Enhancing Part-Level Point Grounding for Any Open-Source MLLMs

Supplementary Material

1. Overview

This supplementary material provides additional details and results that complement the main manuscript. In Sec. 2, we describe the architectural details of the proposed Attention-to-Point (A2P) Decoder. In Sec. 3, we present plots and visualizations of the proposed SDF mapping function and the resulting penalty fields to further illustrate the intuition behind our design. In Sec. 4, we provide statistics and preprocessing steps for the three datasets used in our experiments. In Sec. 5, we outline implementation details. Sec. 6 and Sec. 7 present additional quantitative and qualitative results, respectively. More ablation studies can be found in Sec. 8. Besides, we provide information about the latency and model size in Sec. 9. Finally, we discuss the limitations of our method and potential future directions in Sec. 10.

2. Attention-to-Point (A2P) Decoder

Figure 1 presents the architecture of the proposed Attention-to-Point (A2P) Decoder. Given k synthesized attention maps and image V features from k Q-Synth Modules and LLM localization heads, the A2P Decoder aims to fuse these signals and produce a single high-resolution, point-centric heatmap for 2D point prediction. Specifically, each attention map first modulates its corresponding image-value feature to obtain a spatially weighted feature map $F_i \in \mathbb{R}^{P_h \times P_w \times d_h}$ for $i \in 1, \ldots, k$. An MLP is then used to learn per-head importance weights over the set F_i, and the weighted features are concatenated along the channel dimension to form $F_c \in \mathbb{R}^{P_h \times P_w \times k d_h}$. Consquently, a 1×1 convolution is applied to fuse information across the k heads, producing $F_{\text{fused}} \in \mathbb{R}^{P_h \times P_w \times d_{\text{fused}}}$. In our experiments, we use $k = 5$, $d_h = 128$, and $d_{\text{fused}} = 256$. Next, spatial mixing is performed using four convolutional blocks with skip connections, yielding F_{mixed}. Each block consists of a 3×3 convolution, a GroupNorm layer, and a SiLU activation. To obtain a high-resolution output, we apply two stages of $2\times$ bilinear interpolation, each followed by a convolutional block, resulting in the upsampled feature $F_{\text{up}} \in \mathbb{R}^{4P_h \times 4P_w \times \frac{d_{\text{fused}}}{4}}$. Finally, a 1×1 convolution compresses the channels into a single output map, generating the final heatmap H, which has a spatial resolution four times larger than that of the original attention maps.

3. SDF-based Penalty Field

To supervise the prediction of point-centric heatmaps, we introduce an SDF-based penalty field tailored for our point grounding task. Specifically, each ground-truth mask is first converted into a Signed Distance Field (SDF). We then design a mapping function f that transforms the raw SDF into a penalty field with properties better aligned to point-level supervision. As described in the main manuscript, the mapping function is defined as

$$f(x) \;=\; \text{softplus}\Big(\frac{x}{\tau}\Big) \;+\; \gamma \begin{cases} e^{x/\tau}, & x \le 0, \\ 1, & x > 0\,, \end{cases} \tag{1}$$

where x denotes a scalar SDF value, and τ and γ are hyperparameters controlling the steepness and asymmetry of the penalty inside and outside the mask region.

To visualize the effect of different hyperparameter choices, Figure 2 shows plots of the mapping function under varying (τ, γ) values. In our experiments, we use $\tau = 1.5$ and $\gamma = 1$, which we find to provide the most stable training behavior. Additionally, Figure 3 presents several examples comparing the original binary ground-truth masks with their corresponding transformed penalty fields.

4. Dataset Statistics

For the **PACO [8] dataset**, we use 132,442 part-level training samples and 28,318 testing samples encompassing 422 object parts in total, after applying several preprocessing steps. First, in the original dataset, different object-part instances within the same image are treated as separate annotations; we merge these annotations into a single combined mask per object part for that image. For images containing multiple instances of the same object part, we consider a point prediction correct if it falls within any of the corresponding masks. This evaluation protocol is applied consistently across all baseline methods and our method. Next, we filter out samples whose ground-truth segmentation masks occupy less than 0.1% of the image area. Although these small regions are correctly annotated, they typically correspond to extremely tiny parts that are not meaningful for point grounding, either due to heavy occlusion, distant camera viewpoints, or the inherent small size of certain object-part categories. Examples of such filtered cases are shown in Figure 4. For the **InstructPart [11] dataset**, we use 1,800 training samples and 600 testing samples, and directly adopt the original instruction–mask pairs for our experiments. For the **PointArena Point-Bench [2]**, we evaluate on four tasks—Affordance, Spatial, Reasoning, and Steerability—which contain 198, 195, 193, and 200 testing samples, respectively. Due to GPU memory constraints, we downsample the large input images so that their height and width do not exceed 1,500 pixels. All baseline methods

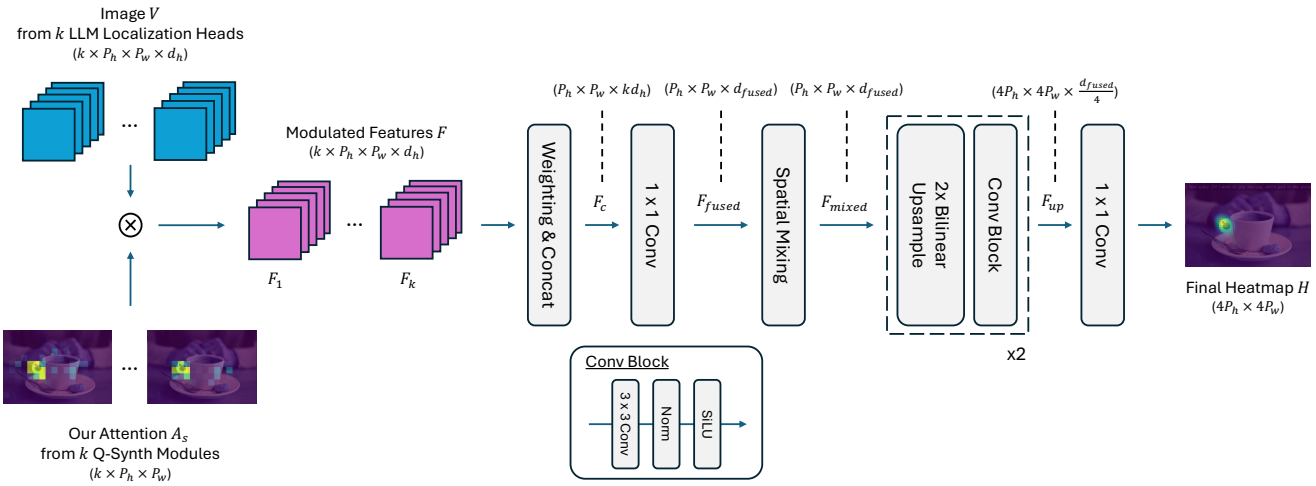

Figure 1. Architecture of the proposed Attention-to-Point (A2P) Decoder. The A2P Decoder fuses k attention maps with image V features and upsamples them into a high-resolution, point-centric heatmap for 2D point grounding. The Spatial Mixing consists of four Conv Blocks with skip connections.

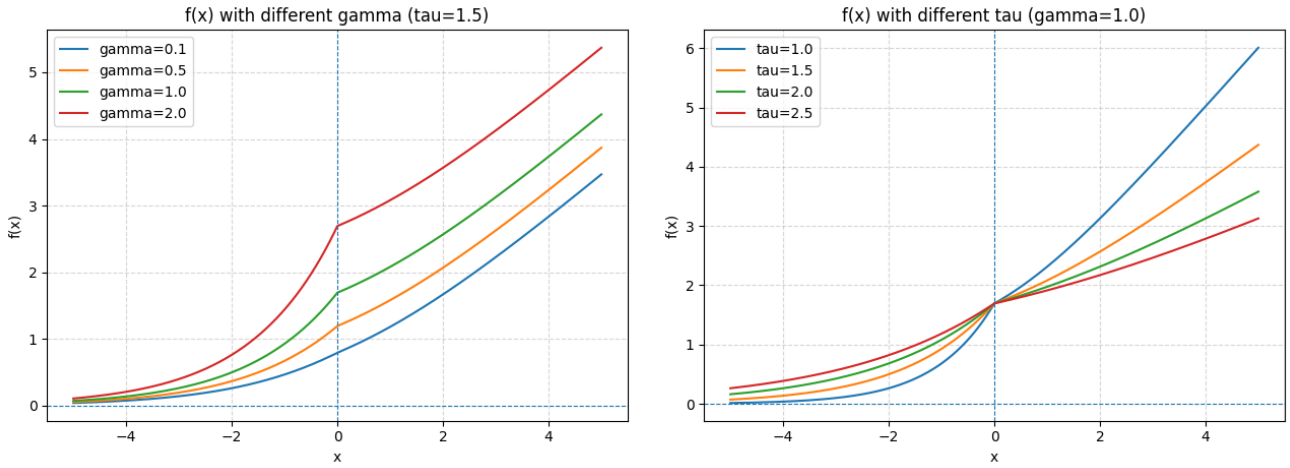

Figure 2. The proposed SDF mapping function f under different (τ, γ) settings. Here, x denotes the original scalar SDF values. In the left plot, γ controls the asymmetry of the mapped values inside versus outside the mask (i.e., for $x < 0$ vs. $x > 0$). In the right plot, varying τ adjusts the overall steepness of the penalty field. The intuition behind this design is that predictions outside the target region should incur large penalties ($x > 0$), while predictions inside the mask should receive much smaller penalties ($x < 0$). At the same time, the function encourages higher confidence near the innermost point of the target by assigning progressively smaller penalties as x decreases. We use $\tau = 1.5$ and $\gamma = 1$ in our experiments.

and our approach are evaluated under the same resolution constraints, ensuring a fair comparison.

5. Implementation Details

We provide the additional hyperparameters used in our experiments as follows. For the Query Synthesis (Q-Synth) Module, we initialize $N = 4$ learnable latent queries and set the number of cross-attention layers to $T = 3$. For the total training loss, we set the weight of the SDF-based loss to $\lambda = 0.001$ to balance its scale relative to the BCE loss. We use AdamW as the optimizer with a learning rate of 1×10^{-4} and a weight decay of 0.01, employ a batch size of 4, and train the model for 30 epochs. All experiments can be run on a single NVIDIA A6000 GPU with 48 GB of memory; in practice, we use four A6000 GPUs for data parallelism.

In addition, the First-Gen-MLLM used in our experiments is LLaVA-1.5-7B [7], although in principle any MLLM that does not natively possess pointing ability can be used. The prompts employed for each MLLM are shown in Figure 5. Note that the prompts differ slightly across

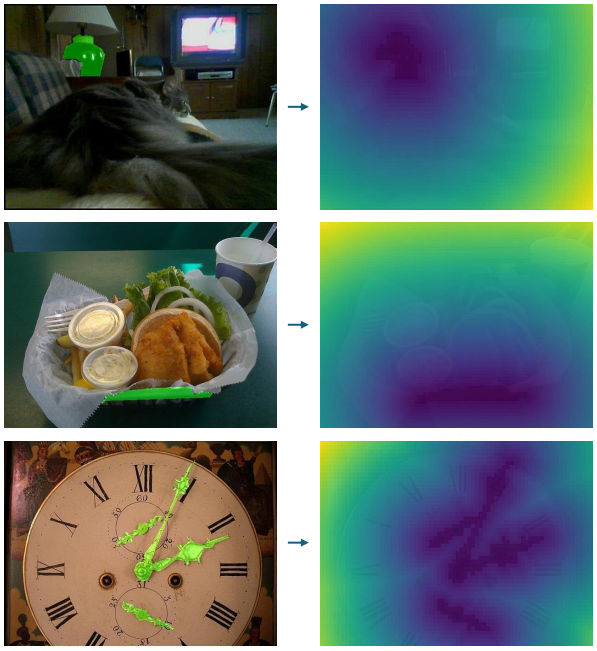

Figure 3. Examples of the original ground-truth masks and the corresponding SDF-based penalty fields. Cooler colors indicate smaller values (i.e., lower penalties). Notably, the third row illustrates a case with multiple object-part instances, where the proposed mapping function still produces a meaningful and well-structured penalty field for supervision.

datasets due to the nature of the tasks, namely direct pointing versus reasoning pointing.

6. More Baseline Results

We provide another type of segmentation-based baseline, Fine-tuned MLLM-based reasoning segmentation, in Table 1. Specifically, we include LISA [5] and GLaMM [9], two representative reasoning-segmentation models. By fine-tuning MLLMs to output a special token `[SEG]` and decode segmentation masks from it, these methods can handle more complex textual instructions that require reasoning. However, such fine-tuning often degrades the original MLLMs' reasoning or VQA capabilities, as it forces the model to generate special-purpose tokens that deviate from natural conversational outputs. In contrast, our method does not modify any MLLM parameters and therefore fully preserves the model's original capabilities.

In addition, we specify the exact baseline model versions used in our experiments. For VLPart [10], we use the Swin-Base Cascade Mask R-CNN architecture pretrained specifically on the PACO [8] dataset. For X-Decoder [14], we adopt the inference script for OpenVocab Referring Segmentation with the Focal-L backbone. For LISA [5], we use the LISA-7B-v1 model, and for GLaMM [9], we employ the GLaMM FullScope model. Notably, among these baselines, VLPart [10] and LISA [5] directly incorporate PACO data during training. In contrast, X-Decoder [14] and GLaMM [9] do not use PACO annotations, but they do train on COCO [6], which contains many of the same underlying images as PACO.

7. More Qualitative Results

We provide additional qualitative results of the PACO [8] and InstructPart [11] datasets in Figure 6. Notably, some examples contain multiple target instances within the same image, and our attention maps successfully highlight all corresponding regions. Although our current framework performs single-point prediction by selecting the most prominent attention peak, these qualitative observations indicate that extending the method to multi-point outputs is feasible, as further discussed in Sec. 10.

8. More Ablation Studies

We follow the setting described in the main manuscript where all the ablation studies are conducted with the First-Gen-MLLM (LLaVA-1.5-7B) on the PACO dataset.

Variants of the Query Synthesis Module. We compare several strategies for utilizing the MLLM's text features in Table 2. Instead of using our Query Synthesis (Q-Synth) Module to extract a single query from the full set of text features, one can alternatively apply average pooling or max pooling, or use the feature of the `[EOS]` token as the text representation. The results show that our proposed Q-Synth Module significantly outperforms these alternatives, demonstrating the effectiveness of our design.

Variants of the SDF-based Supervision. We compare different loss designs for training the Attention-to-Point (A2P) Decoder in Table 3. Instead of supervising the A2P Decoder with the proposed SDF-based Penalty Field, we consider three alternatives: (1) directly using the original binary mask with an L2 loss; (2) using a Gaussian heatmap centered at the innermost point of the mask as supervision; and (3) transforming the mask with the original SDF, without our asymmetric design. The results show that our SDF-based Penalty Field encourages the decoder to produce the most accurate predictions via the point-centric design.

9. Latency and Model Size

For the inference time reported in Table 4, we evaluate LLaVA-1.5-7B and Qwen2.5-VL-7B on the same 100 samples using a single NVIDIA A6000 GPU. Counterintuitively, our method actually has *lower* latency than text pointing (autoregressive text generation), since it requires

Figure 4. Examples of filtered samples from the PACO [8] dataset due to extremely small part areas. These cases correspond to parts that appear tiny because of heavy occlusion, distant camera viewpoints, or the inherent characteristics of certain categories. Red circles highlight the regions of interest, and the green masks within them indicate the tiny ground-truth annotations.

For Direct Pointing (PACO Dataset)

Molmo: f"Point to the {part} of the {object}."

Qwen2.5-VL: f"Point to the {part} of the {object} in the image, output its coordinates in XML format <points x y>object</points>."

LLaVA-1.5: f"Please provide a single 2D point pointing at the {part} of the {object} in the image. This 2D point is represented as a (x, y) tuple, where x is the horizontal image axis, y is the vertical image axis, and the origin is at the top-left corner. The range of these $x - y$ coordinates is normalized as integers between 0 and 1. Follow this output format: Point: (x, y)."

For Reasoning Pointing (InstructPart Dataset & PointArena Point-Bench)

Molmo: f"{Long_instruction}. Please provide a single 2D point pointing at it."

Qwen2.5-VL: f"{Long_instruction}. Please point to it and output its coordinates in XML format <points x y>object</points>."

LLaVA-1.5: f"{Long_instruction}. Please provide a single 2D point pointing at it in the image. This 2D point is represented as a (x, y) tuple, where x is the horizontal image axis, y is the vertical image axis, and the origin is at the top-left corner. The range of these $x - y$ coordinates is normalized as integers between 0 and 1. Follow this output format: Point: (x, y)."

Figure 5. MLLM text-pointing prompts. The Long_instruction used in the reasoning pointing task includes queries that require contextual understanding. For example: "If I want to pick up the knife, which part in the picture can be used?"

Table 1. More baseline results on the PACO [8] (direct pointing) and InstructPart [11] (reasoning pointing) datasets. Our method consistently outperforms all baselines across both point grounding tasks. Notably, even for the MLLM without any point-grounding ability (First-Gen-MLLM), our approach effectively equips it with this capability and yields significant performance gains.

	PACO [8]		InstructPart [11]	
	Patch Accuracy	Accuracy	Patch Accuracy	Accuracy
Segmentation-based Models				
VLPart [10]	0.419	0.381	0.008	0.008
X-Decoder [14]	0.031	0.025	0.185	0.178
LISA-7B [5]	0.345	0.290	0.480	0.464
GLaMM-7B [9]	0.540	0.456	0.412	0.400
MLLM w/ pointing ability				
Molmo-7B text pointing [3]	0.559	0.487	0.737	0.710
Molmo-7B attention pointing [4]	0.517	0.428	0.468	0.378
Molmo-7B **Ours**	**0.603**	**0.510**	**0.900**	**0.868**
Qwen2.5-VL-7B text pointing [1]	0.491	0.407	0.722	0.708
Qwen2.5-VL-7B attention pointing [4]	0.424	0.309	0.352	0.283
Qwen2.5-VL-7B **Ours**	**0.610**	**0.479**	**0.877**	**0.818**
MLLM w/o pointing ability				
First-Gen-MLLM text pointing	0.085	0.068	0.040	0.033
First-Gen-MLLM attention pointing [4]	0.230	0.183	0.227	0.194
First-Gen-MLLM **Ours**	**0.544**	**0.463**	**0.803**	**0.783**

Table 2. Variants of the Query Synthesis Module.

	Avg. Pool	Max Pool	EOS	Ours
Acc.	0.305	0.266	0.290	**0.463**

Table 3. Variants of the SDF-based Supervision.

	L2	Gaussian	Symmetric SDF	Ours
Acc.	0.433	0.436	0.454	**0.463**

Table 4. Latency analysis. Our method actually has lower latency than autoregressive text generation.

	Text pointing	Ours
LLaVA-1.5-7B	588.22 ms	**118.59** ms
Qwen2.5-VL-7B	1351.19 ms	**407.78** ms

only *a single forward pass* through the MLLM to extract the internal QKV features for point prediction. In terms of model size, Q-Synth and the A2P Decoder contain 2.3M and 1.7M parameters, respectively, representing a negligible overhead relative to the 7B-parameter MLLM.

10. Limitations and Future Works

In the main manuscript, we emphasize that our work targets robotic applications in which the model provides a single point per execution step as a high-level signal. A natural extension is enabling the model to output multiple points corresponding to multiple targets within a scene—a capability not currently supported. However, our framework shows promising potential in this direction: the attention patterns produced by the Q-Synth Modules often highlight multiple relevant regions (as shown in Figure 6), and the SDF-based penalty field readily generalizes to multiple masks (Figure 3). Incorporating heuristic or learned mechanisms for detecting and separating attention peaks could enable multi-point prediction in future work.

Another limitation is that our method does not explicitly handle scenarios in which no target is present in the image. This issue could be addressed by applying a confidence-based threshold to reject samples with uniformly low attention responses.

Finally, our enhanced attention patterns may also benefit Visual Question Answering (VQA) pipelines, as recent studies [12, 13] indicate that cropping the image according to MLLM attention improves reasoning via visual chain-of-thought. Exploring how our approach can further strengthen such VQA systems presents an interesting direction for future research.

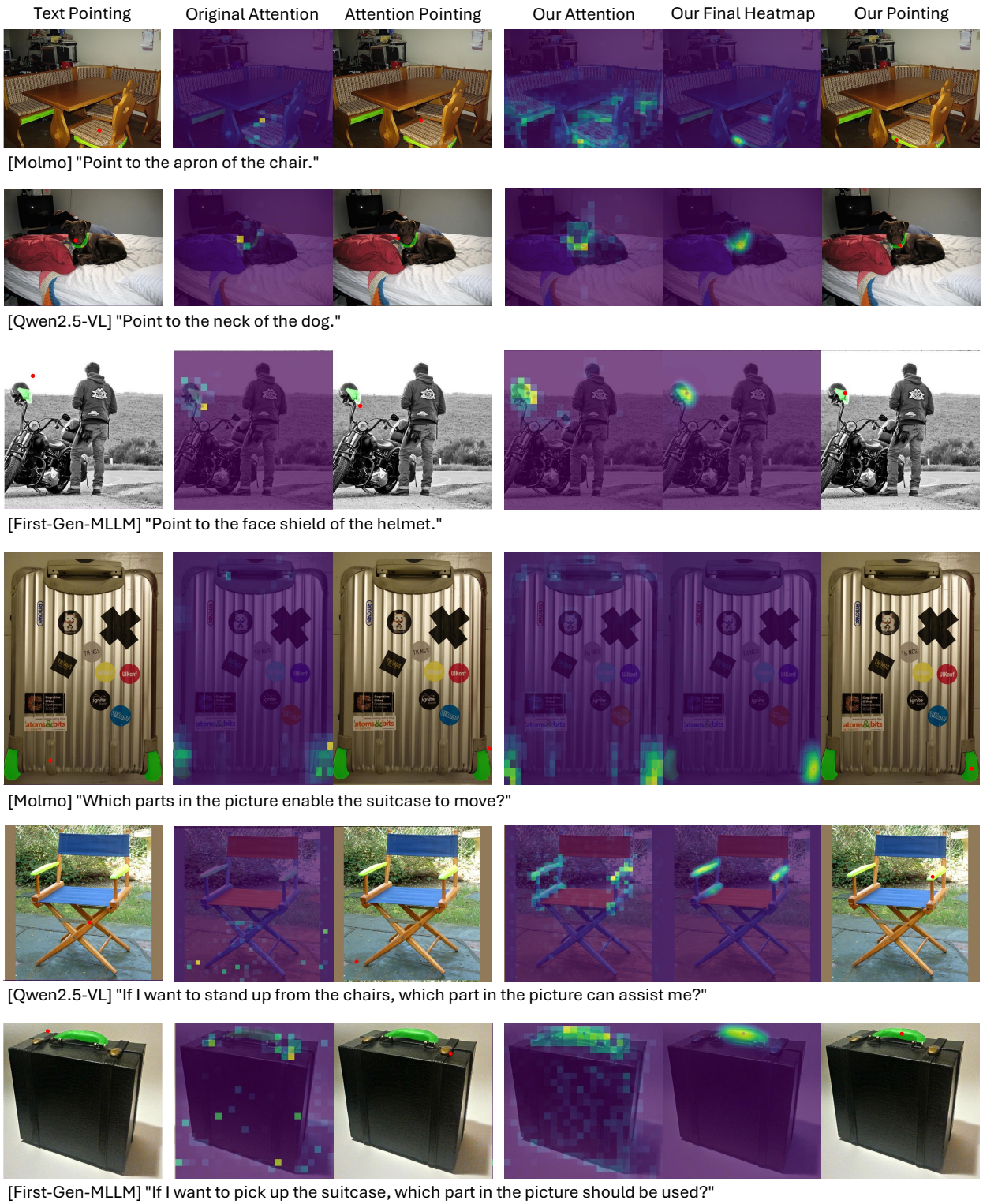

Figure 6. More qualitative results. We compare text pointing, attention pointing, and our proposed method across columns. Each row presents examples from different MLLMs, with the model and text instruction shown below the images. The first three rows correspond to the PACO [8] dataset, while the last three rows correspond to the InstructPart [11] dataset. In the visualizations, green masks denote ground-truth regions, and red points indicate the predicted point grounding locations.

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. 5

[2] Long Cheng, Jiafei Duan, Yi Ru Wang, Haoquan Fang, Boyang Li, Yushan Huang, Elvis Wang, Ainaz Eftekhar, Jason Lee, Wentao Yuan, Rose Hendrix, Noah A. Smith, Fei Xia, Dieter Fox, and Ranjay Krishna. Pointarena: Probing multimodal grounding through language-guided pointing, 2025. 1

[3] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca 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. 5

[4] Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. Your large vision-language model only needs a few attention heads for visual grounding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9339–9350, 2025. 5

[5] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9579–9589, 2024. 3, 5

[6] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. 3

[7] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2024. 2

[8] Vignesh Ramanathan, Anmol Kalia, Vladan Petrovic, Yi Wen, Baixue Zheng, Baishan Guo, Rui Wang, Aaron Marquez, Rama Kovvuri, Abhishek Kadian, et al. Paco: Parts and attributes of common objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7141–7151, 2023. 1, 3, 4, 5, 6

[9] Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M Anwer, Eric Xing, Ming-Hsuan Yang, and Fahad S Khan. Glamm: Pixel grounding large multimodal model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13009–13018, 2024. 3, 5

[10] Peize Sun, Shoufa Chen, Chenchen Zhu, Fanyi Xiao, Ping Luo, Saining Xie, and Zhicheng Yan. Going denser with open-vocabulary part segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15453–15465, 2023. 3, 5

[11] Zifu Wan, Yaqi Xie, Ce Zhang, Zhiqiu Lin, Zihan Wang, Simon Stepputtis, Deva Ramanan, and Katia Sycara. Instructpart: Task-oriented part segmentation with instruction reasoning. In *The 63rd Annual Meeting of the Association for Computational Linguistics*, 2025. 1, 3, 5, 6

[12] Size Wu, Sheng Jin, Wenwei Zhang, Lumin Xu, Wentao Liu, Wei Li, and Chen Change Loy. F-lmm: Grounding frozen large multimodal models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24710–24721, 2025. 5

[13] Jiarui Zhang, Mahyar Khayatkhoei, Prateek Chhikara, and Filip Ilievski. MLLMs know where to look: Training-free perception of small visual details with multimodal LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025. 5

[14] Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, et al. Generalized decoding for pixel, image, and language. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15116–15127, 2023. 3, 5