A Diagnostic Study of Region-Based Representations in Multimodal LLMs

Supplementary Material

A. Experiment Details

In this section, we provide comprehensive details with regard to our model training and inference, as well as our computational resources.

A.1. Implementation Details

Generating Regions from SAM. As mentioned in the main paper, we use SAM2's [4] automatic mask generator for deriving the segmentation of the whole image. In our main experiments, we automatically select the granularity of segmentation from `points-per-side` $= \{48, 64, 96\}$, `stability-score-thresh` $= \{0.92, 0.90, 0.85\}$, `pred-iou-thresh` $= \{0.6, 0.5, 0.4\}$, to soft-bound the number of regions within $[80, 160]$. However, we later observed that using a much coarser granularity of `points-per-side` $=$ 24, `stability-score-thresh` $=$ 0.9, `pred-iou-thresh` $=$ 0.5 only brings negligible changes in performance, which allows us to complete the SAM mask generation in $\sim$ 162ms per image. We also explored integrating Mosaic Augmentation into RADIOv2.5 [2] to fit multiple images within a single SAM input. While this approach could have halved mask generation time, it was ultimately excluded after empirical tests showed a degradation in performance.

Different from the default implementation of upsampling the features into the size of the masks [5], we alternatively downsample the masks into the size of the patch features to reduce both GPU memory usage and computation cost, as well as aligning with the setting of the clustering alternative. Specifically, masks are first preprocessed into the same size as the input resolution of the visual encoder (*e.g.*, 384×384, following the same preprocessing steps like padding and resizing as the input image), and then downsampled to the size of the patch-level feature map (*e.g.*, 24×24). Since each image patch is reduced to one pixel in the patch-level feature map, we alternatively use the following downsampling method: we average pool masks within each image patch, and those pixels with pooled values higher than a fixed threshold are retained in the mask. This means a downsampled pixel is retained if and only if a certain fraction of the original patch is contained in the mask. By setting a small threshold 0.07, we can avoid small regions from vanishing unless they are extremely small.

Generating Regions from Clustering. We mainly adopt UnSAM's [7] iterative merging as the clustering method:

1. We start by setting each patch as a single region;
2. A region's feature is defined by the average of the patch features covered by the region;
3. We iteratively merge adjacent regions with the highest feature cosine similarity, until it drops below a certain threshold;
4. The remaining regions after iterative merging are considered to be the result regions.

We set the similarity threshold of clustering to be 0.7, and additionally use priority queues and disjoint sets to optimize the clustering process. When combining segmentation and clustering, we use DBSCAN [1] with `metric` $= L_2$, `eps` $= 0.7$, `min samples` $= 3$ to further split the segmentation masks containing at least 10 patches into smaller regions using the normalized patch features.

A.2. Hyperparameters

Training. We adopt the same training hyperparameters of LLaVA-1.5 [3]: `total batch size` $= 256$, `learning rate` $= 10^{-3}$ for visual feature alignment, `total batch size` $= 128$, `learning rate` $= 2 \times 10^{-5}$ for visual instruction tuning. For both stages, we train only one epoch with `bfloat16` precision using a fixed seed 42.

Inference. During inference, we set `temperature` $= 0$, or equivalently, greedy decoding on all benchmarks. For short answer tasks, we unify the question prompt to be *<image>+ [question] + "Answer the question using a single word or phrase."* For QA tasks, we unify the question prompt to be *<image>+ [question] + [options] + "Answer with the option's letter from the given choices directly."*

Cross-Attention Aggregation. We use 16 heads with a single query token from the average-pooled patch feature in the cross-attention region feature aggregation.

A.3. Computational Resources

All training experiments are conducted on 4 NVIDIA H100 GPUs, which take roughly 16 hours to complete each two-stage training under the default setting for patch-based representations. Other settings have fluctuated training time from 7 hours to two days, depending on the specific setting. Inference is conducted on a single NVIDIA H100 GPU.

B. Additional Results

In this section, we provide the full evaluation results, as well as additional visualizations of region segmentation examples, visual features, and attention patterns that are not presented in the main paper.

B.1. Complete Experimental Results

Due to page limits, some of the results are not shown in the main paper. Tab. 1 presents the complete results of the seven benchmarks under all investigated settings. Additionally, we utilize the annotations provided by PixCV-Bench [6], which contain segmentation masks of the object of interest for questions in CV-Bench, to compute a *focus metric* for quantitatively evaluating the attention map interpretability. Specifically, for each regular visual token, if the patch/region it represents overlaps with the mask annotation by a certain threshold, it would be considered a target token. The focus metric is then defined by the averaged total attention score of the answer tokens attending to target visual tokens.

B.2. More Visualizations

Fig. 1 visualizes the visual feature norms and MLLM attentions when using normalization on patch-based representations for three visual encoders. As we have introduced in the main paper, normalized features suppress high-norm outliers, and in general produce more diverse and localized attentions focusing on multiple patches around the target object, especially for RADIOv2.5 [2].

Fig. 2 shows additional patch feature visualization results on more visual encoders. Consistent with our observations in the main paper, only self-supervised visual encoders DINOv2 and agglomerative visual encoders RADIOv2.5 produce relatively coherent features.

Fig. 3 shows more visualizations of MLLM's attentions over visual tokens under different settings. On most examples, MLLMs pay more attention to the target object of the question when using region-based representations compared with patch-based ones.

C. Broader Impacts

This work shares the common risks associated with other MLLMs, including the potential to introduce or amplify existing societal biases. While we rely solely on publicly available benchmarks, models, and training data, which avoids private or personal information, biases present in these resources may still influence final outcomes. We do not specifically target fairness mitigation in this study, but we recognize its importance and encourage future research to address these concerns. All resources used are publicly released to support transparency, reproducibility, and community-driven scrutiny.

References

[1] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, 1996. 1

[2] Greg Heinrich, Mike Ranzinger, Hongxu, Yin, Yao Lu, Jan Kautz, Andrew Tao, Bryan Catanzaro, and Pavlo Molchanov. RADIOv2.5: Improved baselines for agglomerative vision foundation models. In *CVPR*, 2025. 1, 2

[3] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2024. 1

[4] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 1

[5] Michal Shlapentokh-Rothman, Ansel Blume, Yao Xiao, Yuqun Wu, Sethuraman TV, Heyi Tao, Jae Yong Lee, Wilfredo Torres, Yu-Xiong Wang, and Derek Hoiem. Region-based representations revisited. In *CVPR*, 2024. 1

[6] Mennatullah Siam. PixFoundation: Are we heading in the right direction with pixel-level vision foundation models? *arXiv preprint arXiv:2502.04192*, 2025. 2

[7] Xudong Wang, Jingfeng Yang, and Trevor Darrell. Segment anything without supervision. In *NeurIPS*, 2024. 1

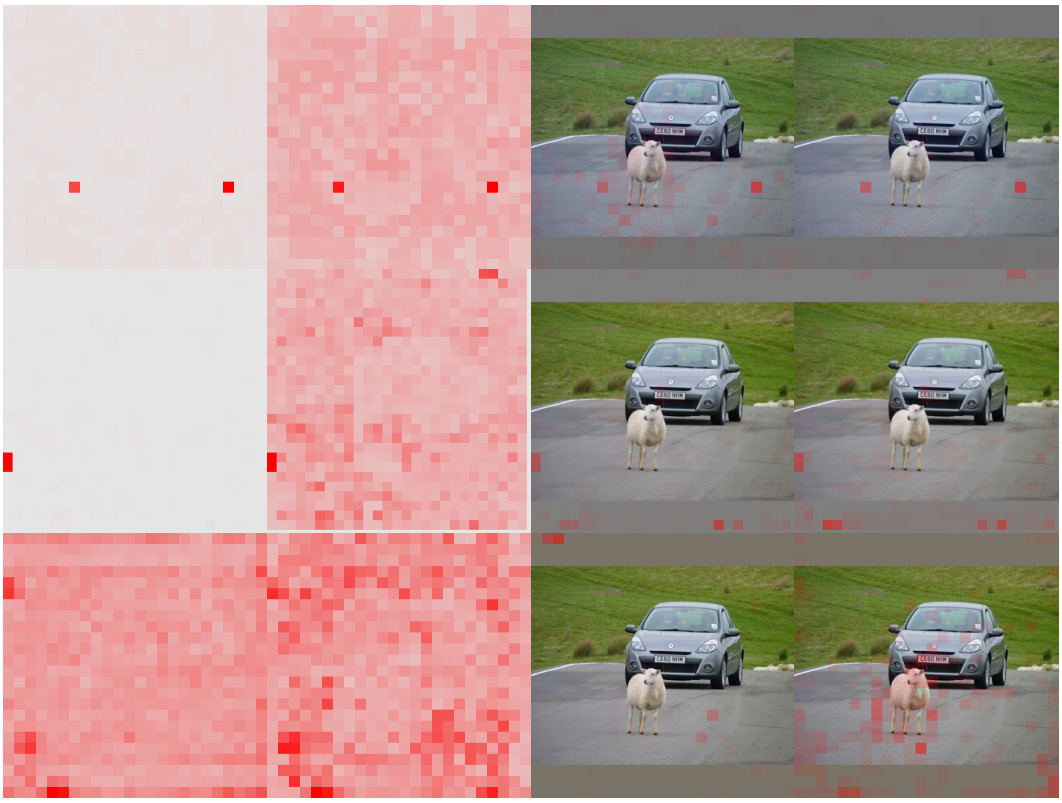


Figure 1. **Feature norms and attention patterns. Left:** feature norms before and after normalization. **Right:** attention visualizations before and after normalization for patch-based representations. The three rows from **top** to **bottom** correspond to using CLIP, SigLIP2, and RADIOv2.5 as the visual encoder respectively. Normalized features suppress high-norm outliers and in general produce more diverse and localized attentions.

Table 1. **Complete evaluation results as an addition to our main results.** Some settings are not included in the main paper due to space limit. If not specifically mentioned, we use the default token order defined in the main paper. Some benchmarks on minor settings are not evaluated due to limited computation resources, which are marked with **-**.

Region Source	Res	Norm	Token Order	#Tok	Focus	POPE	OCRBench	CV-Bench	MMStar	MME		MMBench	MM-Vet
										Perception	Cognition		
CLIP													
Patch		no		576	12.54	86.05	331	55.82	33.47	1500	300	66.84	32.1
Patch		rms		576	11.96	85.39	321	56.09	32.87	1477	285	66.58	32.5
Patch		rms	random	576	-	85.74	312	58.80	34.27	1493	284	65.98	30.2
Segmentation	336	rms		101	15.54	86.21	275	57.69	34.67	1414	320	65.38	29.7
Clustering		rms		257	12.72	85.80	310	57.66	35.80	1493	306	66.84	29.6
Combined		rms		139	14.41	85.48	288	57.24	33.73	1386	316	66.15	32.0
SigLIP2													
Patch		no		729	11.31	85.99	405	60.03	35.27	1463	321	67.53	35.1
Patch		rms		729	11.51	86.17	389	57.67	34.93	1477	333	67.87	34.3
Segmentation	378	rms		101	15.24	84.96	301	57.45	34.07	1441	311	65.81	33.3
Clustering		rms		334	8.34	85.68	379	59.68	36.33	1542	324	69.67	35.8
Combined		rms		142	14.21	85.40	335	59.63	33.40	1457	361	67.18	33.2
RADIOv2.5													
Patch		no		576	10.57	85.82	315	56.92	36.00	1494	305	67.35	29.6
Patch		no	random	-	-	85.10	313	57.45	36.33	1472	289	-	-
Patch		rms		576	12.44	85.25	314	57.61	34.80	1464	326	65.64	30.5
Segmentation		no		101	16.29	84.82	237	55.96	32.53	1371	333	63.57	25.3
Segmentation	384	rms		101	16.22	84.32	250	56.95	33.60	1394	327	64.60	25.7
Clustering		no		117	13.50	84.91	273	58.31	33.13	1463	354	65.12	23.8
Clustering		rms		124	13.39	84.51	280	58.10	35.47	1456	350	65.38	26.7
Combined		no		134	14.86	84.66	260	58.20	34.00	1436	336	65.03	30.3
Combined		rms		134	14.88	84.76	264	59.35	33.27	1420	330	65.12	25.7
Patch	576	no		1296	10.91	86.92	357	59.58	33.40	1498	296	-	-
Segmentation	576	rms		104	15.80	85.91	270	57.51	35.33	1459	326	-	-
Combined	576	rms		159	14.72	86.05	284	57.07	35.80	1479	311	-	-
RADIO, without cross-attention aggregation													
Segmentation	384	no		-	-	84.14	252	56.62	34.00	1432	324	-	-
Clustering	384	no		-	-	84.36	271	57.82	32.80	1451	288	-	-
Combined	384	no		-	-	85.03	259	54.89	33.47	1432	305	-	-

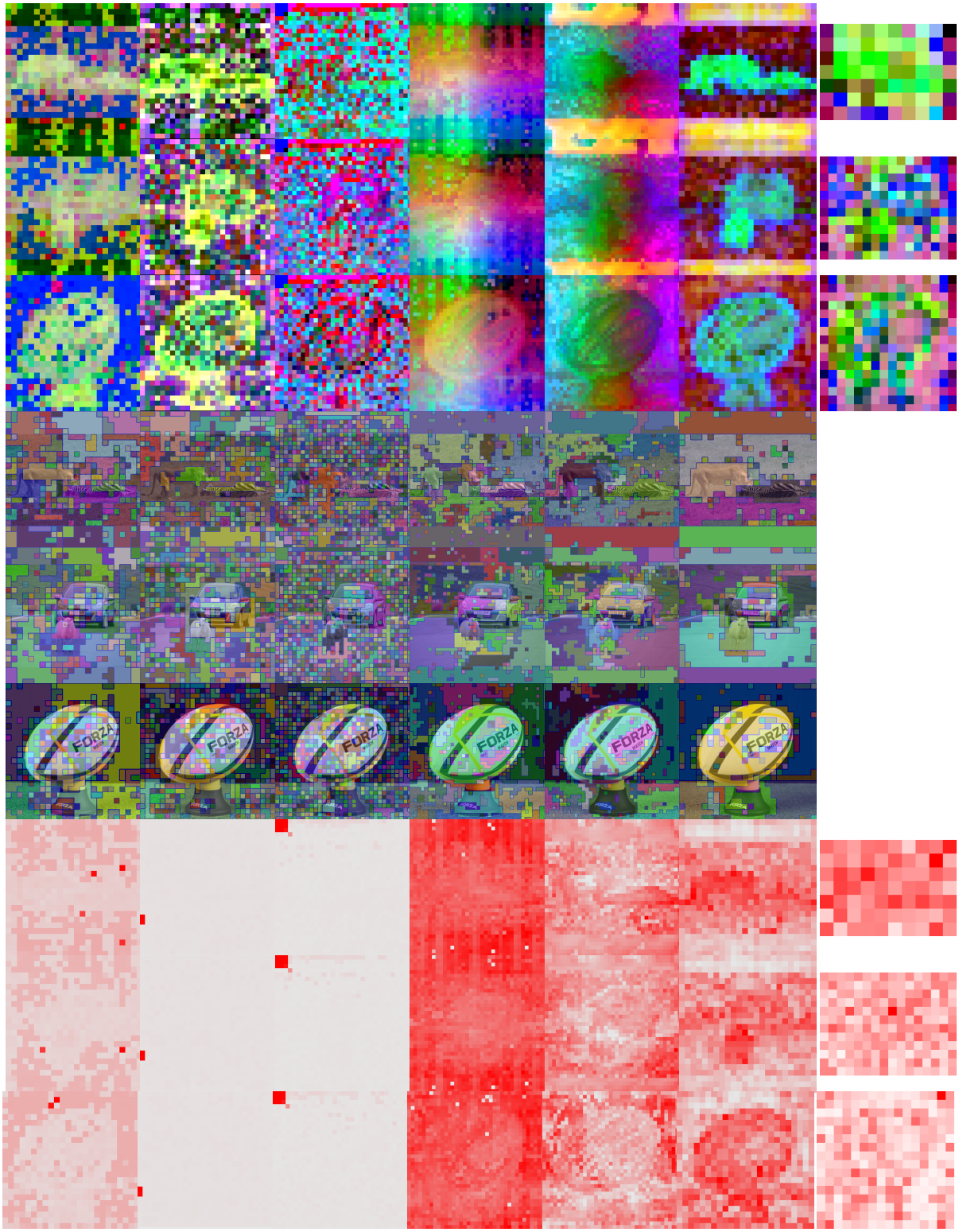
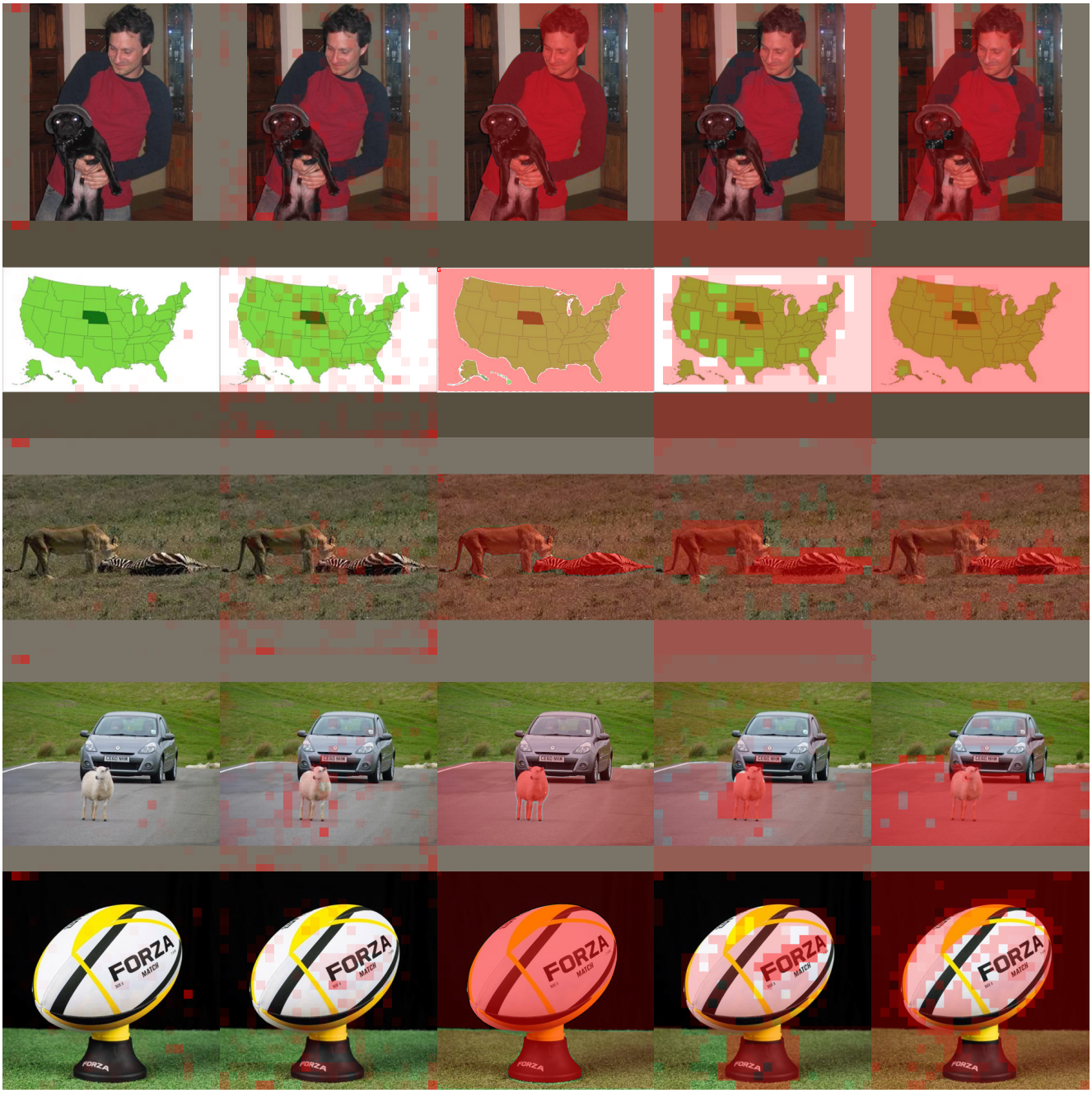


Figure 2. **More visualization results for different visual encoders. Rows 1–3:** patch features (PCA); **Rows 4–6:** regions from clustering; **Rows 7–9:** patch feature norms. Only DINOv2, DINOv2-reg, and RADIOv2.5 produce relatively coherent features. All results are derived from their native resolution, except for RADIOv2.5, where we use 384×384. We use `aimv2-large-patch14-448` for AIMv2, `vit_large_patch14_dinov2.lvd142m` for DINOv2, and `vit_large_patch14_reg4_dinov2.lvd142m` for DINOv2-reg.



(a) Patch (b) Patch (RMSNorm) (c) Segmentation (d) Clustering (e) Combined

Figure 3. **More visualization results for MLLM attention over visual tokens.** When using region-based representations, MLLMs pay more attention on the target object of the input query.