80.6
InfographicVQA	61.9	61.3
MMStar	75.1	74.9

Table 14. **Ablation Study on Keys Utilization in DBDPC for a Reduction Ratio of approximately 60%.** Metrics are reported, as processing time is similar across both configurations.

Dataset	DBDPC	DBDPC w/o Keys
VideoMME	57.40	57.22
MME	1563.80	1526.18
DocVQA	84.70	80.50
MLVU	64.20	64.60
LLaVA-Interleave	62.10	60.80
ChartQA	76.00	68.80
MMBench	80.10	79.21
MuirBench	43.20	41.40
ScienceQA	94.70	91.90
MMMU	48.30	47.90
AI2D	80.70	79.10
InfographicVQA	61.6	56.70
MMStar	60.50	58.40