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

Jin-Cheng Jhang[1] Fu-En Wang[2] Xin Yang[2] Nan Qiao[2] Lu Xia[2]
Min Sun[1,2] Cheng-Hao Kuo[2]
[1]National Tsing Hua University [2]Amazon
frank890725@gapp.nthu.edu.tw sunmin@ee.nthu.edu.tw
{fuenwang, yngxin, qiaonan, luxial, chkuo}@amazon.com

Abstract

Visual grounding aims to associate free-form textual queries with specific regions in an image. While recent Multimodal Large Language Models (MLLMs) have demonstrated promising capabilities in this domain, they primarily excel at object-level grounding and often struggle with part-level grounding—an essential requirement for fine-grained tasks such as robotic manipulation. In this work, we introduce a general approach that equips any open-source MLLMs with accurate 2D part-level point grounding, offering a more direct alternative to conventional grounding representations. Our method leverages the attention mechanisms inherently present in MLLMs. By synthesizing text-conditioned, grounding-aware queries within intermediate layers via the proposed Q-Synth Module, we capture target-relevant attention patterns and refine them with a lightweight Attention-to-Point Decoder, which converts these patterns into a point-centric heatmap for final prediction. Notably, all original MLLM parameters are frozen, ensuring full preservation of their pre-trained capabilities. Experiments show that our design consistently improves part-level grounding accuracy across datasets and can be seamlessly integrated into any open-source MLLMs.

1. Introduction

Point grounding aims to associate a textual description with a specific point in an image, providing a direct cue that links semantic concepts to precise spatial locations. Beyond vision-language understanding, this point-based formulation may also serve as a useful perception primitive for downstream robotic interaction [7, 8, 12, 13, 16, 28].

For instance, in a garment-folding scenario, localizing a point on parts such as the cuff, collar, or hem at each step may provide informative cues for where the garment could be grasped or placed. More broadly, this example highlights the limitation of object-level grounding for physical interaction tasks, where fine-grained manipulation often depends on localizing a specific object part rather than the object as a whole. This makes *part-level* grounding—such as identifying the handle of the knife shown in Figure 1—particularly relevant. Moreover, compared with conventional grounding formats such as bounding boxes or segmentation masks, point-based grounding provides a more direct and compact representation that is naturally aligned with action-relevant locations.

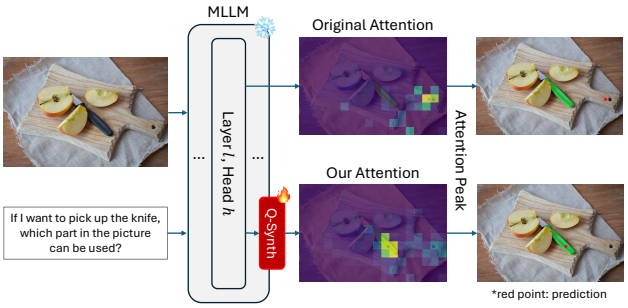

Figure 1. Overview of the core idea. We aim to synthesize grounding-aware queries from a frozen MLLM via the proposed Query Synthesis (Q-Synth) Module to extract more accurate attention patterns for part-level point grounding. Warmer colors (yellow) in the attention maps indicate higher attention values. The red points denote the predicted grounding locations, and the green masks show the corresponding ground-truth regions.

In recent years, Multimodal Large Language Models (MLLMs) have demonstrated remarkable reasoning abilities, enabling them to decompose long-horizon robotic tasks into manageable subtasks [6, 11, 25, 26]. If such capable MLLMs could also acquire accurate visual grounding abilities, they would not only understand *what* steps to take but also *where* to execute them based on their visual observations. Indeed, some recent MLLMs [2, 5, 21] have recognized the importance of point-level grounding and explicitly included it in their learning objectives. However, as shown in Table 1, existing models perform well on object-level grounding but still struggle with part-level targets.

Table 1. Comparison of object-level and part-level point grounding on the PACO [17] dataset. The models generate point coordinates in text form, referred to as text pointing in later experiments. Acc. denotes accuracy, measured by whether the predicted point lies within the ground-truth mask. First-Gen-MLLM refers to a model (<10B parameters) not trained for point grounding. The results show that while existing models perform well on object-level grounding, they still struggle with precise part-level localization.

	Obj-level Acc.	Part-level Acc.
Molmo-7B [5]	0.854	0.487
Qwen2.5-VL-7B [2]	0.838	0.407
First-Gen-MLLM	0.155	0.068

Therefore, how to enhance the existing grounding ability of MLLMs has become a popular and crucial research direction. Several prior works [10, 15, 18, 19, 23] have attempted to equip MLLMs with segmentation capabilities by fine-tuning them to output special tokens (e.g., `[SEG]`) and then decoding the segmentation masks from these tokens. However, such fine-tuning often causes the models to overfit to the special token outputs, degrading their original reasoning and natural conversational abilities, as observed in [24]. To overcome this limitation, recent studies [9, 24] have discovered an emerging property within the attention layers of MLLMs: certain text-to-image attention maps inherently capture strong and spatially consistent signals that reveal what the model is focusing on. These attention patterns can thus be leveraged to infer grounding results without modifying the original language modeling head.

Although directly using the native attention maps offers a feasible way to obtain grounding results from MLLMs, it remains suboptimal in accuracy—particularly for part-level grounding, which demands highly precise spatial localization. Building on prior findings, we propose a learnable approach that refines native attention patterns for more accurate grounding, as illustrated in Figure 1. Specifically, we introduce a **Query Synthesis (Q-Synth) Module** that takes a sequence of text-query features and iteratively aggregates salient semantics to synthesize a single grounding-aware query, which drives a targeted text-to-image attention map. To further achieve higher spatial resolution and finer localization, we design an **Attention-to-Point (A2P) Decoder** that performs progressive upsampling and spatial refinement to generate the final heatmap for precise point grounding. In addition, we introduce an **SDF-based penalty field** that provides point-centric supervision and guides the model to form sharper, more spatially coherent heatmaps. Experiments demonstrate that our method can be seamlessly integrated into various open-source MLLMs and consistently enhances their point grounding performance across different benchmarks. Remarkably, these improvements are achieved without compromising the models' original reasoning and generalization abilities.

To sum up, our main contributions are as follows:

- We identify part-level point grounding as an important perception primitive and present a general framework that equips any open-source MLLMs with this ability while preserving their pre-trained capabilities.
- We design a lightweight attention refinement module that integrates query synthesis, attention decoding, and an SDF-based penalty field to produce grounding-aware attention patterns for precise point localization.
- Experiments across multiple MLLMs show that our method integrates seamlessly with existing architectures and consistently improves grounding performance without additional fine-tuning.

2. Related Works

2.1. Point-based Grounding

Point grounding represents visual targets as spatial points conditioned on language instructions. Recent works have demonstrated the effectiveness of point grounding in real-world scenarios. Rekep [8] and MOKA [7] generate multiple actionable keypoint candidates and query an MLLM (e.g., GPT-4o [1]) via visual prompting to identify relationships or attributes among them, thereby guiding the robot end-effector trajectories for various manipulation tasks. Similarly, SuctionPrompt [13] provides suction-point candidates to an MLLM, which identifies the optimal grasping location and achieves reliable performance in convenience store scenarios. Other than *selecting points* from candidates, RoboPoint [28] fine-tunes MLLMs directly to become pointing specialists, demonstrating the versatility of point grounding across tasks such as manipulation, augmented reality, and navigation.

Meanwhile, several recent MLLMs [2, 5, 21] have begun incorporating point grounding into their training objectives, allowing native text-based pointing outputs. However, these models typically perform well at object-level grounding but struggle with part-level localization, which demands finer spatial precision. Moreover, models lacking explicit point-grounding supervision—referred to as First-Gen-MLLMs—remain unable to generate meaningful pointing results. Therefore, our work aims to enhance the part-level point grounding ability of any MLLMs, enabling more precise and actionable grounding for a broader range of robotic applications.

2.2. Enhance Grounding Ability for MLLMs

Recent studies on enhancing the grounding ability of Multimodal Large Language Models (MLLMs) generally follow two main directions. The first direction focuses on fine-tuning the parameters of MLLMs to explicitly equip them

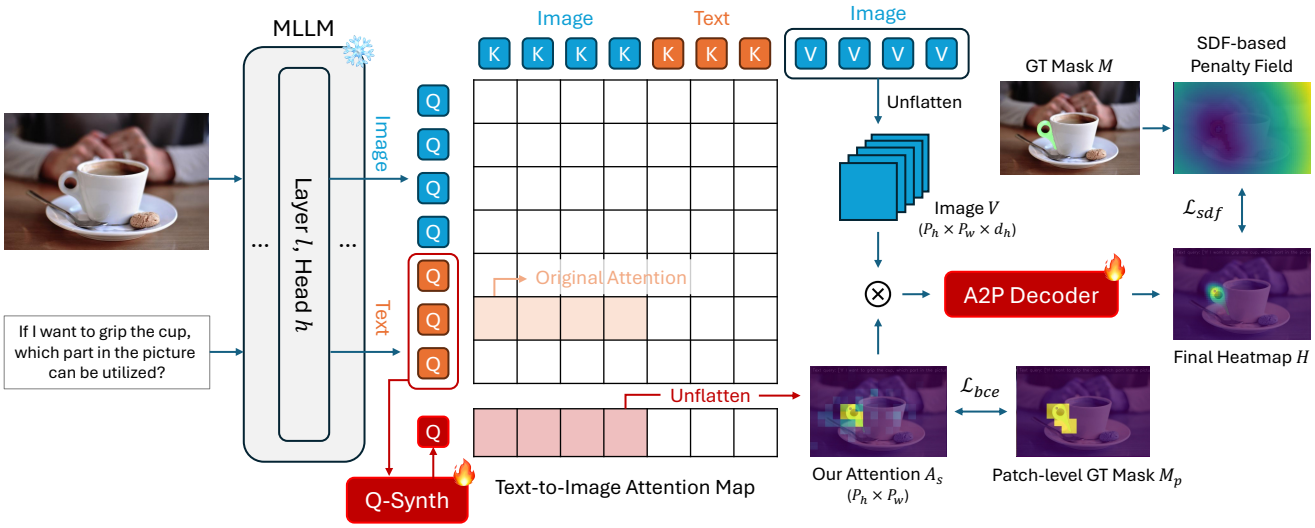

Figure 2. Overview of our framework. Instead of relying on the native text-to-image attention, the proposed Query Synthesis (Q-Synth) Module (Sec. 4.1) conditions on all text features and synthesizes a single grounding-aware query q_s to extract a refined attention map A_s. The Attention-to-Point (A2P) Decoder (Sec. 4.2) then combines A_s with image V to form attention-weighted features and decodes a high-resolution heatmap H for point prediction. For training, we use a per-patch BCE loss to guide A_s and an SDF-based penalty field to shape H (Sec. 4.3). In the penalty field, lower values (shown in cooler colors) indicate the target regions.

with grounding capabilities. Several works [3, 14, 27] train MLLMs to autoregressively output bounding-box coordinates in text form, while others [10, 15, 18, 19, 23] introduce special grounding tokens (e.g., `[SEG]`) and decode boxes or masks from them. Although effective, these methods require extensive training and often degrade the model's native reasoning and conversational abilities.

The second direction explores frozen-parameter approaches, leveraging the inherent attention mechanisms of MLLMs without modifying their pre-trained weights. Recent works [9, 24] reveal an emergent grounding property within the attention patterns of MLLMs, where text-to-image attention maps can naturally highlight the visual regions relevant to a given text query. By extracting and interpreting these attention maps, grounding results can be obtained while fully preserving the model's original multimodal reasoning capabilities. Building upon this observation, we learn to refine and extract more accurate attention patterns, further enhancing grounding performance.

3. Preliminary

MLLM Structure. A Multimodal Large Language Model (MLLM) for image–text input typically consists of a vision encoder, a projector, and a Large Language Model (LLM). Given an image $I \in \mathbb{R}^{H\times W\times 3}$, the vision encoder divides I into a grid of patches and produces visual embeddings $V \in \mathbb{R}^{P_h\times P_w\times d_v}$, where P_h and P_w denote the height and width of the patch grid, and d_v is the visual feature dimension. A projector maps V into the LLM feature space, yielding a sequence of visual tokens $X_v \in \mathbb{R}^{P_h P_w\times d}$, where d is the LLM hidden size. Let $X_t \in \mathbb{R}^{L\times d}$ denote the tokenized text embeddings, where L is the number of text tokens. The multimodal sequence is formed by concatenation, $X = [X_v; X_t] \in \mathbb{R}^{(P_h P_w+L)\times d}$, and is fed into the LLM for autoregressive generation.

Text-to-Image Attention Maps. In MLLMs, text and image features typically interact through attention mechanisms within the LLM. In layer l and attention head h, text features act as queries Q_t, while image features serve as keys K_v and values V_v. For a specific target text query vector q_{tg}, its attention weights a are computed as

$$a^{l,h} = \text{softmax}\left(\frac{q_{\text{tg}} K_v^\top}{\sqrt{d_h}}\right) \in \mathbb{R}^{P_h P_w+L}, \qquad (1)$$

where d_h is the dimension of each attention head, typically d divided by the number of heads. The text-to-image attention component can be obtained by taking the first $P_h P_w$ elements, i.e., $a^{l,h}[:P_h P_w]$, which can be reshaped into a 2D attention map $A^{l,h} \in \mathbb{R}^{P_h\times P_w}$.

Localization Heads. Recent work [9] shows that only a small subset of attention heads within LLM layers is crucial for visual grounding and proposes a method to select these heads. By aggregating the text-to-image attention maps from the selected heads as spatial cues, MLLMs can produce bounding-box or segmentation-mask outputs in a

zero-shot manner. This phenomenon is observed across different MLLMs. Following this finding, we first identify the top k grounding-relevant heads and use them as the basis for our method.

4. Method

Our proposed framework is illustrated in Figure 2. It consists of two main components: the Query Synthesis Module (Sec. 4.1) and the Attention-to-Point Decoder (Sec. 4.2). In addition, we introduce tailored training objectives to jointly optimize both modules in an end-to-end manner (Sec. 4.3).

4.1. Query Synthesis (Q-Synth) Module

In [9], the target text token q_{tg} is always chosen as the last token to represent the entire sentence, following the autoregressive property of the LLM. For example, in the sentence "*Point to the handle of the knife.*", the query vector of the last token [.] is used in their experiments. However, the last token may not always capture the complete semantics of the sentence, especially when it involves multiple concepts or requires further reasoning.

To address this limitation, we propose the Query Synthesis (Q-Synth) Module, as illustrated in Figure 3, which conditions on all text queries $Q_t \in \mathbb{R}^{L\times d_h}$ from LLM layer l and head h to synthesize a single query $q_s \in \mathbb{R}^{1\times d_h}$ for more comprehensive and grounding-aware attention map extraction. The module employs a stack of cross-attention layers that iteratively model the interaction between the text features and a set of learnable latent queries. Specifically, we initialize N learnable latent vectors $Z^{(0)} \in \mathbb{R}^{N\times d_h}$ as the queries, and use the text features as both keys and values, i.e., $K_s = V_s = Q_t$. Through multiple rounds of cross-attention ($t = 1, \ldots, T$), the text features iteratively inject rich semantic information, while the latent queries extract and summarize the most relevant semantics:

$$Z^{(t)} = Z^{(t-1)} + \text{CrossAttn}\Big(Z^{(t-1)},\, K_s,\, V_s\Big), \qquad (2)$$

where $Z^{(t)} \in \mathbb{R}^{N\times d_h}$ denotes the refined latent set at layer t. After T refinement layers, we obtain $Z^{(T)} = [z_1, \ldots, z_N]$. A lightweight multilayer perceptron (MLP) then computes importance weights over these embeddings to synthesize the final query:

$$q_s = \sum_{i=1}^{N} \alpha_i z_i, \quad \alpha_i = \text{softmax}(\text{MLP}(z_i)), \qquad (3)$$

where α_i represents the learned weighting of each latent. The resulting q_s is used to compute our synthesized text-to-image attention map A_s, similar to Eq. 1, serving as a replacement for q_{tg}.

To make q_s grounding-aware and aligned with the visual features, the Q-Synth Module is trained using a binary cross-entropy (BCE) loss, as detailed in Sec. 4.3. Besiedes, since we select the top k attention heads as localization heads, a separate Q-Synth Module is attached to each of them, producing k synthesized attention maps. These maps are designed to highlight the target regions corresponding to the input text description and are subsequently ensembled in Sec. 4.2.

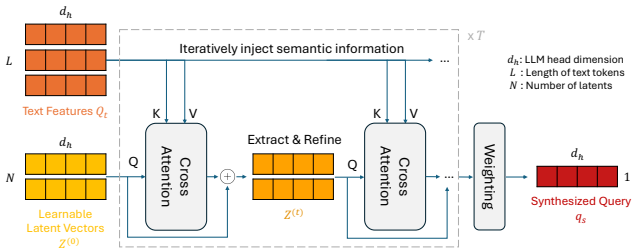

Figure 3. Overview of the Query Synthesis (Q-Synth) Module. Text features Q_t from the LLM and N learnable latent vectors $Z^{(0)}$ are processed by stacked cross-attention layers, where Q_t serves as keys/values to iteratively inject semantics while the latents $Z^{(t)}$ extract and refine the most relevant text information. After T iterations, a lightweight weighting network aggregates the refined latents to produce a single query q_s for grounding-aware attention extraction.

4.2. Attention-to-Point (A2P) Decoder

Due to the patchification of images in the vision encoder, the resolution of the extracted attention maps is relatively low (typically, P_h and P_w are the original image dimensions divided by 14). Even if the correct patch is identified from the attention map, directly pointing to the patch center often fails to yield accurate localization.

To overcome this, we introduce the Attention-to-Point (A2P) Decoder, which produces a higher-resolution, point-focused heatmap for final point prediction. Given the synthesized attention map A_s from the Q-Synth Module, derived from q_s and the image features K_v, we further leverage the original image V_v for decoding, as it contains the actual visual content represented in the LLM attention head. We obtain a spatially modulated feature map $F \in \mathbb{R}^{P_h\times P_w\times d_h}$ by weighting the image V_v with the synthesized attention map A_s, effectively highlighting the target regions while suppressing irrelevant areas.

Since we have k synthesized attention maps from different localization heads, a lightweight MLP learns to weight these k feature maps. The weighted maps are concatenated along the feature dimension and passed through a 1×1 convolution to reduce dimensionality, producing fused feature maps $F_{\text{fused}} \in \mathbb{R}^{P_h\times P_w\times d_{\text{fused}}}$. Finally, a sequence of convolutional layers interleaved with bilinear upsampling operations performs spatial refinement and progressive upscaling, generating the final high-resolution heatmap H for 2D point prediction. More architectural details of the decoder

are provided in the supplementary material.

4.3. Training Objectives

The type of supervision used directly determines the patterns of the synthesized attention map A_s and the final heatmap H, making it a crucial component of our method.

For the Q-Synth Module, we use part-level segmentation masks $M_p \in \mathbb{R}^{P_h \times P_w}$, downsampled from the original segmentation masks $M \in \mathbb{R}^{H \times W}$, as ground truth supervision. We adopt the Binary Cross-Entropy (BCE) loss to guide learning. Intuitively, the goal is to synthesize a query q_s that exhibits the highest similarity to the visual features of the target region in K_v. Thus, we formulate this as a *per-patch classification problem* using BCE loss:

$$\mathcal{L}_{\text{bce}} = \text{BCE}\big(q_s K_v^\top,\, M_p\big)\,, \tag{4}$$

where $q_s K_v^\top$ represents the predicted per-patch similarity scores. In this way, the Q-Synth Module is encouraged not only to extract relevant textual information from the text features but also to refine and align its latent embeddings with the corresponding visual features. Unlike conventional attention maps, our synthesized attention patterns tend to produce near-binary activations around the target regions, which facilitates effective visual feature modulation in Sec. 4.2.

For the A2P Decoder, we aim for the predicted heatmap to not only cover the target region but also concentrate around the most representative point of the object part. To this end, we design a penalty field inspired by the Signed Distance Field (SDF) commonly used in 3D reconstruction. In a standard SDF, boundary pixels have zero values, with positive distances outside the ground-truth mask and negative distances inside. However, this formulation treats both sides of the boundary symmetrically, which is suboptimal for our objective. In our case, predictions located outside the target region should incur large penalties, while those inside the mask are considered correct and should receive much smaller penalties; nevertheless, we still want to encourage the model to focus near the innermost point of the target.

To achieve this, we define a mapping function f that transforms the original SDF into a modified penalty field. Specifically, we apply the softplus function to the original distances and introduce two hyperparameters, τ and γ, to control the *steepness* and *asymmetry* of the field inside and outside the mask:

$$f(x) \;=\; \text{softplus}\Big(\frac{x}{\tau}\Big) \;+\; \gamma \begin{cases} e^{x/\tau}, & x \le 0, \\ 1, & x > 0\,, \end{cases} \tag{5}$$

where x denotes a scalar SDF value. Further details and visualizations of the penalty field are provided in the supplementary material. Finally, our SDF-based loss is computed as a spatial sum of the product between the predicted heatmap distribution and the modified penalty field:

$$\mathcal{L}_{\text{sdf}} = \sum_{u,v} \text{softmax}\big(H(u,v)\big)\; f\big(D(u,v)\big), \tag{6}$$

where H is the predicted heatmap, D is the Signed Distance Field, and (u, v) index pixel positions.

Our total loss is defined as $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{bce}} + \lambda \mathcal{L}_{\text{sdf}}$, and we train the model end-to-end. Notably, all original MLLM parameters are frozen during training, enabling us to enhance the pointing ability while fully preserving the models' pre-trained capabilities.

5. Experiments

5.1. Settings

Datasets. We include three datasets in our experiments: **(1) PACO [17]** is originally designed for part-level segmentation. Each annotation follows the format *<object>:<part>* (e.g., *knife:handle*). For point grounding, we convert the annotations into textual queries such as "*Point to the handle of the knife.*" Since the instruction directly specifies the target part, we refer to this as the *direct pointing* task. **(2) InstructPart [22]** provides instruction–mask pairs at the part level. For example, "*If I want to open the oven, which part should be utilized?*" Such queries often omit the target part and require reasoning, so we term this the *reasoning pointing* task. **(3) PointArena Point-Bench [4]** is a benchmark for evaluating multimodal pointing ability across five tasks: Affordance, Spatial, Reasoning, Steerability, and Counting. Among them, only the Affordance task focuses on part-level pointing, while the others mainly involve object-level grounding. However, we still include these tasks to evaluate the generalizability of our method beyond part-level scenarios. The Counting task, which requires numerical outputs, is excluded as it falls outside our framework's scope. Further dataset details are provided in the supplementary material.

Segmentation-based Baselines. We evaluate two categories of segmentation-based models: **(1) Open-vocabulary part-level segmentation.** We adopt VL-Part [20] as a representative model, since it is trained on the PACO dataset [17] for part-level segmentation. However, it lacks reasoning ability—although it accepts free-form text inputs, it is expected to fail on the reasoning pointing task in the InstructPart dataset [22]. **(2) Open-vocabulary referring segmentation.** We select X-Decoder [29] for this category. Although it is not specifically trained on part-level data, it can process longer referring expressions and is therefore expected to perform better on reasoning pointing tasks. More segmentation-based baselines are included in the supplementary material. For all segmentation-based

Table 2. Main quantitative results on the PACO [17] (direct pointing) and InstructPart [22] (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 [17]		InstructPart [22]	
	Patch Accuracy	Accuracy	Patch Accuracy	Accuracy
Segmentation-based Models				
VLPart [20]	0.419	0.381	0.008	0.008
X-Decoder [29]	0.031	0.025	0.185	0.178
MLLM w/ pointing ability				
Molmo-7B text pointing [5]	0.559	0.487	0.737	0.710
Molmo-7B attention pointing [9]	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 [2]	0.491	0.407	0.722	0.708
Qwen2.5-VL-7B attention pointing [9]	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 [9]	0.230	0.183	0.227	0.194
First-Gen-MLLM **Ours**	**0.544**	**0.463**	**0.803**	**0.783**

models, we extract the innermost point of the largest predicted mask as the final point prediction.

MLLMs. We evaluate two *pointing-capable* MLLMs, Molmo-7B [5] and Qwen2.5-VL-7B [2], which have been trained with point-grounding objectives. We also include one First-Gen-MLLM, defined as a *non-pointing* MLLM under 10B parameters without point-grounding supervision. Notably, Molmo [5] is specialized for pointing and attains state-of-the-art results among open-source models.

We consider three evaluation modes: **(1) Text pointing**: the MLLM outputs the target coordinates in text form through autoregressive generation. **(2) Attention pointing**: we use the native text-to-image attention maps from the selected localization heads [9] and identify the peak-attention patch for pointing. **(3) Ours**: we apply our Q-Synth and A2P modules to each MLLM, keeping the backbone frozen, and evaluate whether our method can consistently improve point grounding across open-source models.

Metrics. Following the PointArena benchmark [4], we evaluate using a hit-or-miss criterion, where a prediction is considered correct if the predicted point lies within the ground-truth mask. Accuracy is then computed as the ratio of hits to total samples. As in the PointArena protocol, when an MLLM outputs multiple points in text form, we take the first predicted point as the final output. This setup naturally aligns with our motivation for robotic interaction, where the model generates a single actionable point at each step to guide the robot's manipulation.

In addition, attention-based methods, including ours, produce attention maps and locate the target by pointing to the center of the peak attention patch. Therefore, we can also compute *patch accuracy*, which reflects the localization precision before patch-level quantization errors. Although our A2P Decoder alleviates this through upsampling, some residual error remains in the final heatmap. For non-attention-based methods, we convert their point predictions back to the patch coordinates to enable fair comparison under the same metric.

5.2. Quantitative Results

We present our main quantitative results in Table 2. Across both the direct pointing task on the PACO dataset [17] and the reasoning pointing task on the InstructPart dataset [22], our proposed method consistently outperforms all baselines, including the state-of-the-art pointing-specialized MLLM Molmo [5].

Notably, compared to the attention-pointing baseline, our approach achieves substantial improvements, especially on the InstructPart dataset [22], where the input texts are generally longer and require reasoning. This indicates that our method effectively extracts relevant textual semantics and refines the native attention patterns to yield more accurate and spatially aligned grounding maps (see Sec. 5.3 for qualitative comparisons). Furthermore, even for a model without any point-grounding capability (First-

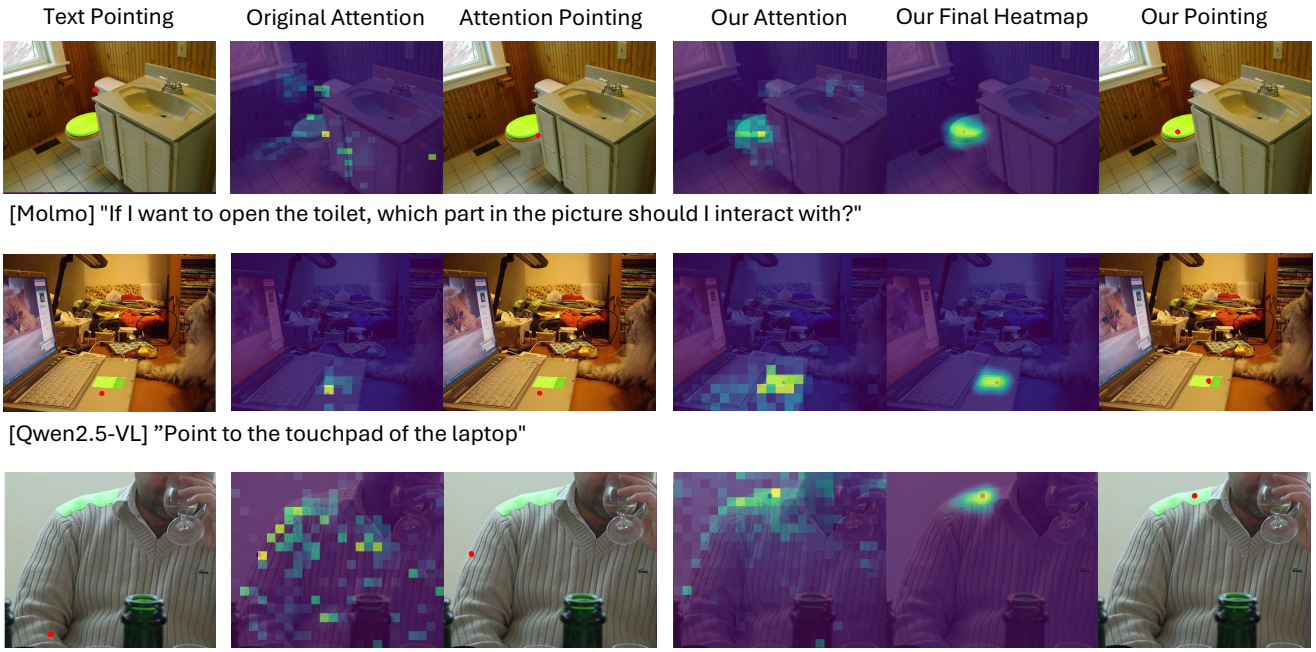

Figure 4. Qualitative results. We compare text pointing, attention pointing, and our proposed method across columns. Each row presents examples from different MLLMs, with the model and text instruction shown below the images. In the visualizations, green masks denote ground-truth regions, and red points indicate the predicted point grounding locations. Notably, the sweater-part example in the third row echoes our motivation in Sec. 1, highlighting the relevance of part-level point grounding for tasks such as cloth folding.

Gen-MLLM), our method can still leverage its internal attention mechanism to produce competitive pointing results comparable to those of pointing-capable models. This demonstrates the generality of our framework: as long as an MLLM incorporates an attention mechanism—which is true for virtually all existing MLLMs—our method can enhance its pointing ability while fully preserving its pre-trained reasoning and language understanding capabilities, since no parameters of the original model are modified.

5.3. Qualitative Results

To better illustrate the improved attention behavior of our method, we present qualitative examples in Figure 4. Our method produces attention maps that more completely cover the target regions, benefiting from the design of the Q-Synth Module and the BCE-based supervision. Moreover, the A2P Decoder further refines these patch-level attention maps and generates high-resolution heatmaps that lead to more accurate point predictions. Additional visualizations are provided in the supplementary material.

5.4. Generalizability

We train on the PACO [17] dataset with *direct pointing* supervision and evaluate on the PointArena benchmark [4] with *reasoning pointing* prompts to assess generalizability of our method. The results are shown in Table 3. Among the various tasks in this benchmark, only the Affordance task is similar to PACO, while the others represent unseen tasks during training. Surprisingly, across all models and tasks, our method consistently improves over the original attention-pointing baseline. This transferability demonstrates that our method learns to summarize the overall semantic intent of the input text and align it with the most relevant visual concepts, enabling the model to better associate language and visual evidence even in unseen scenarios. In other words, the Q-Synth Module learns to condense textual meaning into a grounding-aware query, while the A2P Decoder translates refined attention into spatially coherent heatmaps, yielding more interpretable attention patterns across different grounding tasks and datasets.

5.5. Ablation Study

We conduct ablations on the First-Gen-MLLM using the PACO [17] dataset, as the model lacks inherent pointing capability; hence, any improvement on the pointing task can be directly attributed to our proposed components. More ablation studies are provided in the supplementary material.

Proposed Components. Table 4 presents the ablation results of key components in our framework. We first examine the effect of using the image V as input to the A2P

Table 3. Generalizability experiment results. We conduct cross-dataset evaluation on the PointArena Point-Bench [4] after training on PACO [17] (direct pointing). Our method generalizes to unseen tasks and improves over native attention pointing, showing that the proposed components effectively condense textual meaning into grounding-aware queries that align with target visual concepts and produce more meaningful attention patterns across datasets and tasks. P-Acc. and Acc. denote Patch Accuracy and Accuracy, respectively.

	Affordance		Spatial		Reasoning		Steerability	
	P-Acc.	Acc.	P-Acc.	Acc.	P-Acc.	Acc.	P-Acc.	Acc.
Molmo-7B attention pointing [9]	0.495	0.419	0.436	0.369	0.513	0.430	0.285	0.275
Molmo-7B **Ours**	**0.793**	**0.748**	**0.554**	**0.508**	**0.653**	**0.591**	**0.450**	**0.410**
Qwen2.5-VL-7B attention pointing [9]	0.359	0.303	0.431	0.380	0.373	0.321	0.227	0.172
Qwen2.5-VL-7B **Ours**	**0.838**	**0.828**	**0.585**	**0.544**	**0.658**	**0.622**	**0.430**	**0.390**
First-Gen-MLLM attention pointing [9]	0.510	0.417	0.256	0.232	0.389	0.322	0.173	0.167
First-Gen-MLLM **Ours**	**0.672**	**0.641**	**0.424**	**0.384**	**0.439**	**0.378**	**0.405**	**0.369**

Table 4. Ablation study of the proposed components. Removing the Q-Synth Module leads to the largest drop, indicating that the main limitation lies in suboptimal native attention patterns. Therefore, synthesizing a grounding-aware query is crucial for extracting more accurate attention and enhancing the grounding ability of MLLMs.

Q-Synth	A2P Decoder	Image V	Accuracy
✓	✓	✓	**0.463**
✓	✓		0.453
✓			0.429
	✓	✓	0.336

Decoder. Without these features, the decoder only receives the k-channel attention maps concatenated from k localization heads, leading to suboptimal results due to the lack of fine-grained visual information.

Next, we remove the A2P Decoder entirely and use only the low-resolution attention maps for point prediction. Following [9], we aggregate attention maps across localization heads via element-wise summation. While performance decreases further (0.429), it remains noticeably higher than the baseline attention pointing from First-Gen-MLLM (0.183) as shown in Table 2. This indicates that the Q-Synth Module alone can produce more accurate and semantically consistent attention patterns.

Finally, removing the Q-Synth Module and using only the original attention maps as input to the A2P Decoder results in the largest performance drop. This confirms that the main limitation lies not merely in the low resolution of the attention maps but in their semantic precision. Even with upsampling and spatial refinement, poor attention initialization limits grounding quality. Therefore, synthesizing a grounding-aware query via the Q-Synth Module is crucial for enhancing the point grounding ability of an MLLM.

Table 5. Ablation on the number of selected localization heads. Increasing k improves performance due to richer feature diversity for the A2P Decoder.

	$k = 1$	$k = 3$	$k = 5$ (Ours)
Accuracy	0.427	0.451	**0.463**

Number of Selected heads. In Table 5, we present the performance when using different numbers of selected localization heads. We observe that increasing k generally improves pointing accuracy. This trend differs from the finding in [9], where k=3 yielded the best average performance. We attribute this difference to our A2P Decoder, which leverages image V features, and benefits from a larger set of heads that provides a richer and more diverse feature basis for refinement. When k exceeds 5, additional heads show weaker localization and less distinctive attention patterns, while computation also increases. Therefore, we do not consider larger values of k in our experiments.

6. Conclusion

In this work, we introduced a general framework that enhances Multimodal Large Language Models (MLLMs) with accurate part-level point grounding while preserving their original pre-trained capabilities. Our method refines native attention patterns through the Query Synthesis Module, and the Attention-to-Point Decoder produces high-resolution, point-centric heatmaps guided by an SDF-based penalty field, enabling precise and interpretable visual grounding. Experiments across diverse MLLMs and datasets demonstrate that our framework universally strengthens point grounding ability without retraining the base models, offering a scalable way to adapt future MLLMs for fine-grained grounding tasks.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2

[2] 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. 1, 2, 6

[3] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023. 3

[4] 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. 5, 6, 7, 8

[5] 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. 1, 2, 6

[6] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023. 1

[7] Kuan Fang, Fangchen Liu, Pieter Abbeel, and Sergey Levine. Moka: Open-world robotic manipulation through mark-based visual prompting. *Robotics: Science and Systems (RSS)*, 2024. 1, 2

[8] Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *arXiv preprint arXiv:2409.01652*, 2024. 1, 2

[9] 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. 2, 3, 4, 6, 8

[10] 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. 2, 3

[11] Daixun Li, Yusi Zhang, Mingxiang Cao, Donglai Liu, Weiying Xie, Tianlin Hui, Lunkai Lin, Zhiqiang Xie, and Yunsong Li. Towards long-horizon vision-language-action system: Reasoning, acting and memory. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6839–6848, 2025. 1

[12] Lucas Manuelli, Wei Gao, Peter Florence, and Russ Tedrake. kpam: Keypoint affordances for category-level robotic manipulation. In *The International Symposium of Robotics Research*, pages 132–157. Springer, 2019. 1

[13] Tomohiro Motoda, Takahide Kitamura, Ryo Hanai, and Yukiyasu Domae. Suctionprompt: Visual-assisted robotic picking with a suction cup using vision-language models and facile hardware design. *Journal of Robotics and Mechatronics*, 37(2):374–386, 2025. 1, 2

[14] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*, 2023. 3

[15] Renjie Pi, Lewei Yao, Jiahui Gao, Jipeng Zhang, and Tong Zhang. Perceptiongpt: Effectively fusing visual perception into llm. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 27124–27133, 2024. 2, 3

[16] Zengyi Qin, Kuan Fang, Yuke Zhu, Li Fei-Fei, and Silvio Savarese. Keto: Learning keypoint representations for tool manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7278–7285. IEEE, 2020. 1

[17] 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. 2, 5, 6, 7, 8

[18] 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. 2, 3

[19] Zhongwei Ren, Zhicheng Huang, Yunchao Wei, Yao Zhao, Dongmei Fu, Jiashi Feng, and Xiaojie Jin. Pixellm: Pixel reasoning with large multimodal model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26374–26383, 2024. 2, 3

[20] 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. 5, 6

[21] Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025. 1, 2

[22] 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. 5, 6

[23] Cong Wei, Haoxian Tan, Yujie Zhong, Yujiu Yang, and Lin Ma. Lasagna: Language-based segmentation assistant for complex queries. *arXiv preprint arXiv:2404.08506*, 2024. 2, 3

[24] 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. 2, 3

[25] Yike Wu, Jiatao Zhang, Nan Hu, Lanling Tang, Guilin Qi, Jun Shao, Jie Ren, and Wei Song. Mldt: Multi-level decomposition for complex long-horizon robotic task planning with open-source large language model. In *International Conference on Database Systems for Advanced Applications*, pages 251–267. Springer, 2024. 1

[26] Zhutian Yang, Caelan Garrett, Dieter Fox, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Guiding long-horizon task and motion planning with vision language models. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16847–16853. IEEE, 2025. 1

[27] Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. Ferret: Refer and ground anything anywhere at any granularity. *arXiv preprint arXiv:2310.07704*, 2023. 3

[28] Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. Robopoint: A vision-language model for spatial affordance prediction for robotics. *arXiv preprint arXiv:2406.10721*, 2024. 1, 2

[29] 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. 5, 6