Uncovering Object Localization Mechanisms in VLMs

Anonymous Author(s)
Affiliation
Address
email

Abstract

Visually-grounded language models (VLMs) are highly effective in linking visual and textual information, yet they often struggle with basic classification and localization tasks. While classification mechanisms have been studied more extensively, the processes that support object detection remain less clear. In this work, we analyze foundational VLMs and show that image tokens corresponding to the object directly contain the information required for localization. We find that the model applies a containerization mechanism: it uses object-related tokens to define spatial boundaries, while largely discarding semantic context. Our analysis further reveals that this information is processed in the early to middle layers of the language model and that classification and detection rely on shared mechanisms. Finally, we demonstrate that spatial grounding does not come solely from positional encodings in the visual backbone, but rather from residual positional signals combined with the language model's ability to infer spatial order from token sequences.

1 Introduction

Visually-grounded Language Models (VLMs) combine a pre-trained vision encoder with a large language model (LLM), typically refined through vision-language instruction tuning. The visual encoder extracts grid-level features from an image, a multimodal adapter maps them into the language embedding space, and the resulting tokens are processed jointly with text by the LLM. This architecture allows VLMs to link visual and textual inputs and has enabled strong performance on tasks such as visual question answering, captioning, and open-ended reasoning about images [13, 5, 6, 1].

Despite these advances, VLMs continue to struggle with core vision tasks. They often misclassify or fail to accurately localize objects [20, 23]. While the mechanisms underlying classification have been studied [14, 23], much less is known about localization and detection. Closing this gap is important because most VLMs inherit visual features from CLIP [17], which was trained with global image-text supervision and struggles with the pixel-level precision required for localization and detection [3, 18, 24]. Yet VLMs can still answer queries that require identifying and locating objects, suggesting that these models build spatial structure from weak visual signals. This raises the question of how the mechanisms enabling localization and detection emerge in VLMs.

In this paper, we present an initial study of the mechanisms underlying object detection in VLMs. Our main findings are:

1. **Grounding through containerization.** Information needed for localization is directly encoded in the visual tokens. The model groups these tokens into *containers* that define object boundaries, largely independent of semantics.
2. **Implicit spatial layout learning.** The LLM infers the two-dimensional structure of the image from the one-dimensional token sequence by learning implicit line breaks, without spatial modeling.
3. **Early–middle layer processing.** The mechanisms that support detection emerge in the early to middle layers of the LLM, indicating overlap with the layers that drive classification.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

Related Work Several studies have focused on improving localization performance of VLMs [22, 16, 6, 1], but none analyze their underlying mechanisms. Prior mechanistic interpretability studies of VLMs focus on reasoning, VQA, or hallucinations ([2, 9, 15, 21, 11, 14]). We build on ablation and attention knockout methods [14], adapting them to study localization mechanisms in object detection tasks.

2 Method

Visually Grounded Language Models We study two variants of LLaVa 1.5 [13], an open-source VLM with a simple, yet effective architecture. Visual features are encoded by CLIP ViT-L/14 [17] and projected into the LLM embedding space via a two-layer MLP. We analyze the two available versions: LLaVa-7B and LLaVa-13B, which use Vicuna-7B/13B [4] as the language backbone, respectively.

Dataset We filter the COCO [12] training split to obtain a controlled subset of 8,442 images. Specifically, we retain images containing 1-4 annotated objects, with no more than one instance per category to avoid category-level ambiguity. Each object is further required to cover 10%-50% of the image area. Following [14], we additionally control for hallucinations by keeping only samples where the model correctly identifies objects in the original image, but fails when objects are masked with noise. This ensures that predictions rely on visual evidence rather than context.

Task We compare model performance across two tasks: (i) **Classification**: We prompt the model with the query `‘‘Is there {art} {cls} in the image?’’`, where `cls` denotes the object category and `art` the appropriate article. Predictions are evaluated by searching for the token “Yes” in the response, and success rate is reported as the fraction of correctly classified instances. (ii) **Detection**: We prompt the model with `‘‘Please provide the bounding box coordinates of the {cls}.’’` The predicted bounding boxes are parsed and compared against ground-truth annotations using the intersection-over-union (IoU) metric. Performance is measured as the success rate, defined as the proportion of samples where IoU exceeds thresholds of 0.5, 0.7, and 0.9. The final detection score is obtained by averaging over these three thresholds.

3 Experiments

3.1 Object Localization

Method We conduct an ablation study to investigate the contribution of visual input tokens to the performance of the VLM on the classification and detection task. To preserve domain-consistent statistics while effectively removing image-specific information, we replace the original image tokens with their average embedding computed over the ImageNet [7] validation set.

We evaluate three strategies for selecting visual tokens for ablation: (i) *Object*: We project the object mask onto the image token grid and include all tokens that overlap with it by at least one pixel. To probe for boundary sensitivity and context dependence, we shrink or dilate the mask by 1 or 2 token padding. A visualization of the masking is provided in the Appendix Figure 3. (ii) *Integrated Gradients*: We identify the top k important image tokens using the Integrated Gradients attribution method [19], computed over 50 steps with respect to the “Yes” logit (for classification) or bounding box coordinates (for detection). (iii) *Random*: As a control, we randomly select k image tokens, repeat the process with three different seeds, and report the standard deviation.

Result As Table 1 shows, both detection and classification tasks rely on the information encoded in object tokens, as ablating them results in a significantly larger performance decline compared to removing an equal number of tokens either randomly or via gradient-based selection. Detection is more affected than classification: removing object tokens reduces detection performance nearly to zero, while classification still succeeds in 60–80% of cases. Positive padding around the object further amplifies the effect, while maintaining the original boundaries through negative padding has minimal influence on performance. These findings indicate that the essential information for both tasks resides within the object boundaries.

Next, we investigate the mechanisms by which the model encapsulates objects to generate bounding boxes. To test this, we artificially expand the ground-truth mask by padding it with p layers of tokens.

Table 1: Performance after token ablation. The baseline indicates no performed token ablation and is compared to ablating the object mask with different paddings, highest-gradient and random token ablation.

	LLaVA 7B		**LLaVA 13B**	
Ablation (*Avg. Token Count*)	Detection (%)	Classification (%)	Detection (%)	Classification (%)
Baseline (*0*)	54.79	99.25	65.96	99.46
– 2 Padding (*31*)	53.17	99.13	66.61	99.37
– 1 Padding (*67*)	46.41	98.51	63.27	98.76
Object (*122*)	**8.11**	**63.70**	**14.61**	**77.36**
+ 1 Padding (*176*)	1.46	36.47	2.00	54.81
+ 2 Padding (*230*)	0.76	33.84	1.11	48.50
Integrated Gradients (*100*)	23.90	93.52	19.29	96.32
Integrated Gradients (*200*)	8.53	79.83	5.77	83.26
Random (*100*)	52.44 ± 0.65	99.29 ± 0.01	62.79 ± 0.30	99.42 ± 0.02
Random (*200*)	47.82 ± 1.85	99.29 ± 0.05	57.96 ± 0.32	99.37 ± 0.02

Concretely, we sample tokens from the existing object and place them into the surrounding padding region, effectively extending the object's footprint while keeping its semantic content unchanged. We then measure whether the predicted bounding box expands accordingly across three random sampling seeds. For LLaVA-7B, detection accuracy is $61.84 \pm 0.09\,\%$ with $p = 1$ and $66.56 \pm 0.10\,\%$ with $p = 2$. For LLaVA-13B, average accuracy is $71.79 \pm 0.21\,\%$ at $p = 1$ and $78.97 \pm 0.17\,\%$ for $p = 2$. These results indicate that detection depends mainly on the presence of object tokens rather than their specific semantic arrangement. We show qualitative examples in Appendix Figure 4.

To further support this claim, we shuffle the image tokens within the object mask directly at the LLM input. As shown in App. Table 2, detection performance remains unchanged under this perturbation. Together, these findings suggest that the model employs a form of **containerization**, where tokens collectively define an object's spatial extent, independent of their internal semantic structure.

3.2 Position Encoding

Method We next study how positional information is processed within the VLM, focusing on the role of the visual backbone's positional encoding (PE) and the subsequent token ordering in the LLM for spatial grounding. To study how positional information remains identifiable throughout the model's hierarchical processing, we train a separate linear classifier for each model layer to predict the position of an image token within the 24×24 input grid (576 classes). The classifiers are trained for 10 epochs using 50,000 ImageNet images for training and 10,000 for testing.

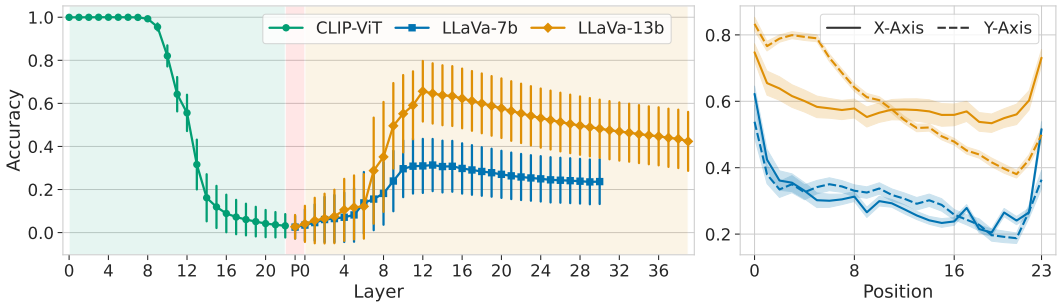

Figure 1: Positional decoding results. Left: average position accuracy per layer for CLIP (0-22), the multimodal projection (P) and LLaVA language models (0-39). Right: per-position accuracy at layer 11, showing higher accuracy at the image corners.

Result In Figure 1, we observe that positional information in the CLIP visual backbone is decodable from early layers but largely vanishes by the final layers, consistent with prior findings that CLIP-ViT trades spatial precision for semantic abstraction over depth [10]. In contrast, in the LLM, positional

identifiability is initially low. However, it increases rapidly and peaks around layers 12 and 13. The multimodal projection retains strong signals for the four image corners (app. Fig. 5), which appear sufficient for the LLM to infer approximate row boundaries ("line breaks") across the token sequence. Tokens aligned with these inferred line breaks are predicted with higher probability than other positions (Fig. 1, right), suggesting that the model uses them as structural anchors when reconstructing spatial layouts. In App. B, we provide additional experiments supporting these claims.

3.3 Attention Knockout

Method Our next experiments aim to identify where in the network the visual information that is required for the task is extracted and processed. We apply the *attention knockout* technique [8, 14], which blocks attention and thereby prevents the communication between tokens. Unlike [14], we eliminate attentions from all tokens following the image tokens to the object tokens, effectively removing any information extracted from the object tokens in those layers. We combine layers in groups of four and block all attention heads inside each group. We evaluate the resulting performance drop on classification and detection tasks using a 1,024-image subset of our filtered COCO dataset.

Result Our results are shown in Figure 2. Blocking attention to the object significantly decreases performance in the early to mid layers of the model, while perturbing attention flow in later layers leaves performance largely unaffected. Both detection and classification rely on early shared layers, after which detection depends on additional task-specific layers. This progression suggests a two-step process: the model first identifies the object, then localizes it. Moreover, the layers with the largest detection decline align with those that retain strong positional information (see Sec. 3.2). Compared to layer-wise knockouts (app. Fig. 6), ablating groups of layers amplifies the effect, particularly for classification. This suggests that the model accumulates object-related information across consecutive layers rather than relying on layer-specific mechanisms.

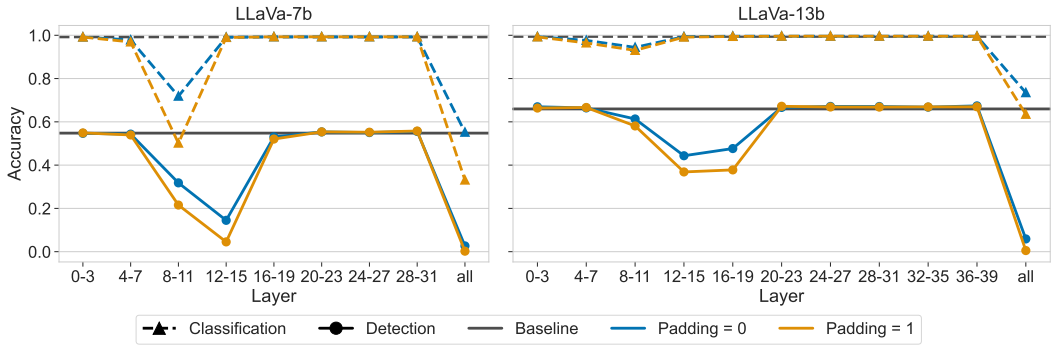

Figure 2: Effect of attention knockout on classification and detection. Performance drops mainly in early and mid layers, while later layers remain unaffected.

4 Discussion and Limitations

Our findings shed light on the fundamental mechanisms through which VLMs capture and encode spatial structure. Our experiments reveal that positional information is reconstructed in the LLM, rather than relying on positional information encoded in the visual backbone. Localization depends on a containerization process in which object tokens collectively define spatial boundaries, largely independently of semantics. Attention knockout reveals that detection and classification rely on overlapping early-middle layers, with localization emerging after object identification. These results refine our understanding of the inner workings of VLMs and open several directions for future work.

We focus our study on LLaVa as a representative VLM, but future work could extend the analysis to models with other architectural choices such as visual token compression, alternative positional encodings, or richer grounding objectives. Our experiments address the simplest case of single-object localization, which provides a foundation for exploring more complex relative localization tasks. Moreover, our methods could be extended to examine the mechanisms of broader capacities, such as reasoning or hallucination.

References

[1] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023. URL https://arxiv.org/abs/2308.12966.

[2] S. Basu, M. Grayson, C. Morrison, B. Nushi, S. Feizi, and D. Massiceti. Understanding information storage and transfer in multi-modal large language models. *Advances in Neural Information Processing Systems*, 37:7400–7426, 2024.

[3] W. Bousselham, F. Petersen, V. Ferrari, and H. Kuehne. Grounding everything: Emerging localization properties in vision-language transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3828–3837, 2024.

[4] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, and E. P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

[5] W. Dai, J. Li, D. Li, A. Tiong, J. Zhao, W. Wang, B. Li, P. N. Fung, and S. Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in neural information processing systems*, 36:49250–49267, 2023.

[6] M. Deitke, C. Clark, S. Lee, R. Tripathi, Y. Yang, J. S. Park, M. Salehi, N. Muennighoff, K. Lo, L. Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 91–104, 2025.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[8] M. Geva, J. Bastings, K. Filippova, and A. Globerson. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12216–12235, 2023. URL https://aclanthology.org/2023.emnlp-main.751/.

[9] M. Hinck, C. Holtermann, M. L. Olson, F. Schneider, S. Yu, A. Bhiwandiwalla, A. Lauscher, S.-Y. Tseng, and V. Lal. Why do LLaVA vision-language models reply to images in English? pages 13402–13421. Association for Computational Linguistics. URL https://aclanthology.org/2024.findings-emnlp.783/.

[10] D. Jiang, Y. Liu, S. Liu, X. Zhang, J. Li, H. Xiong, and Q. Tian. From clip to dino: Visual encoders shout in multi-modal large language models. *arXiv preprint arXiv:2310.08825*, 2023.

[11] N. Jiang, A. Kachinthaya, S. Petryk, and Y. Gandelsman. Interpreting and editing vision-language representations to mitigate hallucinations. *arXiv preprint arXiv:2410.02762*, 2024.

[12] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[13] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[14] C. Neo, L. Ong, P. Torr, M. Geva, D. Krueger, and F. Barez. Towards interpreting visual information processing in vision-language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=chanJGoa7f.

[15] V. Palit, R. Pandey, A. Arora, and P. P. Liang. Towards vision-language mechanistic interpretability: A causal tracing tool for blip. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2856–2861, 2023.

[16] Z. Peng, W. Wang, L. Dong, Y. Hao, S. Huang, S. Ma, Q. Ye, and F. Wei. Grounding multimodal large language models to the world. In *The Twelfth International Conference on Learning Representations*, 2024.

[17] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[18] T. Shao, Z. Tian, H. Zhao, and J. Su. Explore the potential of clip for training-free open vocabulary semantic segmentation. In *European Conference on Computer Vision*, pages 139–156. Springer, 2024.

[19] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

[20] S. Tong, Z. Liu, Y. Zhai, Y. Ma, Y. LeCun, and S. Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9568–9578, 2024.

[21] Z. Wang and K. Wang. Multishap: A shapley-based framework for explaining cross-modal interactions in multimodal ai models. *arXiv preprint arXiv:2508.00576*, 2025.

[22] H. Zhang, H. Li, F. Li, T. Ren, X. Zou, S. Liu, S. Huang, J. Gao, Leizhang, C. Li, et al. Llava-grounding: Grounded visual chat with large multimodal models. In *European Conference on Computer Vision*, pages 19–35. Springer, 2024.

[23] Y. Zhang, A. Unell, X. Wang, D. Ghosh, Y. Su, L. Schmidt, and S. Yeung-Levy. Why are visually-grounded language models bad at image classification? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://openreview.net/forum?id=MwmmBg1VYg`.

[24] Y. Zhong, J. Yang, P. Zhang, C. Li, N. Codella, L. H. Li, L. Zhou, X. Dai, L. Yuan, Y. Li, et al. Regionclip: Region-based language-image pretraining. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16793–16803, 2022.

Appendix

We include additional experiments and visualizations in the appendix to support our statements.

A Ablation

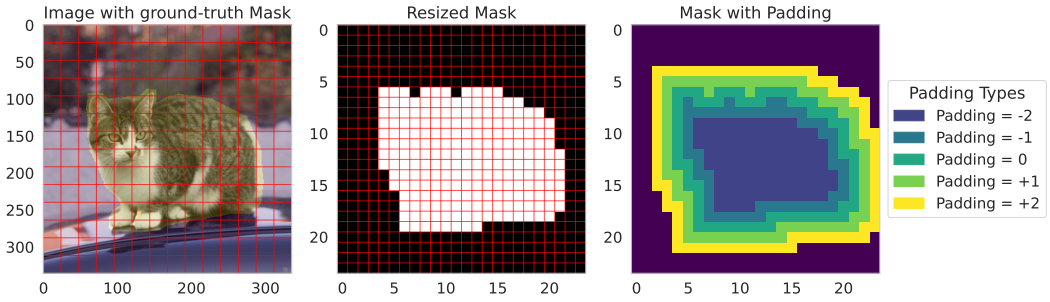

Figure 3: **Visualization of object mask for the ablation experiment**. Left: the original 336×336 input image in pixel space with an annotated object mask. This mask is mapped onto the 24×24 token grid of the vision transformer, where a token is selected if it has any pixel overlap with the original mask. Right: examples of padding applied to the token mask. Negative padding removes adjacent tokens and shrinks the ablated region, while positive padding adds neighboring tokens and expands it.

B Position Encoding

We support our findings from Section 3.2 with controlled shuffling experiments that selectively perturb positional information at different stages of the multimodal pipeline.

Token Order Shuffling We first shuffle the sequence of visual tokens after the multimodal projection but before they are passed into the LLM. This preserves all positional information encoded by the visual backbone while disrupting the sequence order. As shown in Table 2, detection performance drops to almost zero, while classification performance remains unaffected. This indicates that the LLM does not mainly utilize spatial signals from the visual backbone. Instead, it depends strongly on

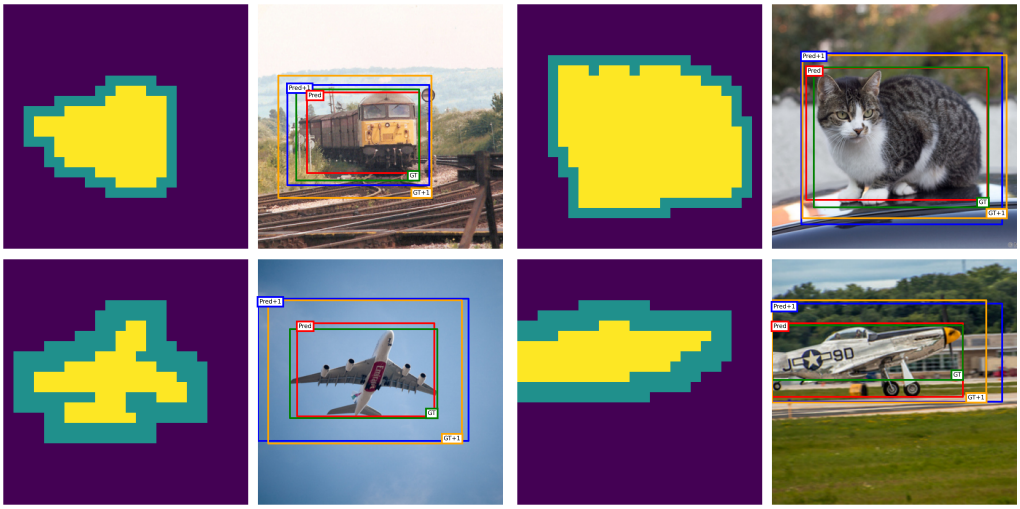

Figure 4: **Examples of the object extension experiment.** Each image shows the input with its mask in pixel space. The yellow region indicates the original mask, while the green region denotes the padding p added by sampling tokens from the object. The top row corresponds to $p = 1$ and the bottom row to $p = 2$. We display both the predicted and ground-truth bounding boxes for the original and the extended object. The predicted boxes expand consistently with the mask, suggesting that the model containerizes object tokens to define spatial boundaries.

Table 2: Detection and classification results for LLaVA models under different token shuffling tests. Reported values include the standard deviation over 10 seeds. We distinguish between shuffling visual tokens at the LLM input and shuffling positional encodings at the visual backbone input. Complete shuffle randomizes all tokens, while mask shuffle restricts randomization to tokens inside the object mask. Row/Column shuffle permutes tokens within rows/columns while keeping columns/rows fixed.

		LLaVA 7B		**LLaVA 13B**	
Model Position	Shuffle	Detection (%)	Classification (%)	Detection (%)	Classification (%)
-	Baseline	54.79	99.25	65.96	99.46
LLM Tokens	Complete	4.67 ± 0.07	99.19 ± 0.03	1.06 ± 0.04	99.34 ± 0.02
	Mask	56.3 ± 0.17	99.26 ± 0.02	67.7 ± 0.17	99.40 ± 0.03
ViT PE	Complete	6.22 ± 0.13	89.49 ± 0.21	11.20 ± 0.10	96.10 ± 0.08
	Row	31.32 ± 0.20	98.21 ± 0.11	43.44 ± 0.19	98.95 ± 0.06
	Column	15.14 ± 0.12	97.33 ± 0.12	42.51 ± 0.20	99.08 ± 0.05

token order, its own positional encoding, and possibly some weak residual signals from the backbone. For object classification, however, token order appears irrelevant, suggesting that the model relies on object-level signals rather than spatially structured semantics.

When tokens are shuffled in a structured way, either row-wise or column-wise, performance degrades less severely than under full shuffling but bounding boxes show systematic distortions, stretching along the shuffled axis (compare Fig. 7 last two rows). Since the LLM is not dependent on the backbone's positional encoding, this supports the *containerization* hypothesis (Sec. 3.1): tokens spread along one axis are grouped into a "container", which the model interprets as the object's spatial extent.

Backbone Position Encoding Shuffling Next, we examine the extent to which ViT's own positional encodings contribute to localization capabilities. A complete shuffle of the PE causes a severe degradation in localization performance (Tab. 2). However, this perturbation also destroys the ViT's internal spatial priors, confounding interpretation. To confirm this, we measure zero-shot ImageNet accuracy using the CLIP text encoder: shuffling the PE reduces classification accuracy from 63.29% to 4.45%, showing that the ViT itself fails to provide usable features under such a manipulation.

To better isolate the effect of spatial rearrangements, we performed structured shuffles of the PE, permuting entire rows or columns. In LLaVA-7B, column shuffling disrupts detection accuracy to a larger extent than row shuffling. We hypothesize that the LLM learns a row-major 1D ordering of image patches, and therefore depends on accurate "line-break" information. Column shuffling corrupts this alignment more strongly, whereas row shuffling preserves it. For LLaVA-13B, the difference between row and column shuffling is much smaller, suggesting that larger models are more robust and can reconstruct positional structure without fully relying on the ViT's PE. Qualitative examples supporting these observations are provided in the appendix (Fig. 7).

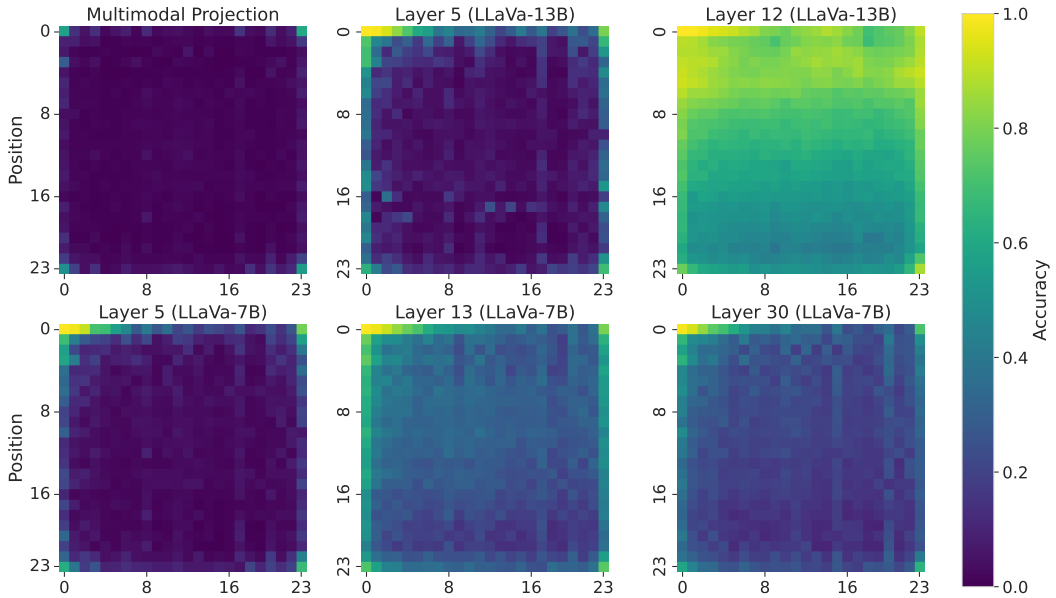

Figure 5: **Heatmap visualizations of positional decoding accuracy at selected stages of LLaVA.** Each heatmap shows the probability of correctly predicting the position of a visual token in the 24×24 grid. The multimodal projection retains positional information mainly at the four corners, effectively marking the image boundaries needed to infer its dimensions. Accuracy then increases within the LLM, becoming highest in the mid-layers (layer 13 in LLaVA-7B and layer 12 in LLaVA-13B). Corners and boundary regions remain the most reliably recovered, indicating that they serve as anchors for reconstructing the global spatial layout.

C Attention Blocking

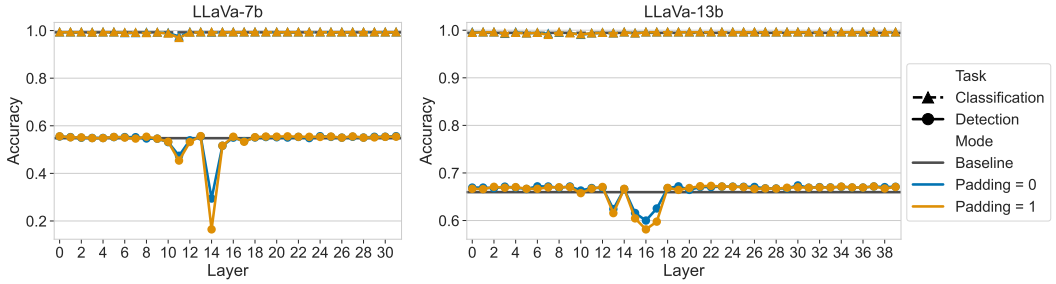

Figure 6: **Attention blocking per layer.** We report classification and detection accuracy when blocking all attention heads of a single layer from tokens following the image tokens to the object tokens. Detection performance drops sharply in the early to mid layers, while classification remains relatively stable throughout the network. Removing attention to padding tokens further decreases localization performance.

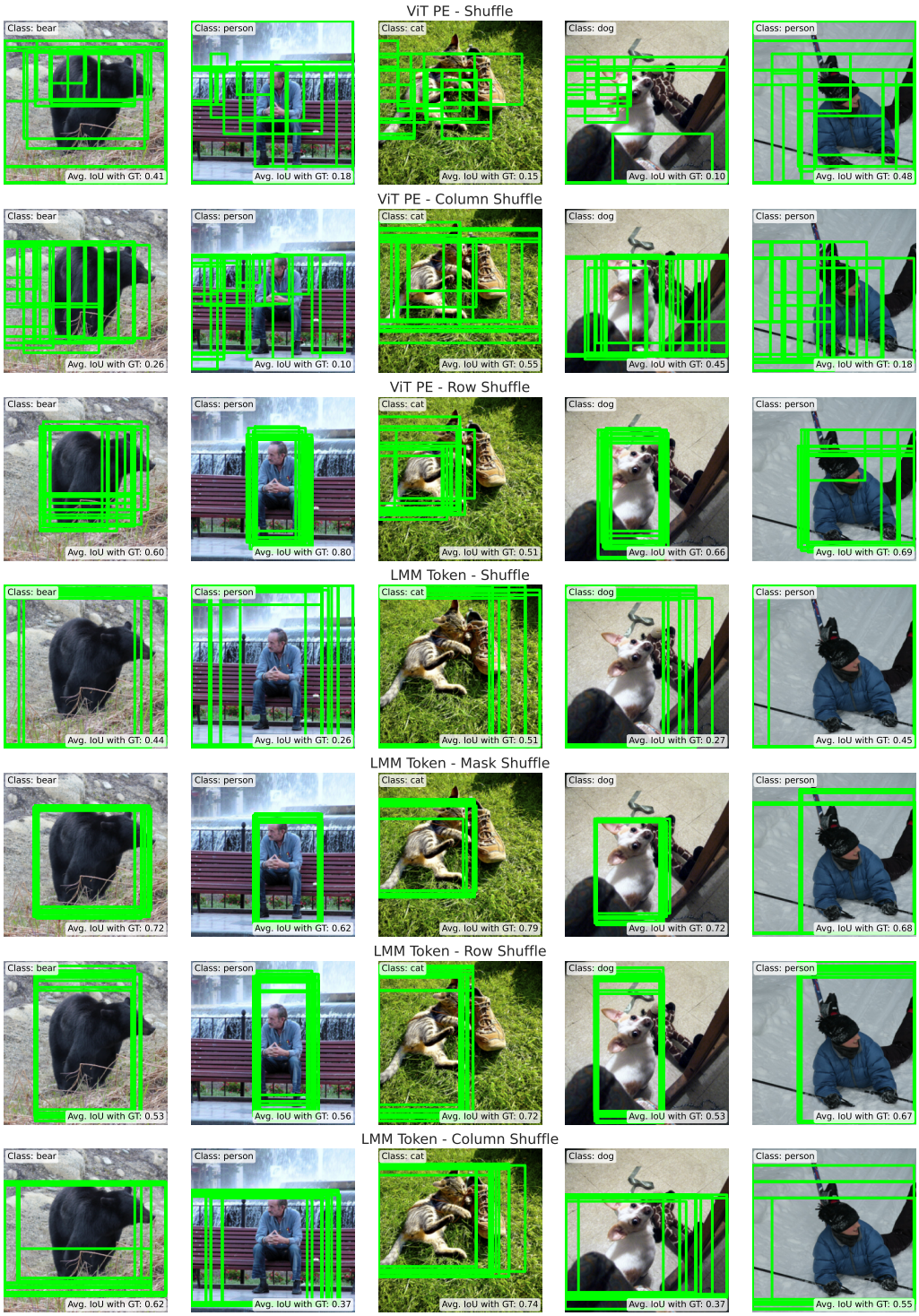

Figure 7: **Qualitative results of the shuffling experiments from Appendix B.** Each column shows the same image with 20 bounding boxes sampled from different seeds after applying a specific shuffling strategy. We prompt LLaVA-7B to detect the main visible object and report the average IoU with the ground-truth annotation. Permuting the positional encoding of the visual backbone globally or column-wise (rows 1 and 2) disrupts detection, whereas row-wise shuffling (row 3) preserves it. Shuffling the LLM's visual input tokens (row 4) causes clear model failure, while restricting shuffling to tokens inside the object mask (row 5) still produces accurate predictions. When shuffling the LLM's visual tokens row- or column-wise (last two rows), detections become stretched along the respective axis, supporting the containerization hypothesis (Sec. 3.1).