A More Word-like Image Tokenization for MLLMs

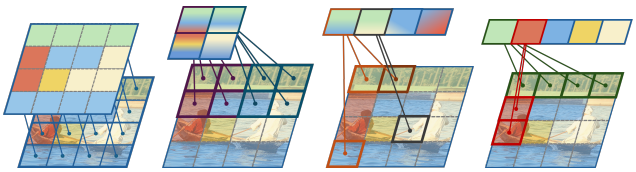
Hyun Lee[1], Hyemin Jeong[1], Yejin Kim[1], Hyungwook Choi[1]
Hyunsoo Cho[2*†], Soo Kyung Kim[2*†], Joonseok Lee[1*]
[1]Seoul National University, [2]Ewha Womans University
{hyun86,hyeminjeong,a2000yejin,chooi221,joonseok}@snu.ac.kr, {chohyunsoo,sookim}@ewha.ac.kr

Abstract

Modern multimodal large language models (MLLMs) typically keep the language model fixed and train a visual projector that maps the pixels into a sequence of tokens in its embedding space, so that images can be presented in essentially the same form as text. However, the language model has been optimized to operate on discrete, semantically meaningful tokens, while prevailing visual projectors transform an image into a long stream of continuous and highly correlated embeddings. This causes the visual tokens to behave differently from the word-like units that LLMs are originally trained to understand. We propose a novel Disentangled Visual Tokenization (DiVT) that clusters patch embeddings into coherent semantic units, so each token corresponds to a distinct visual concept instead of a rigid grid cell. DiVT further adapts its token budget to image complexity, providing an explicit accuracy-compute trade-off modifying neither the vision encoder nor the language model. Across diverse multimodal benchmarks, DiVT matches or surpasses baselines with significantly fewer visual tokens, demonstrating robustness under limited token budgets, significantly reducing memory cost and latency while making visual inputs more compatible with LLMs. Our code is available at https://github.com/snuviplab/DiVT.

1. Introduction

Large Language Models (LLMs) have shown remarkable capabilities in understanding and generating language through fine-grained textual representations. Extending it to the visual domain, Multimodal Large Language Models (MLLMs) aim to seamlessly integrate visual information in a similar form with the text tokens, enabling unified multimodal understanding. To avoid the high computational cost of training from scratch, it is common to adopt a pre-trained LLM for its reasoning ability and a pre-trained vision encoder (*e.g.* CLIP [35], SigLIP [49]) to map the pixel-level signals to a semantic latent space. Since these two pre-trained models operate on different latent spaces, a visual projector is trained to map from one to the other, usually from the visual space to the textual. Consequently, the projector should be able to organize visual information into a token sequence that mirrors the semantic and structural properties of text tokens to fully leverage the LLM's linguistic and reasoning capabilities.



(a) MLP (b) Grid-wise (c) Resampler (d) DiVT (Ours)

Figure 1. **Comparison with existing projectors.** Patch features (bottom layer) are mapped to visual tokens (top layer). Each color represents the principal semantic of the patch.

However, most existing visual projectors are implemented as a simple linear layer [8, 31] applied to fixed patch features from the vision encoder (Fig. 1a). This design adheres to the rigid patchification scheme of ViT [16], which evenly splits an image into a set of fixed-size image patches. While this approach provides a straightforward way to utilize all the visual information from the encoder, it inevitably introduces significant redundancy and incurs unnecessary computational cost. To mitigate this redundancy, recent work [6, 10, 14, 27, 40] aggregates adjacent patches to form a smaller set of visual tokens (Fig. 1b). Resampler-based approaches [3, 15, 26] generate a compact set of learnable queries that summarize information by globally attending to all visual features, enjoying further flexibility (Fig. 1c).

We see, however, the following three primary discrepancies between the widely-used visual tokens and text tokens in their formation. First, **the patch features are semantically entangled**, not just because they are constructed with a fixed grid without considering their content but also be-

*Corresponding authors
†Department of AI, Institute for Multiscale Matter and Systems

cause they have already undergone multiple layers of self-attention in the vision encoder. Thus, using these raw patch features without explicit semantic disentanglement inevitably propagates such mixed representations into the projector, leading to entangled tokens. Text tokens, in contrast, are generated by discrete tokenizers such as Byte Pair Encoding (BPE) [36], which segment text into fixed, independent units with limited inter-token interactions. Thus, their contextual connection usually emerges only after repeated self-attentions in LLMs.

Second, although the amount of information greatly varies across images, existing methods still produce **a fixed number of visual tokens** for every input. This often induces an unnecessarily redundant set of tokens for a large portion of patches depicting a single concept (Fig. 1a), or conversely, a loss of details when the image is forced into fewer tokens than it actually needs. This contrasts with how a linguistic expression used to describe a scene, where the length of an expression usually varies according to the complexity of the scene (Fig. 1d).

Third, existing approaches **lack control over the amount of information** encoded within each token. Whereas textual tokenization offers variable segmentation (*e.g*. sub-words vs. whole words) to balance expressiveness and sequence length, current visual tokenization relies on spatial operations with no principled way to control how finely or coarsely an image is partitioned in a manner compatible with LLM-based reasoning.

To address these limitations, we propose a clustering-based visual tokenization approach that aims to explicitly align with LLMs. Instead of mapping each patch independently or aggregating patches purely by *spatial* proximity, our method clusters patch features from the vision encoder into *semantically* coherent units, with each cluster forming a distinct visual token. This encourages disentanglement across tokens, so that each token corresponds more closely to a specific visual concept (*e.g*. an object, part, or salient region) rather than a mixed patch of unrelated content. Crucially, the tokenizer is trained using only the language modeling objective, without any external supervision, so that the resulting visual tokens are shaped by how the LLM internally organizes and exploits semantic information. Henceforth, we use the terms *projector* and *tokenizer* interchangeably throughout this paper.

Our design incorporates dynamic token allocation; that is, the number of tokens to represent an image is adaptively determined by its content. The token budget naturally scales with the image's semantic complexity, avoiding both excessive redundancy in simple scenes and insufficient capacity in cluttered ones. Also, the amount of information per token is controlled by a clustering threshold at training, providing an interpretable knob to adjust how finely or coarsely image regions are grouped to form a token. Interestingly,

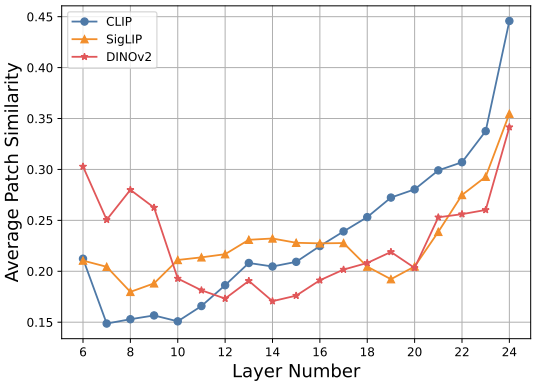


Figure 2. **Patch similarity across ViT layers**. Patch-wise cosine similarity increases in deeper layers, indicating that repeated self-attention homogenizes patch embeddings within an image.

this threshold can also be adjusted at inference time, allowing practitioners to trade-off representational detail against memory and latency without retraining, and to match computational budgets in deployment.

We evaluate our LLM-friendly visual tokenizer across a broad suite of multimodal benchmarks under varying token budgets. Empirically, it consistently matches or surpasses baselines with significantly fewer visual tokens, demonstrating robustness under limited token budgets, where the performance gain becomes more evident as the token budget gets more constrained. The ability to adapt token counts and granularity at inference time yields a practical, training-free mechanism to balance cost and fidelity. Notably, our approach is agnostic to the choice of vision encoder and remains effective when scaled to larger LLMs, underscoring its practicality and broad applicability for cost-efficient, human-aligned multimodal understanding.

2. Motivation for Redesigning Visual Tokens

We first question if the current visual representations fed into MLLMs behave like language tokens, or collapse to overly similar embeddings within each input.

2.1. Entanglement within Visual Embeddings

We hypothesize that repeated self-attention and image-level objectives make visual embeddings within an image highly similar, and that a linear projector in Fig. 1a largely preserves this redundancy in the language model space. To test this hypothesis, we conduct a toy-scale experiment. Specifically, we take a CLIP-style vision transformer (ViT) encoder following the standard patchification scheme. For each image, we produce patch embeddings at different transformer layers using this encoder, and measure cosine similarity between all pairs of patches from the same image, then take the average over pairs and images. We report the pairwise cosine similarity averaged over all token pairs

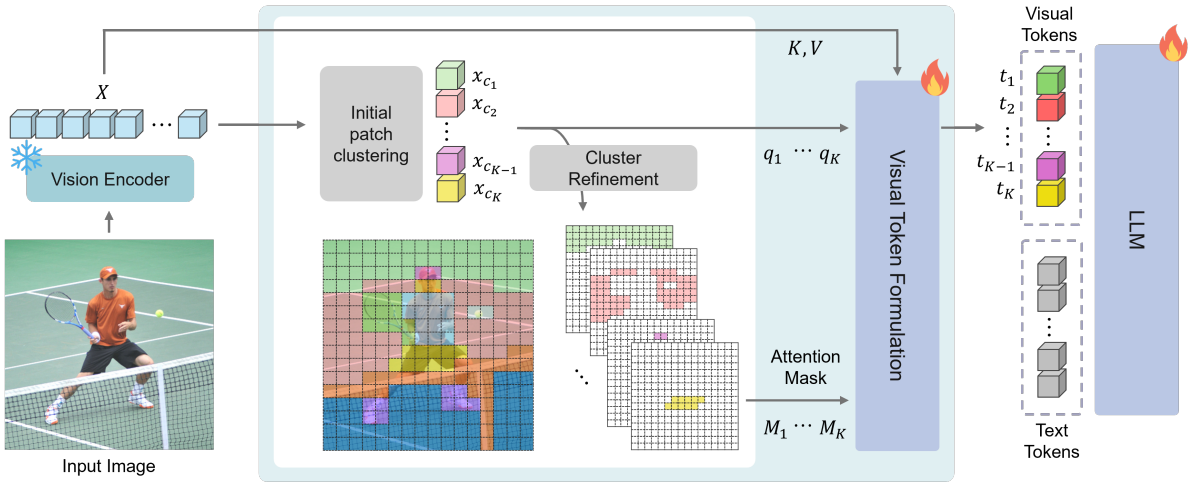


Figure 3. **Overview of DiVT**. The process consists of three main stages: (1) *Initial patch clustering*, which elects representative patch centroids based on feature diversity (Sec. 3.1); (2) *Cluster refinement* for semantically more coherent groups (Sec. 3.2); (3) *Visual token formulation* to aggregate information within each cluster to semantically disentangled visual tokens (Sec. 3.3).

of 500 images from MMBench [32].

Fig. 2 shows that intermediate layers maintain moderate variation among visual embeddings, whereas higher layers exhibit much skyrocketed similarity within each image. This suggests that visual embeddings become strongly entangled as global context is repeatedly mixed through self-attention, especially under image-level pre-training objectives such as contrastive learning or classification. As a result, the final visual tokens form a long sequence with many near-duplicate entries, which redundantly inflates the KV cache and spreads attention over redundant evidence.

2.2. Similarity between visual and language tokens

To quantitatively compare the visual and language sides of the MLLMs, we measure the token similarity within each modality. Language tokens show significantly lower mutual similarity of 0.0378 ± 0.0002 on average, consistent with their discrete and well-separated nature. Visual tokens from the MLP projector reveal significantly higher similarity of 0.3823 ± 0.0018, verifying the redundancy suggested above.

Overall, these results reveal a structural mismatch inside the current visual-language models. Vision encoders tend to produce entangled embeddings, and linear projectors preserve this redundancy when mapping them into the language model space. This motivates a visual tokenization scheme that explicitly targets semantic disentanglement and compact concept-level tokens in the next section.

3. Disentangled Visual Tokenization

Motivated by the analysis on the entanglement problem inherent in current visual tokens (Sec. 2), we introduce our Disentangled Visual Tokenization (DiVT) designed to produce semantically coherent and disentangled visual tokens,

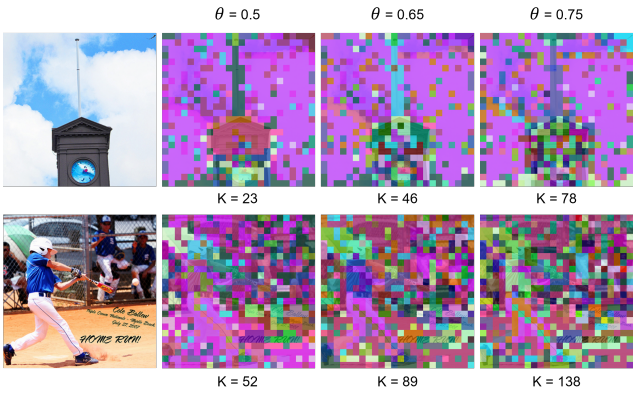


Figure 4. **Illustration of dynamic token clustering**. An image with relatively simpler content (top) uses less number of clusters than one with a more complex scene (bottom). See Appendix E for more examples.

supporting adaptive token lengths for each image. Unlike the linear projectors that ignore patch entanglement, our approach restructures visual information into *semantic* units, better compatible with the discrete text tokens in LLMs. We subsequently detail our proposed approach in Fig. 3: initial patch clustering (Sec. 3.1), cluster refinement (Sec. 3.2), visual token formulation (Sec. 3.3), and semantic granularity control (Sec. 3.4).

3.1. Initial Patch Clustering

In order to produce a set of visual tokens that are semantically disentangled and free from redundancy, we begin with clustering the patches based on their features. A natural option is to use a standard clustering algorithm such as k-means, but it requires prefixing the number of clusters k, thereby enforcing the same token budget for all images regardless of their content. This highlights a broader limi-

tation of existing tokenization strategies, including the linear projectors and spatial reduction methods such as strided pooling, pixel-shuffle, or grid-level subsampling. All of these methods produce a fixed number of visual tokens *a priori*, rather than letting the token count reflect the semantic complexity of the input image. Unlike text, where detailed or information-rich sentences naturally expand into a longer sequence, fixed-budget visual tokenization cannot allocate more tokens to scenes containing richer content.

To overcome this structural rigidity, we aim to dynamically decide the number of clusters based on the feature distribution itself. Our design is guided by the observation that patches whose feature vectors lie in dense regions of the embedding space, *i.e.*, have many neighbors above a cosine-similarity threshold, tend to correspond to semantically dominant structures, whereas patches in sparse regions typically encode fine-grained or isolated details. Instead of imposing a predetermined token budget, we thereby derive cluster centroids directly from the similarity structure of the patch embeddings.

Formally, we denote the set of patch features by $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{N}$, where N is the number of patches per image. With a ViT-like vision encoder, N is usually fixed regardless of the image content. Given $\mathcal{X}$, we compute the patch-wise cosine similarity matrix $\mathbf{S} \in \mathbb{R}^{N \times N}$. We define the patches whose pairwise similarity exceeds some threshold θ as *neighbors*. That is, taking $\mathbf{S}$ only with its elements above θ as the adjacency matrix would create a neighborhood graph among the patches. In this graph, we greedily select a node (patch) with the largest degree (the number of edges connected to it). The chosen node $\mathbf{x}$ becomes the centroid of the first cluster c_1, denoted by $\mathbf{x}_{c_1}$, and all neighboring nodes to it construct the cluster c_1. We remove all nodes belonging to c_1, and repeat this process to select the subsequent clusters c_k for $k = 2, 3, ..., K$, until no node remains. In this way, the remaining candidates are at least θ apart from any previously chosen centroids. This guarantees that subsequently chosen centroids are semantically distinct from previously selected ones, rather than redundant variations of the same structure. Algorithm 1 summarizes these steps. While the algorithm induces an implicit ordering of centroids, we simply arrange them based on their spatial coordinates to form a sequence for LLM input.

This clustering approach satisfies two desirable properties. First, the number of clusters is adaptively decided based on the image content. Images with richer visual content naturally yield a larger number of clusters, while simpler or homogeneous images yield fewer, as illustrated in Fig. 4. The resulting token budget therefore adapts to the intrinsic complexity of the scene. Second, fine-grained details are not discarded. Patches that rarely resemble others form isolated clusters, ensuring that outliers, subtle edges, and small distinctive objects are preserved rather than absorbed into another broader cluster. Through this mechanism, the visual token count is determined adaptively by the feature distribution, providing a natural basis for controllable token granularity discussed in Sec. 3.4.

Algorithm 1 Adaptive Centroid Selection

Input: Patch features $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{N}$, similarity threshold θ, $\mathbb{I}(\cdot)$ is an indicator function.
Output: Centroid indices $\mathcal{C}$

$\mathbf{S}_{i,j} = \cos\_\text{sim}(\mathbf{x}_i, \mathbf{x}_j)$
$n_i = \sum_j \mathbb{I}(\mathbf{S}_{i,j} > \theta)$
candidates = argsort(n, desc)
$\mathcal{C} = [\,]$
while candidates $\neq \emptyset$ **do**
  $c =$ candidates[0]
  $\mathcal{C}.append(c)$
  candidates = candidates $\setminus \{j \mid \mathbb{I}(\mathbf{S}_{c,j} > \theta)\}$
end while
return $\mathcal{C}$

3.2. Cluster Refinement for Disentanglement

Although we have generated the initial clusters in Sec. 3.1, the cluster allocation can be far from optimum due to its greedy nature. To be specific, an image patch may belong to more than one cluster (that is, it is close enough to more than one centroid patch). According to Algorithm 1, its initial cluster would be the one having more neighbors, but it would be ideal to associate it with the closest centroid, which has been discovered in the later steps. For this reason, we refine the cluster assignment for better semantic disentanglement across the clusters.

Given patch features $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{N}$ and their corresponding centroid indices $\{c_k\}_{k=1}^{K}$ constructed in Sec. 3.1, each patch is assigned to the centroid with the highest similarity:

$$\mathcal{C}_k = \{\, \mathbf{x}_i \mid k = \underset{j}{\text{argmax}} \cos(\mathbf{x}_i, \mathbf{x}_{c_j}) \,\}, \tag{1}$$

where $\mathbf{x}_{c_j}$ indicates the centroid of the cluster j. Each resulting $\mathcal{C}_k$ for $k = 1, ..., K$ represents a set of patches that share coherent semantics in the feature space. In this way, the initial clustering step in Sec. 3.1 serves only for selecting the set of centroids (roughly speaking, this can be seen as a discrete density estimation), while this refinement step achieves the desired disentanglement.

3.3. Visual Token Formulation

Once the clusters are finalized in Sec. 3.2, we aggregate information from the patches in each cluster to produce a single visual token per cluster. Specifically, we adopt cross-attention using the centroid as the query:

$$\mathbf{Q}_k = \mathbf{W}^Q \mathbf{x}_{c_k},\ \mathbf{K}_i = \mathbf{W}^K \mathbf{x}_i,\ \mathbf{V}_i = \mathbf{W}^V (\mathbf{x}_i + \mathbf{P}_i), \tag{2}$$

where $\mathbf{W}^{\{Q,K,V\}}$ are learnable parameters, and $\mathbf{P}_i$ is a learnable positional embedding that provides spatial context. We inject $\mathbf{P}_i$ only into the value branch because it directly influences the aggregated token content, whereas adding it to $\mathbf{Q}$ or $\mathbf{K}$ would merely perturb attention scores without providing meaningful structural cues.

To ensure each token to attend inside its cluster only, we apply a cluster-restricted attention mask:

$$\mathbf{M}_{k,i} = \begin{cases} 0, & \text{if } i \in \mathcal{C}_k, \\ -\infty, & \text{otherwise.} \end{cases} \tag{3}$$

The resulting visual token $\mathbf{t}_k$ for the cluster k is obtained by

$$\mathbf{t}_k = \mathrm{MLP}\left(\sum_i \mathrm{softmax}(\mathbf{Q}_k\mathbf{K}_i^\top + \mathbf{M}_{k,i})\mathbf{V}_i\right). \tag{4}$$

This design has two key advantages. First, because the aggregation is restricted to the patches that share relevance with the centroid, the resulting token naturally encapsulates a well-separated semantic unit. Second, every patch belongs to one cluster, ensuring that no visual content is discarded. Even fine-grained details (*e.g.*, small objects, object parts, or text characters) contribute to some token, preventing information loss. Putting them together, these properties enable the DiVT to compress images into a compact token set while preserving patch-level information.

3.4. Controlling Semantic Granularity

The similarity threshold θ serves as a principled means to control the semantic granularity of the resulting visual tokens. Since θ defines which patches are considered neighbors, it implicitly determines the scale at which visual structures are grouped. A higher θ enforces stricter grouping criteria, producing a larger number of finer-grained tokens that capture subtle variations across the image. It tends to preserve a more detailed structure and form smaller semantic units, at the cost of higher token counts and computational cost. Conversely, a lower θ encourages broader grouping, generating fewer clusters and more coarse-grained tokens. Such representations remain compact, but may lose some localized details, merging visually distinct patches even when their similarity is not extremely high.

This behavior mirrors the continuum of textual tokenization level, from character-level to subword- and word-level, to balance expressiveness, semantic specificity, and sequence length. Finer granularity supports detail-oriented representations but leads to longer sequences, while coarser granularity improves efficiency at the risk of under-representing subtle information. By adjusting θ, practitioners can flexibly navigate this trade-off to match the needs for the downstream task or computational budget. Notably, θ can also be adjusted *training-free* at inference time, allowing the model to reduce the number of tokens and consequently lower the inference cost without retraining (see Sec. 4.3 for details).

Method	# Tokens	MMB	VQA^{v2}	GQA	MME	MM-Vet	VQA^{Text}	SQA^{IMG}	POPE
MLP	576	64.3	78.5	62.0	1510.7	31.1	58.2	66.8	85.6
ToME*	128	53.3	63.0	52.4	1088.4	27.2	49.1	59.6	62.8
FastV*	128	56.1	61.8	49.6	1208.9	28.1	50.6	60.2	59.6
PruMerge+*	128	61.3	74.7	57.8	1420.5	28.7	54.3	67.6	81.5
VisionZip*	128	62.0	75.6	57.6	1432.4	32.6	56.8	68.9	83.2
VisPruner*	128	62.7	75.8	58.2	1461.4	**33.7**	57.0	69.1	84.6
ATP-LLaVA*	144†	66.0	76.4	59.5	**1473.9**	-	-	69.1	84.2
TokenPacker	144	65.1	77.9	61.9	-	33.0	57.2	-	**87.0**
DiVT $_{\theta=0.75}$	136.5†	**66.7**	**78.2**	**62.0**	1457.6	30.2	**57.7**	**70.0**	86.2
ToME*	64	43.7	57.1	48.6	922.3	24.1	45.3	50.0	52.5
FastV*	64	48.0	55.0	46.1	1019.6	25.8	47.8	51.1	48.0
PruMerge+*	64	59.3	67.4	54.9	1198.2	25.9	53.0	68.6	77.4
VisionZip*	64	60.1	72.4	55.1	1365.6	31.7	55.5	69.0	77.0
VisPruner*	64	61.3	72.7	55.4	1369.9	**32.3**	55.8	69.1	80.4
ATP-LLaVA*	88†	64.7	73.3	56.8	1401.5	-	-	67.2	82.6
TokenPacker	64	64.1	77.2	61.1	-	31.7	-	-	**86.3**
DiVT $_{\theta=0.65}$	74.1†	**65.5**	**77.7**	61.4	**1465.7**	32.1	**57.2**	68.1	85.8
DiVT $_{\theta=0.62}$	63.7†	64.3	**77.7**	**61.6**	1463.0	30.6	57.0	**70.6**	86.2
ToME*	32	31.6	46.8	43.6	828.4	17.3	38.3	41.4	39.0
FastV*	32	37.8	43.4	41.5	884.6	20.7	42.5	42.6	32.5
PruMerge+*	32	56.8	54.9	51.1	940.8	21.4	50.6	68.5	70.9
VisionZip*	32	57.7	67.1	51.8	1247.4	25.5	53.1	68.8	68.7
VisPruner*	32	58.4	67.7	52.2	1271.0	28.8	53.9	**69.2**	72.7
TokenPacker	36	62.8	75.0	59.6	-	29.6	-	-	**86.2**
DiVT $_{\theta=0.5}$	35.7†	**65.0**	**77.0**	**60.6**	**1458.2**	**31.7**	**57.1**	68.2	85.8
DiVT $_{\theta=0.4}$	22.4†	64.7	76.4	60.1	1450.9	31.7	56.1	69.1	84.8
DiVT $_{\theta=0.3}$	13.5†	64.2	75.3	59.2	1462.8	28.0	55.4	69.4	84.3

Table 1. **Comparison with token compression methods on LLaVA-1.5 7B** [31]**.** Training-free methods are marked with *. †denotes that the number of tokens is averaged across the test set, as each sample uses varied number of tokens. Token counts for ATP-LLaVA are brought from its original paper [47]. **Bold** and underline indicate the best and second-best results, respectively.

4. Experiments

We conduct extensive experiments to verify the effectiveness of our method on a diverse set of benchmarks designed to test various capabilities.

4.1. Experimental Setup

Benchmarks. We evaluate on eight widely-used benchmarks: general conversation and VQA (MMBench [32], VQAv2 [19], GQA [23], MME [18], MM-Vet [48]), OCR-related tasks (TextVQA [38]), knowledge-based tasks (SQA-IMG [33]), and specialized tasks measuring object hallucination (POPE [28]). A higher score indicates better performance across all benchmarks.

Baselines. We compare our DiVT against two categories of baselines: training-free token reduction methods, including ToME [5], FastV [9], PruMerge [37], VisionZip [45], VisPruner [50], and ATP-LLaVA [47], and TokenPacker [27] which requires training a modified projector.

To specifically isolate the contribution of our projector itself, we further compare DiVT against other prominent visual projectors (*e.g.*, Resampler [3], C-Abstractor [6]), orig-

inally proposed as a component of various MLLMs.

Implementation Details. We replace the MLP projector in LLaVA-1.5 with our proposed projector, leaving all other experimental settings including the model backbone and training datasets unchanged. We cross-validate $\theta \in \{0.5, 0.65, 0.75\}$, which correspond to approximately 35.7, 74.1, and 136.5 tokens, respectively. Additionally, we include $\theta = 0.62$ (63.7 tokens) for a fairer comparison with baselines operating at a 64-token budget, and $\theta = 0.3$ and 0.4 (13.5 and 22.4 tokens, respectively) to examine performance under a more aggressively limited budget regime. See Appendix A for more details.

4.2. Main Results

Comparison with Baselines. Tab. 1 compares the performance of competing methods using a fixed LLaVA-1.5 7B backbone. For methods generating a variable number of tokens specific to each image, ATP-LLaVA and DiVT, we report the average token count measured at evaluation.

First of all, our method achieves highly competitive performance compared to the full LLaVA (576 tokens) despite the drastically reduced number of tokens (136.5 to 22.4, depending on θ). For instance, with 136.5 tokens, our method achieves 66.7 on MMBench and 78.2 on VQAv2, achieving comparable or even superior performance, compared to the full model's 64.3 and 78.5, respectively.

While most other methods are training-free (marked with *), TokenPacker is the only other approach that trains its projector. This direct comparison reveals that DiVT demonstrates superior performance, especially on general conversation and VQA tasks (*e.g.*, MMB, VQAv2 or GQA). This suggests that our approach of organizing tokens based on semantic content is more effective than methods relying on static grid-based compression.

Notably, the performance gain with our projector gets more pronounced with a tighter token budget. DiVT $_{\theta=0.5}$, using 35.7 tokens, achieves 65.0 on MMB and 77.0 on VQAv2, substantially outperforming TokenPacker's 36-token model, which scored 62.8 and 75.0, respectively. This advantage consistently holds across GQA (60.6 *vs.* 59.6) and MM-Vet (31.7 *vs.* 29.6). Furthermore, our most compact models (DiVT $_{\theta=0.3}$, DiVT $_{\theta=0.4}$) emphatically prove this; despite using only 13.5 to 22.4 tokens, they still outperform other baselines on majority of benchmarks. Grid-based methods such as TokenPacker or token reduction approaches focus on *compressing* tokens and tend to lose core information under limited token budgets, while our DiVT controls the *semantic granularity* of each token, resulting in robust performance with more informative visual tokens.

Comparison with Visual Projectors. Tab. 2 compares the performance of various visual projectors, following the experimental setup in TokenPacker [27] with LLaVA-1.5 7B. Our method consistently achieves the highest scores on MMBench, VQAv2, and GQA, while remaining competitive on MM-Vet and POPE. This demonstrates that DiVT effectively distills the most important visual information into a compact token set, preserving model performance even with a reduced token budget. Overall, these results highlight the efficiency and effectiveness of our visual projector compared to existing approaches.

Projector	# Tokens	MMB	VQA^{v2}	GQA	MM-Vet	POPE
MLP (full)	576	64.3	78.5	62.0	31.1	85.9
Average-Pooling	64	62.4	72.6	58.8	27.1	85.4
Resampler [3]	64	63.4	74.1	57.7	29.2	83.4
C-Abstractor [6]	64	62.5	74.4	59.3	29.0	85.0
Pixel-Shuffle [10]	64	63.2	74.6	59.1	28.5	85.2
LDP-v2 [14]	64	63.7	75.3	59.7	30.0	85.5
TokenPacker [27]	64	<u>64.1</u>	<u>77.2</u>	<u>61.1</u>	**31.7**	**86.3**
DiVT (Ours)	63.7	**64.3**	**77.7**	**61.6**	<u>30.6</u>	<u>86.2</u>

Table 2. **Comparison of different visual projectors.**

Qualitative Analysis. Fig. 5 illustrates the attention map produced by the MLP projector (top) and DiVT(bottom) for a prompt "Describe this image". Each set of attention maps highlights some regions in the image that the model attends to for a specific word token within its generated caption.

Our method consistently demonstrates more focused and semantically coherent attention, clearly concentrating on the corresponding object. This sharply contrasts with the MLP projector, which exhibits a more dispersed and less interpretable attention pattern. These results underscore that our DiVT effectively disentangles visual information into meaningful semantic units, leading to improved interpretability and object grounding.

4.3. Further Analysis

Encoder-Agnostic Versatility. A key strength of DiVT is its design as an encoder-agnostic component. We validate this by applying DiVT to diverse vision encoders, including CLIP [35], SigLIP [49], and DINOv2 [34]. We set the threshold to $\theta = 0.6$ for SigLIP and $\theta = 0.66$ for DINOv2, so their average token counts remain comparable to CLIP.

According to the results presented in Tab. 3, our method consistently proves its general applicability, sufficiently preserving the encoders' inherent capabilities with a significantly reduced number of tokens. Notably, despite using about 13% of the tokens of the full MLP projector, our method achieves highly comparable performance across all encoders. This demonstrates that the significant gains in computational efficiency with aggressive token reduction far outweigh the marginal, task-specific performance trade-offs. This strongly demonstrates that our DiVT does not overfit to a specific feature space but generalizes effectively, confirming its robustness as a plug-and-play module.

Scalability to Larger LLMs. We further examine the scalability of DiVT by integrating it with a larger LLaVA-1.5 13B model (based on Vicuna-13B [13]). The results in

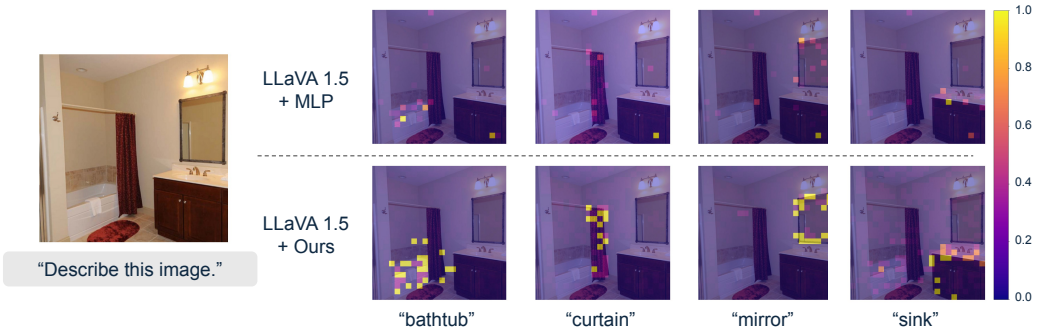


Figure 5. **Qualitative demonstration.** Attention maps highlight the regions in the image that the model attends to for specific object tokens. Our method produces attention clusters tightly focused on the object token, yielding more interpretable patterns, while the MLP projector exhibits more scattered attention over irrelevant regions. See Appendix D for more examples.

Backbone	# Tokens	MMB	VQA^{v2}	GQA	MME	MM-Vet	VQA^{Text}	SQA^{IMG}	POPE
CLIP	576	64.3	78.5	62.0	1510.7	31.1	58.2	66.8	85.6
+ DiVT	74.1	65.5	77.7	61.4	1465.7	32.1	57.2	68.1	85.8
SigLIP	576	66.6	80.0	62.9	1498.3	33.7	60.3	69.5	85.3
+ DiVT	74.3	66.4	78.5	62.1	1506.6	31.5	58.1	68.1	84.5
DINOv2	576	59.9	75.5	62.1	1266.1	25.0	46.6	66.8	85.6
+ DiVT	70.9	56.9	74.2	61.0	1362.8	23.9	46.6	65.3	85.0
Vicuna-13B	576	67.7	80.0	63.3	1531.3	36.1	61.3	71.6	85.9
+ DiVT	74.1	68.4	78.8	62.4	1491.9	34.3	59.3	70.7	86.2

Table 3. **Performance comparison with various backbones.**

Tab. 3 imply that the benefit of the semantic disentanglement in visual tokens is robustly maintained and scales effectively with a more capable LLM.

Training-Free Token Adjustment. Recall from Sec. 3.4 that our DiVT allows training-free control of token numbers by adjusting the similarity threshold. When a lower θ is used at inference, the number of resulting tokens would be reduced, increasing the amount of information for each token.

Tab. 4, reporting the performance when a different θ is used from the one used at training (0.65 and 0.75, respectively), indicates that our DiVT trained with a larger θ can be used with a coarser granularity with minimal performance loss. Although adjusting the threshold slightly affects the consistency of the semantic granularity of each visual tokens, it provides a flexibility to balance between efficiency and performance. In contrast, increasing the threshold does not further improve performance. This is expected, since the tokens have no way to have more fine-grained information than it has been trained on.

We further observe that the model trained with a randomly sampled θ exhibits consistently stable performance across various inference thresholds, demonstrating robustness to varied token granularities. However, models trained with a fixed θ tend to achieve the best performance when evaluated at their corresponding training threshold, suggesting that specialized training for a specific granularity remains optimal.

θ	# Tokens	MMB	VQA^{v2}	GQA	MME	MM-Vet	VQA^{Text}	SQA^{IMG}	POPE
0.5	35.7	63.8	76.8	60.3	1462.7	30.6	56.7	68.2	82.7
0.65	74.1	65.5	77.7	61.4	1465.7	32.1	57.2	68.1	85.8
0.75	136.5	65.1	77.8	61.0	1465.8	30.3	57.5	69.1	83.9
0.5	35.7	65.3	76.2	60.1	1429.8	31.9	56.1	69.6	83.3
0.65	74.1	65.8	77.7	61.7	1479.0	32.1	57.0	69.3	85.4
0.75	136.5	66.7	78.2	62.0	1457.6	30.2	57.7	70.0	86.2
0.5	35.7	65.5	76.9	60.6	1456.1	32.2	56.5	69.9	85.5
0.65	74.1	65.6	77.6	61.1	1482.6	32.4	57.0	70.0	86.2
0.75	136.5	65.9	77.8	62.4	1494.5	31.9	57.0	69.9	86.4

Table 4. **Performance of DiVT with varied similarity threshold at inference**. The original models are trained with $\theta = 0.65$, $\theta = 0.75$, and a randomly sampled $\theta \in \{0.5, 0.65, 0.75\}$, respectively.

Dataset	MMB	VQA^{v2}	GQA	MME	MM-Vet	VQA^{Text}	SQA^{IMG}	POPE	Avg.
# Tokens	58.3	73.5	81.3	69.3	65.2	83.4	48.3	80.1	74.1

Table 5. **Adaptively decided token counts at $\theta = 0.65$.**

Adaptively-decided Token Counts. Since our tokenizer adaptively determines the number of tokens based on the visual complexity of each image, the resulting token count naturally varies across benchmarks. Tab. 5 reports the average number of tokens for each benchmark at $\theta = 0.65$. Benchmarks containing visually simple images (*e.g.* SQA-IMG) tend to require fewer tokens, whereas those involving more complex or text-heavy scenes (*e.g.* POPE, TextVQA) result in a significantly higher number of tokens. This variation reflects the intended behavior of our method, which assigns more tokens only when the visual content genuinely calls for them. See Appendix F for more details.

4.4. Discussion

We further discuss a few other approaches that share common goal with ours. Training-free token pruning [2, 7, 9, 21, 30, 37, 43–45, 47, 50] reduces the number of visual tokens after feature extraction or at an intermediate LLM layer, typically by discarding or merging tokens based on attention scores or feature similarity. These approaches offer similar advantage of being plug-and-play like ours, reducing inference cost without retraining. Operating *only* at in-

ference time, however, they do not reduce training cost and often introduce a mismatch between the model's training dynamics and its pruned inference behavior, which can lead to substantial performance degradation under aggressive token reduction. In contrast, our method generates a compact token set *upfront*, guided by LLM supervision. This allows the semantic structure to be learned end-to-end, maintaining full compatibility with the pre-trained LLM.

Chat-Univi [24] and SeTok [42] share a common goal with our method in generating semantically meaningful visual tokens, but differ in focus and design. In particular, Chat-Univi generates a fixed number of tokens to represent images, where each patch contributes to multiple tokens, which contrasts with our objective of semantic disentanglement and adaptive token allocation. SeTok treats tokenization as a part of the vision encoder, requiring *large-scale supervised training* with object-level annotations and multiple transformer layers for token refinement. In contrast, our approach serves as a lightweight projector trained end-to-end with the LLM, requiring only a single forward pass and no additional supervision. While SeTok specializes in object-level representations, our method flexibly controls semantic granularity, enabling a broader token spectrum.

5. Related Work

Multimodal Large Language Models. MLLMs extend text-only LLMs by incorporating vision encoders and lightweight adapters, enabling image understanding, visual reasoning, and multi-image or video based tasks [26, 31]. Early models typically pair LLMs with pre-trained vision encoders through simple projection [8, 31], while recent systems introduce stronger vision backbones, improved alignment objectives, and richer multimodal instruction tuning [11, 29, 41]. In parallel, compact MLLMs leverage smaller LLMs (*e.g.*, Phi [1], Gemma [39], Minicpm [22]) with improved adapters and data-efficient training [4, 14, 46, 52]. However, across these models, images are still converted to a long sequence of fixed patch-level tokens, leading to semantically entangled and redundant representations that inflate KV-cache size and latency without commensurate accuracy gains. We resolve this long-standing bottleneck by selecting semantically aligned tokens with an adaptively sized budget tied to image complexity.

Efficient MLLMs. On the language side, simplified fusion [20] and inference accelerations such as layer skipping or speculative/adaptive decoding [17, 25, 51] have been applied to reduce parameters or computational overhead. On the other hand, projector-centric methods compress vision features prior to the LLM via query-based summarization [15, 26], convolution layers that reduce spatial resolution [6, 14], and grid-wise aggregation that downsamples tokens via spatial grouping or patch aggregation [10, 27]. While effective for bandwidth, these techniques largely view the projector as a feature compressor that preserves proximity rather than semantics; the resulting tokens often remain semantically entangled. In contrast, we form semantically coherent visual words, and align them with the LLM's discrete interface, boosting reasoning at equal or lower token budgets without encoder-specific heads.

Another major paradigm focuses on eliminating redundant tokens at inference. Intra-LLM pruning [7, 9, 21, 30, 43, 47, 50] discards tokens at intermediate layers based on criteria such as layer index, attention-score, or with lightweight pruning modules. However, these approaches operate post-hoc on already entangled patch tokens and can interact unpredictably with kernel-level optimizations like KV-cache policies or FlashAttention. To mitigate this, a pre-LLM selection strategy [2, 37, 44, 45] is adopted to filter visual tokens before entering the language model, commonly by measuring their similarity to a global CLS token or text embedding, or by promoting token diversity to preserve complementary visual cues. These methods improve efficiency by reducing token budgets prior to multimodal fusion, but they primarily perform compression rather than semantic alignment. Instead, we address the root cause by constructing semantically aligned tokens upfront.

Most aligned with our work, tokenizers grouping patches into variable-length semantic units [12, 24, 42] have been proposed recently. While effective for grounding, they often require auxiliary supervision, reconstruction losses, or multi-stage refinement, increasing training cost and reducing generality. Moreover, token granularity is frequently controlled only indirectly, and adjusting budgets can require retraining or some heuristic. Our method targets the same goal but with a lighter, more general mechanism: feature-space clustering trained solely by the LLM's objective. A single similarity threshold controls granularity and an adaptive token budget, tunable at inference without retraining.

6. Summary

Motivated by mismatched properties between the visual and text tokens, we introduce a novel visual projector in MLLMs to address visual-text modality gap. Our method produces compact visual tokens in which each token corresponds to a semantically coherent concept, effectively mitigating the entanglement, redundancy, and loss of details inherent in conventional projectors. By dynamically allocating token counts based on image complexity and controlling semantic granularity through a single similarity threshold, DiVT adapts naturally to diverse visual inputs while supporting training-free adjustment at inference. Across benchmarks, this semantically aligned representation achieves competitive or superior performance with only a small fraction of the original tokens, substantially reducing memory and latency.

Acknowledgments

This work was also supported by the SOFT Foundry Institute at SNU, Samsung Electronics, Youlchon Foundation, National Research Foundation of Korea (NRF) grants (RS-2021-NR05515, RS-2024-00336576, RS-2023-0022663, RS-2025-25399604, RS-2024-00342044, RS-2025-16063688, RS-2025-02215813), and the Institute for Information & Communication Technology Planning & Evaluation (IITP) grants (RS-2022-II220264, RS-2024-00353131) funded by the Korean government.

References

[1] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv:2404.14219*, 2024. 8

[2] Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. DivPrune: Diversity-based visual token pruning for large multimodal models. In *CVPR*, 2025. 7, 8

[3] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv:2309.16609*, 2023. 1, 5, 6

[4] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. PaliGemma: A versatile 3B VLM for transfer. *arXiv:2407.07726*, 2024. 8

[5] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *ICLR*, 2023. 5

[6] Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. Honeybee: Locality-enhanced projector for multimodal LLM. In *CVPR*, 2024. 1, 5, 6, 8

[7] Junjie Chen, Xuyang Liu, Zichen Wen, Yiyu Wang, Siteng Huang, and Honggang Chen. Variation-aware vision token dropping for faster large vision-language models. *arXiv:2509.01552*, 2025. 7, 8

[8] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal LLM's referential dialogue magic. *arXiv:2306.15195*, 2023. 1, 8

[9] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *ECCV*, 2024. 5, 7, 8

[10] 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. *Science China Information Sciences*, 67(12):220101, 2024. 1, 6, 8

[11] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *CVPR*, 2024. 8

[12] Donghwan Chi, Hyomin Kim, Yoonjin Oh, Yongjin Kim, Donghoon Lee, Daejin Jo, Jongmin Kim, Junyeob Baek, Sungjin Ahn, and Sungwoong Kim. Slot-MLLM: Object-centric visual tokenization for multimodal LLM. *arXiv:2505.17726*, 2025. 8

[13] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, 2023. 6

[14] Xiangxiang Chu, Limeng Qiao, Xinyu Zhang, Shuang Xu, Fei Wei, Yang Yang, Xiaofei Sun, Yiming Hu, Xinyang Lin, Bo Zhang, et al. MobileVLM v2: Faster and stronger baseline for vision language model. *arXiv:2402.03766*, 2024. 1, 6, 8

[15] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *NeurIPS*, 2023. 1, 8

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1

[17] Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, et al. Layerskip: Enabling early exit inference and self-speculative decoding. In *ACL*, 2024. 8

[18] 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. 5

[19] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017. 5

[20] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *Conference on language modeling*, 2024. 8

[21] Lianyu Hu, Fanhua Shang, Liang Wan, and Wei Feng. iLLaVA: An image is worth fewer than 1/3 input tokens in large multimodal models. *arXiv:2412.06263*, 2024. 7, 8

[22] Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024. 8

[23] Drew A Hudson and Christopher D Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, 2019. 5

[24] Peng Jin, Ryuichi Takanobu, Wancai Zhang, Xiaochun Cao, and Li Yuan. Chat-univi: Unified visual representation empowers large language models with image and video understanding. In *CVPR*, 2024. 8

[25] Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *ICML*, 2023. 8

[26] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023. 1, 8

[27] Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jie Qin, Jianke Zhu, and Lei Zhang. Tokenpacker: Efficient visual projector for multimodal LLM. *International Journal of Computer Vision*, pages 1–19, 2025. 1, 5, 6, 8

[28] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *EMNLP*, 2023. 5

[29] Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. VILA: On pre-training for visual language models. In *CVPR*, 2024. 8

[30] Zhihang Lin, Mingbao Lin, Luxi Lin, and Rongrong Ji. Boosting multimodal large language models with visual tokens withdrawal for rapid inference. In *AAAI*, 2025. 7, 8

[31] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 1, 5, 8

[32] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. MMBench: Is your multi-modal model an all-around player? In *ECCV*, 2024. 3, 5

[33] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *NeurIPS*, 2022. 5

[34] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *arXiv:2304.07193*, 2023. 6

[35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 6

[36] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *ACL*, 2016. 2

[37] Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. LLaVA-PruMerge: Adaptive token reduction for efficient large multimodal models. In *ICCV*, 2025. 5, 7, 8

[38] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *CVPR*, 2019. 5

[39] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv:2403.08295*, 2024. 8

[40] Bo Tong, Bokai Lai, Yiyi Zhou, Gen Luo, Yunhang Shen, Ke Li, Xiaoshuai Sun, and Rongrong Ji. FlashSloth: Lightning multimodal large language models via embedded visual compression. In *CVPR*, 2025. 1

[41] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv:2409.12191*, 2024. 8

[42] Shengqiong Wu, Hao Fei, Xiangtai Li, Jiayi Ji, Hanwang Zhang, Tat-Seng Chua, and Shuicheng Yan. Towards semantic equivalence of tokenization in multimodal llm. *arXiv:2406.05127*, 2024. 8

[43] Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. Conical visual concentration for efficient large vision-language models. In *CVPR*, 2025. 7, 8

[44] Bingxin Xu, Yuzhang Shang, Yunhao Ge, Qian Lou, and Yan Yan. freePruner: A training-free approach for large multimodal model acceleration. *arXiv:2411.15446*, 2024. 8

[45] Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. VisionZip: Longer is better but not necessary in vision language models. In *CVPR*, 2025. 5, 7, 8

[46] Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. MiniCPM-V: A GPT-4V level mllm on your phone. *arXiv:2408.01800*, 2024. 8

[47] Xubing Ye, Yukang Gan, Yixiao Ge, Xiao-Ping Zhang, and Yansong Tang. ATP-LLaVA: Adaptive token pruning for large vision language models. In *CVPR*, 2025. 5, 7, 8

[48] 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. In *ICML*, 2024. 5

[49] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023. 1, 6

[50] Qizhe Zhang, Aosong Cheng, Ming Lu, Renrui Zhang, Zhiyong Zhuo, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. Beyond text-visual attention: Exploiting visual cues for effective token pruning in vlms. In *ICCV*, 2025. 5, 7, 8

[51] Wenhong Zhu, Hongkun Hao, Zhiwei He, Yiming Ai, and Rui Wang. Improving open-ended text generation via adaptive decoding. In *ICML*, 2024. 8

[52] Yichen Zhu, Minjie Zhu, Ning Liu, Zhiyuan Xu, and Yaxin Peng. LLaVA-Phi: Efficient multi-modal assistant with small language model. In *International Workshop on Efficient Multimedia Computing under Limited*, 2024. 8

A More Word-like Image Tokenization for MLLMs

Supplementary Material

A. Implementation Details

Hyperparameters. Our implementation closely follows the standard LLaVA-1.5 training pipeline, with only minimal adjustments required to integrate DiVT into the multimodal architecture. Pretraining is performed for one epoch following the LLaVA-1.5 recipe; we use batch size of 256, initial learning rate of 10^{-3} with cosine decay and 3% warm-up, AdamW without weight decay, and DeepSpeed Stage 2. We update only the parameters of our projector. Finetuning adopts the same training configuration, differing only in the decreased batch size of 128, a reduced learning rate of 2×10^{-5}, and DeepSpeed Stage 3, through which the projector and LLM are jointly optimized. Unless stated otherwise, we adopt $\theta = 0.65$ as the default threshold for our DiVT.

More on Model Architecture. We use two ViT-based vision encoders, `facebook/dinov2-large` and `google/siglip-large-patch16-384` for experiment in Tab. 3. Since DINOv2 is originally trained with 224×224 inputs and therefore outputs merely 256 patch embeddings, we modify its preprocessing to accept 336×336 resolution so that it produces 576 patch features, making its output shape consistent with CLIP or SigLIP encoders. All experiments are conducted on a compute cluster equipped with eight NVIDIA RTX A6000 GPUs (48GB).

B. Performance with Various Thresholds θ

A key advantage of DiVT is that the similarity threshold θ controls semantic granularity of the clusters. We experiment with $\theta \in \{0.3, 0.4, 0.5, 0.62, 0.65, 0.75, 0.8\}$, spanning coarse to highly fine-grained clustering regimes.

Tab. I provides the full numerical results together with the corresponding average token counts. The table clearly shows a larger θ expands the token budget and this increase tends to align with the observed performance gains across most benchmarks. Once the threshold θ reaches beyond 0.75, however, the clusters become overly fragmented and semantically redundant, leading to a mild degradation in accuracy despite further increases in token count. This behavior confirms that a moderate granularity provides the best trade-off between accuracy and token efficiency.

C. Training and Inference Time Analysis

Tab. II summarizes the computational advantages of DiVT. By treating substantially fewer visual tokens than the 576-token MLP baseline, our DiVT significantly shortens the multimodal forward pass and leads to notable speedups at both training and inference.

The efficiency comparison in Tab. II highlights how DiVT substantially reduces computational cost across pretraining, finetuning, inference, and KV-cache memory usage. The largest improvement appears in the pretraining stage, where the LLM is frozen and the computational cost is largely determined by the sequence length processed through each transformer layer. Lower thresholds significantly shorten this sequence, leading to proportionally large reductions in attention computation and, in turn, overall pretraining time. Finetuning likewise benefits from the reduced token count, though the gains are somewhat moderated by the need to update the full LLM. Still, the lighter visual sequence consistently improves optimization efficiency, providing meaningful savings in both training phases.

Inference reveals additional practical advantages. Because the sequence length linearly scales KV-cache memory, reducing the number of tokens shrinks the KV-cache footprint by over 90% at coarse thresholds such as θ=0.4. Such reductions substantially ease the memory burden and suggest potential scalability benefits for scenarios involving multi-images or video, where limited KV-cache capacity and context length frequently become bottlenecks.

Prefill latency is influenced both by the number of visual tokens and by the cost of our clustering algorithm. At low thresholds, the substantial reduction in token count dominates the overall computation, making the clustering overhead comparatively minor and enabling nearly a 2× speedup over the MLP projector. Even at θ=0.75, where the clustering step becomes slightly heavier, the resulting prefill latency remains close to that of the MLP baseline, indicating that the additional cost introduced by clustering is not a major bottleneck in practice. Crucially, this overhead appears only once during the prefill stage. After tokenization, the subsequent decoding process depends solely on the final number of visual tokens, not on how they were formed. As a result, DiVT benefits from reduced inference-time computation throughout the entire generation process, whereas the MLP projector must continue to handle a much longer visual sequence at every decoding step. This separation demonstrates that the overhead associated with clustering is limited while the gains in end-to-end efficiency are substantial.

Overall, the threshold parameter θ allows practitioners to modulate computational cost with a single control knob, ranging from highly compact and efficient configurations to more detailed representations when resources permit. This simple controllability, together with consistently lower KV-cache usage and reduced decoding cost, makes DiVT an ap-

θ	# Tokens	MMB	VQA^{v2}	GQA	MME	MM-Vet	VQA^{Text}	SQA^{IMG}	POPE
0.3	13.5	64.2	75.3	59.2	1462.8	28.0	55.4	69.4	84.3
0.4	22.4	64.7	76.4	60.1	1450.9	31.7	56.1	69.1	84.8
0.5	35.7	65.0	77.0	60.6	1458.2	31.7	57.1	68.2	85.8
0.62	63.7	64.3	77.7	61.6	1463.0	30.6	57.0	**70.6**	**86.2**
0.65	74.1	65.5	77.7	61.4	**1465.7**	**32.1**	57.2	68.1	85.8
0.75	136.5	**66.7**	**78.2**	**62.0**	1457.6	30.2	**57.7**	70.0	**86.2**
0.8	175.3	65.3	**78.2**	61.9	1456.5	31.3	57.4	68.8	85.8

Table I. **Performance of our DiVT under varying similarity thresholds**

Method	Pretraining (h)	Finetuning (h)	Inference (h)	KV-Cache (MB)	Prefill Latency (ms)
MLP Projector	5.7 (100%)	20.0 (100%)	5.5 (100%)	288.0 (100%)	138.2 (100%)
DiVT$_{\theta=0.4}$	1.1 (19.3%)	12.9 (64.5%)	3.4 (61.8%)	11.0 (3.8%)	71.3 (51.6%)
DiVT$_{\theta=0.5}$	1.4 (24.6%)	13.1 (65.5%)	3.6 (65.5%)	17.6 (6.1%)	76.6 (55.4%)
DiVT$_{\theta=0.65}$	1.9 (33.3%)	13.7 (68.5%)	3.9 (70.9%)	36.8 (12.8%)	104.4 (75.6%)
DiVT$_{\theta=0.75}$	2.7 (47.4%)	14.5 (72.5%)	4.7 (85.5%)	68.1 (23.6%)	138.3 (100.1%)

Table II. **Training and inference cost of DiVT across different similarity thresholds**. Training time is measured using eight RTX A6000 GPUs, and inference time is measured on the VQAv2 evaluation set using a single RTX A6000 GPU. KV-cache memory is computed analytically from the LLaMA-7B architecture, where each visual token contributes approximately 0.5 MB of KV-cache. Prefill latency is measured by averaging multiple stable forward passes after warm-up.

pealing alternative to the MLP projector from an efficiency standpoint.

D. Additional Attention Map Visualizations

Fig. I illustrates additional examples of attention patterns comparing DiVT with the standard MLP projector. To visualize the attention received by each textual token, we aggregate the attention weights assigned to a given DiVT token and project them onto all patches belonging to that token's cluster. This cluster-level visualization highlights which semantic region the model relies on when it processes each text token.

Since DiVT (bottom) aggregates patches into coherent semantic clusters, the resulting attention maps reveal clear and localized patterns. Each textual token tends to focus on a distinct visual concept, making the grounding behavior easy to interpret. In contrast, MLP projectors (top) operate at the patch-level and thus often assign disproportionately high attention to a small subset of tokens, regardless of the query. This obscures which visual evidence the model is using, thereby leading to diffused or noisy activation patterns and hurting interpretability.

E. Additional Cluster Visualization

We provide additional qualitative examples in Fig. II that illustrates how DiVT forms semantically coherent clusters across diverse scenes. Each example assigns a distinct color to patches belonging to the same cluster, allowing us to inspect how the feature-space grouping translates into spatial regions in the original image. As in the main manuscript, the number of clusters is determined dynamically based on the image content, and the resulting visual patterns clearly reflect this adaptivity; that is, relatively simpler scenes yield compact clusters with large spatial support, while more complex or cluttered scenes produce a larger number of fine-grained clusters. Varying the similarity threshold θ also produces the expected behavior, where a higher value leads to a more fragmented grouping.

These examples highlight that DiVT consistently discovers semantic units such as objects, parts, and salient regions without any pixel-level annotation, segmentation masks, or bounding box supervision. Clusters formed purely from feature similarity often align with intuitive semantic boundaries, illustrating the effectiveness of our disentanglement mechanism.

F. Analysis on Resulting Token Counts

Tab. 5 of the main manuscript reports the average number of tokens produced at $\theta = 0.65$. In this section, we extend this by providing mean and standard deviation statistics across multiple thresholds. These measurements are computed over all images within each benchmark, illustrating how DiVT adapts its token budget depending on both the contents of input images and the similarity threshold.

In Tab. III, we observe clear and consistent trends of the resulting token counts across thresholds and datasets. A lower threshold such as $\theta = 0.4$ yields compact token

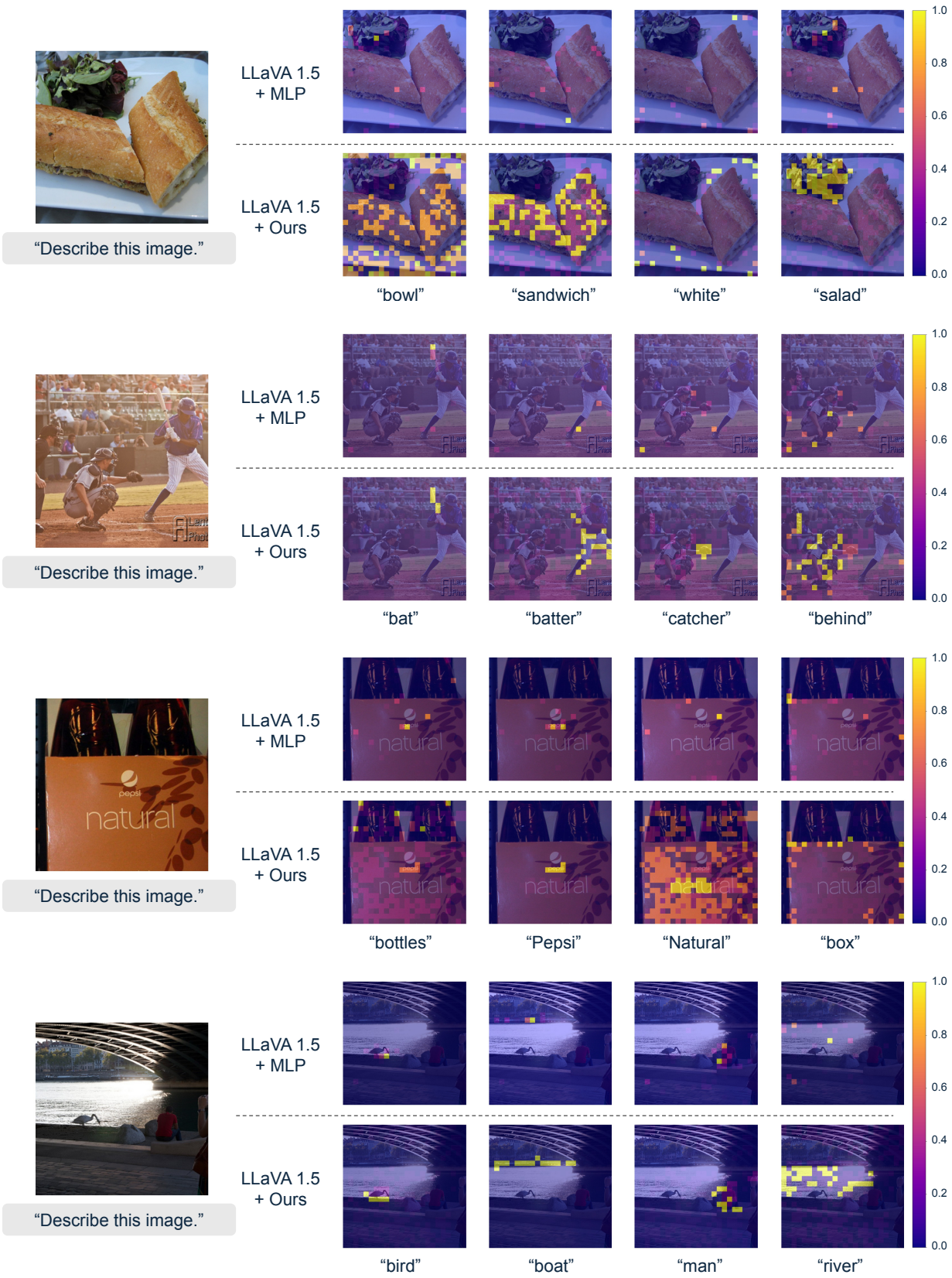


Figure I. **Additional attention map comparisons between DiVT and the MLP projector**. The cluster-based representation in DiVT leads to more consistent and interpretable attention behavior across textual tokens.

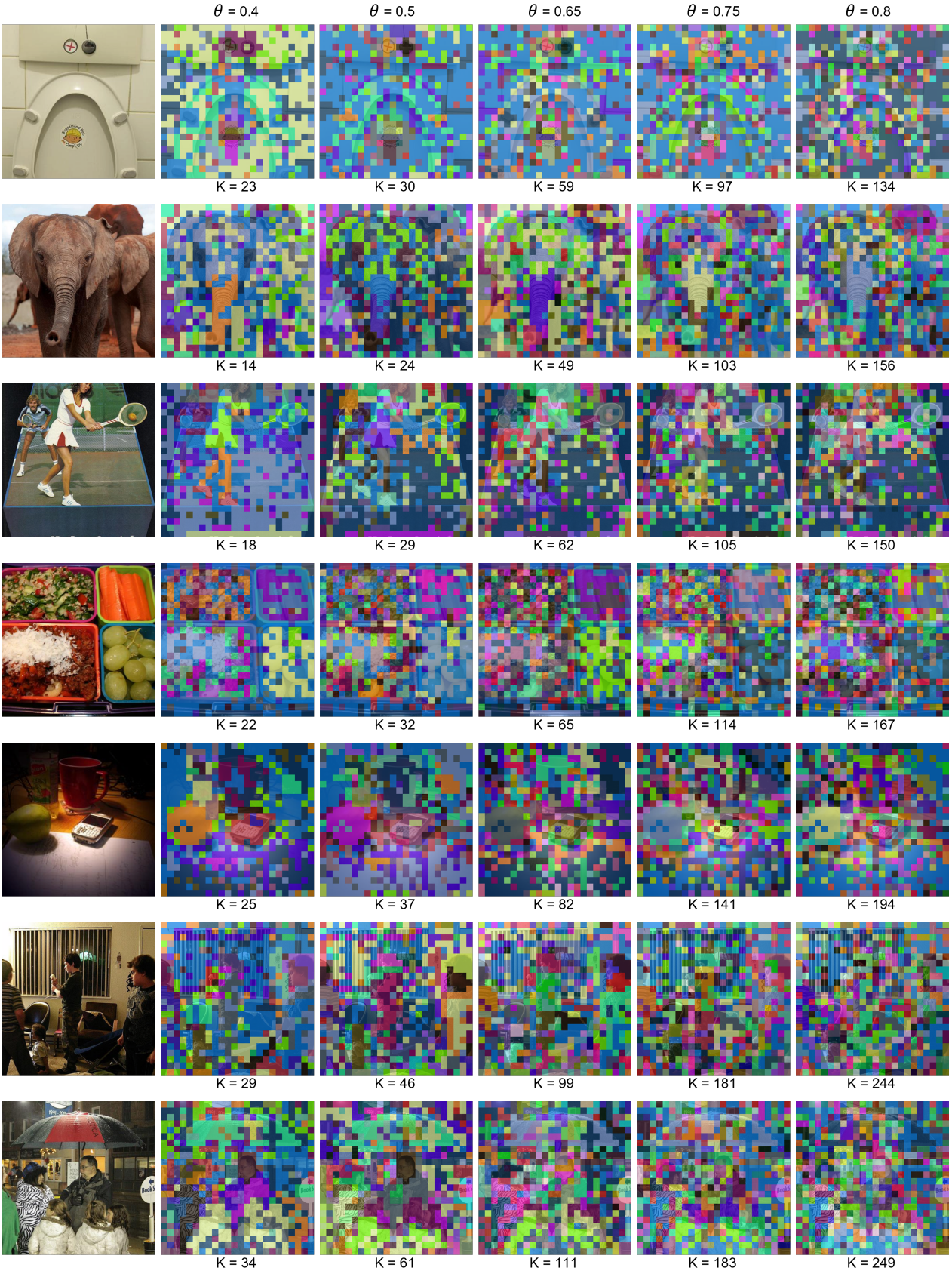


Figure II. **Additional cluster visualizations produced by DiVT.**

θ	**Pretrain**	**Finetune**	**MMB**	**VQA^{v2}**	**GQA**	**MME**	**MM-Vet**	**VQA^{Text}**	**SQA^{IMG}**	**POPE**	**Avg.**
0.3	14.8 ± 8.4	14.9 ± 5.3	11.2 ± 4.3	13.3 ± 3.9	14.6 ± 3.7	12.6 ± 5.1	15.4 ± 7.3	18.3 ± 8.1	10.6 ± 5.0	14.3 ± 3.9	13.5 ± 4.3
0.4	22.1 ± 11.7	23.7 ± 7.9	18.0 ± 6.5	22.0 ± 6.4	24.7 ± 6.2	20.5 ± 7.2	23.0 ± 9.9	28.7 ± 11.8	16.2 ± 6.5	24.0 ± 6.2	22.4 ± 6.9
0.5	32.4 ± 15.7	37.2 ± 12.2	27.1 ± 10.9	35.1 ± 10.3	40.0 ± 10.5	32.5 ± 11.4	33.0 ± 13.2	45.8 ± 19.4	22.4 ± 8.1	39.3 ± 11.8	35.7 ± 11.4
0.62	54.9 ± 23.4	66.0 ± 18.8	50.0 ± 17.6	63.1 ± 18.3	70.1 ± 17.1	59.1 ± 18.3	57.1 ± 18.1	72.8 ± 24.7	41.6 ± 13.4	68.7 ± 18.2	63.7 ± 18.9
0.65	62.3 ± 26.0	76.5 ± 24.7	58.3 ± 20.5	73.5 ± 21.1	81.3 ± 19.3	69.3 ± 21.3	65.2 ± 19.8	83.4 ± 27.7	48.3 ± 15.7	80.1 ± 21.0	74.1 ± 21.8
0.75	110.3 ± 42.0	138.8 ± 41.2	108.4 ± 42.0	136.2 ± 38.5	146.2 ± 33.8	133.2 ± 44.8	114.2 ± 36.1	145.8 ± 47.3	88.5 ± 31.1	147.1 ± 39.2	136.5 ± 39.6

Table III. **Resulting token counts of our proposed method across datasets for multiple thresholds**. Reported as mean $\pm$ standard deviation. The Avg. column averages over evaluation benchmarks only.

sets with small variance, as visually dominant regions are merged into broader clusters. Increasing θ makes the clustering more selective, producing more finer-grained tokens and higher token-count variance, particularly on benchmarks containing text, cluttered objects, or complex compositions (*e.g.*, TextVQA or POPE). Conversely, datasets with simpler scenes, such as SQA-IMG, maintain a narrow token-count range across all thresholds.

Collectively, DiVT adjusts its token budget according to the inherent visual complexity of each image rather than relying on a fixed grid-based reduction. The resulting distribution of token counts demonstrates that DiVT responds naturally to semantic density, enabling compute-efficient representations without sacrificing expressiveness.