POINTING AT PARTS: TRAINING-FREE FEW-SHOT GROUNDING IN MULTIMODAL LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Part-level pointing is important for fine-grained interaction and reasoning, yet existing Multimodal Large Language Models (MLLMs) remain limited to instance-level pointing. Part-level pointing presents unique challenges: annotation is costly, parts are long-tail distributed, and many are difficult to specify precisely in language. We introduce **PO**inting at **P**arts (POP), a training-free, plug-and-play approach that addresses these challenges under a few-shot setup. POP fuses textual and visual attention maps with self-supervised visual correspondences from query image and few-shot examples. On average across the three evaluated datasets, POP achieves accuracy gains of up to 8.9 points in the one-shot setting and 16.4 points in the three-shot setting for the pointing-capable MLLMs—Qwen2.5-VL, Ovis2.5, and Molmo. Notably, even MLLMs without pointing capability benefit significantly from the proposed approach. These results establish a simple yet effective path toward fine-grained spatial grounding in MLLMs.

1 INTRODUCTION

Pointing, or precise spatial grounding, is one of the most universal nonverbal languages in our daily communication (Tomasello et al., 2007). For example, infants point to food to express their needs, teachers point to a diagram to guide students' attention, etc. Inspired by its universality, a pointing-capable model allows agents to act in their environments and communicate rich and grounded information with humans, e.g. pointing to waypoints for navigation (Zhang et al., 2024b), pointing to paths for manipulation (Jason Lee, 2025), etc.

Recent work in Multimodal Large Language Models (MLLMs) has demonstrated promising pointing capabilities through generating pixel locations in text (Deitke et al., 2025; Bai et al., 2025; Lu et al., 2025). Despite this progress, current MLLMs fall short of realizing the full potential of pointing: Most systems operate well at the instance level but struggle pointing at the ***part level***, such as a keyboard of the laptop computer, as shown in Fig.1. Part-level pointing unlocks affordances (Yuan et al., 2024), enabling precise interaction and reasoning with objects (Jason Lee, 2025). For example, it allows robotic agents to grasp or manipulate items at the correct functional region, supports fine-grained image or video editing, and facilitates more detailed visual understanding tasks such as identifying defects (Hussain, 2023) or highlighting anatomical structures.

Moving from instance-level to part-level pointing increases both task complexity and annotation costs. At finer granularity, the long-tail of parts grows rapidly since many occur rarely or only in specific contexts. Parts are also hard to describe unambiguously: for example, "the horizontal bar connecting the two legs of a chair" or "the small tab under a soda can lid." Such expressions are often imprecise, inconsistent, or absent from common vocabularies. Few-shot examples offer a practical compromise, helping models ground references to visually specified parts without exhaustive annotations or precise terminology.

In this work, we present a training-free and plug-and-play approach that enables MLLMs to perform part-level pointing under the few-shot setup. Our approach, **PO**inting at **P**arts (POP), leverages strong text understanding from MLLMs and visual correspondences from self-supervised encoders such as DINOv3 (Siméoni et al., 2025). Specifically, it integrates language-guided attention maps from MLLMs with dense visual features that establish part-level correspondences between support and target images. By combining these complementary signals, POP enables effective few-shot part pointing with no additional training. On average across the three evaluated datasets, POP improves

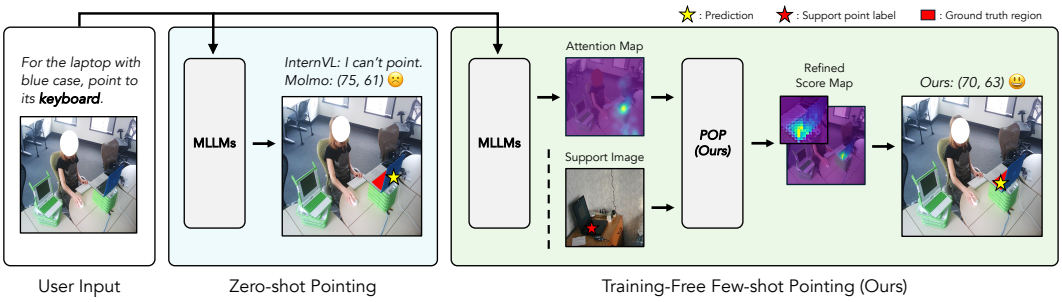

Figure 1: **We introduce a training-free approach that enhances part-level pointing in MLLMs in few-shot settings.** Prior works in MLLMs excel at instance-level pointing, but fall short in part-level pointing (blue). In this work, we present **PO**inting at **P**arts (POP) that leverages the attention maps in MLLMs and few support examples to improve part-level pointing (green). Furthermore, POP works for both point-capable and non-pointing capable MLLMs without additional training, resulting in consistent improvements across 5 MLLMs from different families.

each of the pointing-capable MLLMs —Qwen2.5-VL-7B (Bai et al., 2025), Ovis2.5-9B (Lu et al., 2025), and Molmo-7B-D (Deitke et al., 2025)—by up to **8.9** accuracy points with a single shot. Remarkably, POP also benefits MLLMs without instance-level pointing post-training—InternVL3-8B (Zhu et al., 2025) and Kimi-VL-A3B (Team et al., 2025b)—achieving improvements of up to **30.9** accuracy points with a single shot. To sum up, our contributions are as follows.

1. We propose **PO**inting at **P**arts (POP), which enables part-level pointing by combining attention information from MLLMs and self-supervised visual encoders.
2. On average across the three evaluated datasets, POP improves each of the pointing-capable MLLMs —including Qwen2.5-VL, Ovis2.5-9B, and Molmo-7B-D—by up to 8.9 accuracy points in the one-shot setting and up to 16.4 points in the three-shot setting.
3. On the same datasets, POP also benefits MLLMs without specialized pointing post-training, including InternVL3-8B and Kimi-VL-A3B, achieving improvements of up to 30.9 accuracy points with a single shot.

2 Related Works

Pointing in MLLMs. Recent advances in multimodal large language models (MLLMs) have demonstrated strong pointing capabilities (Deitke et al., 2025; Yuan et al., 2024; Team et al., 2025a; Bai et al., 2025; Lu et al., 2025). For instance, Molmo (Deitke et al., 2025) shows that pointing benefits both natural grounding and counting by sequentially localizing individual instances, thereby providing interpretable reasoning traces. In robotics, while MLLMs can provide high-level action plans in natural language, language often lacks the spatial specificity needed for reliable execution, e.g., "place the cup next to the plate". RoboPoint (Yuan et al., 2024) addresses this by grounding user query into 2D action points which are projected into 3D space for manipulation and navigation. Similarly, Gemini Robotics (Team et al., 2025a) highlights pointing as a core embodied reasoning capability, supporting tasks such as grasp prediction and trajectory planning. Beyond interacting in the physical worlds, pointing provides a natural interface for GUI agents where models directly point to elements like buttons and icons rather than predicting bounding boxes (Cheng et al., 2025).

These works establish point-based visual grounding as an emerging paradigm bridging multimodal understanding with downstream applications in robotics, GUI interaction, and counting. Building on this direction, our work focuses on training-free object part pointing, leveraging exemplars to combine the strengths of MLLMs and visual foundation models for precise part-level grounding.

Visual Grounding with Attention Maps in MLLMs. Recent works in MLLMs exploit attention maps in frozen MLLMs to highlight regions relevant to text queries, providing effective grounding without modifying their weights (Wu et al., 2025; Kang et al., 2025). F-LMM (Wu et al., 2025) uses these maps with a lightweight refinement module for visual grounding while preserving MLLMs'

knowledge and conversational ability. Kang et al. (2025) further identify Localization Heads with strong grounding ability and propose a fully training-free method to derive bounding boxes from pseudo-masks. Following this direction, we leverage attention maps from frozen MLLMs to extract part-level semantic cues for localization. By combining textual semantics with exemplar-based visual correspondences, our method achieves fine-grained part-level grounding.

Visual Semantic Correspondence. Semantic correspondence (Liu et al., 2011) aims to establish dense pixel correspondences across objects sharing the same semantics but differing in appearance, viewpoint, or deformation. Recent self-supervised representation learning, particularly with DINO (Caron et al., 2021; Oquab et al., 2024; Siméoni et al., 2025), has advanced this task by providing strong patch-level features. These representations enable reliable cross-image matching (Liu et al., 2024) and capture robust visual semantics (Amir et al., 2022; Zhang et al., 2023). Building on this idea, methods such as Matcher (Liu et al., 2024) and GF-SAM (Zhang et al., 2024a) leverage DINO-v2 (Oquab et al., 2024) as a vision foundation model to extract exemplar-to-target patch correspondences, which can then be integrated with segmentation foundation models (Kirillov et al., 2023) for training-free few-shot segmentation. These findings show that features from DINO are particularly well-suited for training-free visual correspondence, enabling effective exemplar-based semantic transfer and accurate part localization in target images. In this work, we adopt a similar strategy by utilizing the off-the-shelf vision foundation model DINO-v3 (Siméoni et al., 2025) to build visual correspondences between support and target images, which serve as semantic localization cues that are further combined with text information to achieve precise part pointing.

3 BACKGROUND

Problem Formulation and Notations. We study *few-shot part-level pointing*. In a K-shot setting, each episode consists of a support set, a target image I and a text query q describing a part of interest, e.g., a mug's handle. The goal is to predict a point p within the coordinate system of I that reflects the text query. The support set is used to provide clues to specify the part of interest in q, consisting of K tuples of support images and points (I_s, p_s).

In this work, we leverage multimodal large language models (MLLMs) and self-supervised visual encoders. An MLLM typically consists of three main components: a vision encoder, an adapter, and a large language model (LLM). Given a target image I_t, the vision encoder together with the adapter transforms it into a sequence of visual tokens with shape $\mathbb{R}^{H_l W_l \times d_l}$, where H_l and W_l denote the number of visual tokens along the height and the width, respectively, and d_l is the hidden dimension of the LLM. These visual tokens are then concatenated with the text tokens and fed into the LLM for autoregressive generation. On the other hand, a self-supervised visual encoder projects a target image and a set support images into patch-level features $z_t, z_s \in \mathbb{R}^{H_v W_v \times d_v}$, where H_v and W_v correspond to the number of patches along the height and width, and d_v is the hidden dimension of the visual encoder.

Pointing via Attention Maps in MLLMs. Previous works (Wu et al., 2025; Kang et al., 2025) have shown that attention maps in MLLMs can be used for visual grounding. In particular, F-LMM (Wu et al., 2025) demonstrate that attention scores from all heads can serve as *language-guided localization priors*. Kang et al. (2025) further identify a subset of attention heads, called *localization heads*, that consistently attend to the visual token best describing the query text. We extend their approach by using these localization head attention maps for training-free pointing: the visual token receiving the highest attention from the text token is selected, and the center of this patch is used as the predicted point. This provides a simple yet effective pointing strategy without additional training. Further details on the identification of localization heads are provided in Appendix B.1. In the next section, we propose to further enhance the use of attention maps in MLLMs as localization cues by leveraging few-shot exemplars.

4 METHOD

In this section, we introduce **PO**inting at **P**arts (POP). We first present the motivations of POP with the motivations and preliminary observations in Sec. 4.1. Then, in Sec. 4.2, we elaborate POP, a training-free, plug-and-play approach that leverages few-shot exemplars to enhance part pointing.

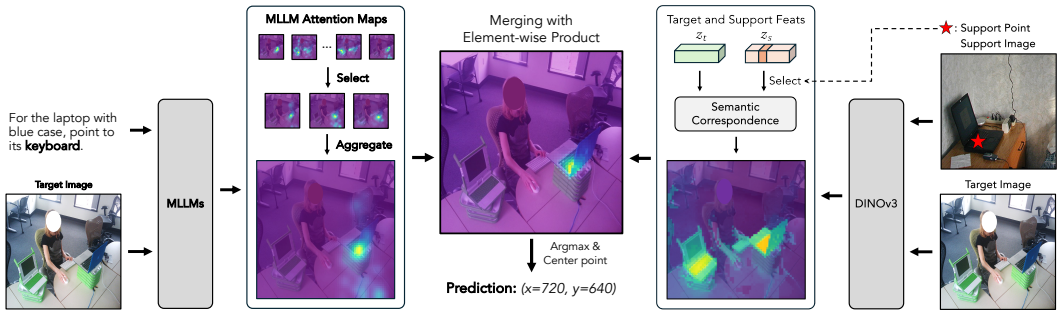

Figure 2: **Overview of the introduced approach, POP.** Our approach incorporates the attention maps from MLLMs (left) and the semantic correspondences from self-supervised visual encoder (right), such as DINOv3, to produce precise part-level point predictions.

In particular, POP uses dense visual features from self-supervised visual encoder to establish fine-grained correspondences between a target image and a labeled support set.

4.1 Motivations

In Sec. 3, Kang et al. (2025) shows the potentials to perform instance-level pointing from attention maps in MLLMs. However, when moving to part-level pointing, we find the prior work results in imprecise and coarse pointing predictions. For instance, in Fig. 3, the query "neckband" activates the correct region but still assigns the highest score to the "body", showing that text alone gives only a rough localization cue. To address the imprecision of the attention maps of localization heads in MLLMs, we explore to refine the collected attention maps by leveraging a support set and a self-supervised vision encoder.

In particular, we extract all patch-level features from the target image and labeled patch features from each of the K support images, defining patch-level similarity as the cosine similarity between them. We observe that DINOv3 (Siméoni et al., 2025) features capture fine-grained part-level information. However, this purely visual approach lacks referring ability: when multiple similar objects appear (e.g., two sweaters or several bowls), all corresponding regions are highlighted simultaneously, making it impossible to disambiguate the query.

These findings suggest that localization cues from MLLMs attention maps and patch-level correspondences between support and target images are complementary. The former provides semantic grounding and referring ability, while the latter supplies precise local matches. Integrating both yields more accurate and semantically consistent part localization.

4.2 POP: POinting at Parts

Language-guided Localization Priors. Given a query and an input image, we extract attention maps from the last query token to all image tokens using the selected localization heads within the MLLM. The attention maps are reshaped to $\mathbb{R}^{H_l \times W_l}$ and smoothed with a Gaussian filter. Finally, the smoothed maps are aggregated via element-wise summation to obtain the final language-guided localization score map $S_{\text{Text}} \in \mathbb{R}^{H_l \times W_l}$. For more details, please refer to Appendix B.2.

Visual Semantic Correspondences. Without loss of generality, we first describe the case of 1-shot part pointing. At the end of this section, we will elaborate how we extend to the K-shot setup. Given a support image I_s and point p_s, we use DINOv3 (Siméoni et al., 2025) to extract patch-level features that capture visual semantics, allowing us to locate the corresponding part in target images I_t. Our hypothesis is that target patches most similar to the support patch containing p_s likely belong to the same part (Amir et al., 2022; Zhang et al., 2023; Liu et al., 2024; Zhang et al., 2024a). Conversely, if a target patch corresponds to a part, its nearest support patch in feature space should also lie on that part.

Based on this intuition, we adopt a bidirectional spatial similarity computation strategy. The forward similarity is defined by comparing the patch containing p_s in I_s with all patches in I_t. The backward

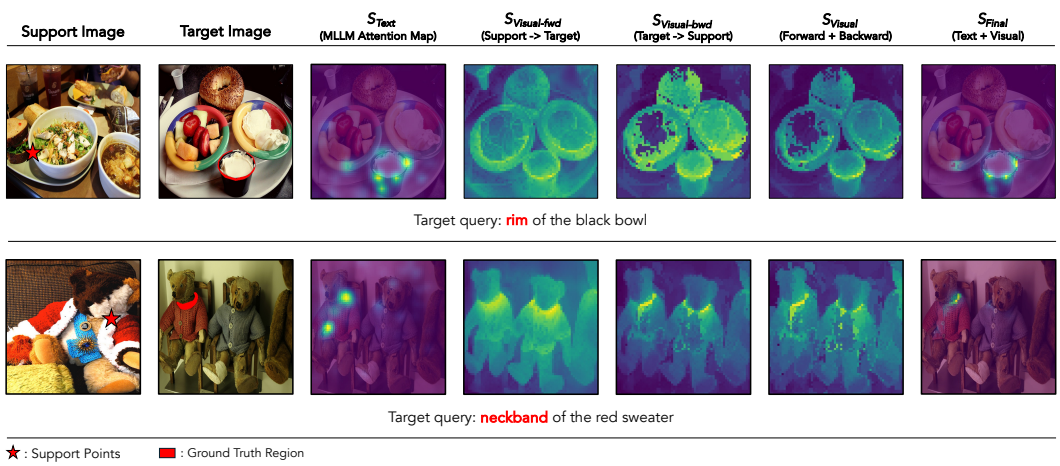

Figure 3: **Visualization of score maps from MLLM attention map, visual semantic correspondences, and their integration.** Attention maps from MLLMs provide rough localization but lack precision, while visual correspondences from DINOv3 capture fine-grained details but are ambiguous across similar objects. Integrating the two yields accurate and semantically consistent part-level localization.

similarity is computed by first finding, for each patch in I_t, its most similar patch in I_s, and then measuring the similarity between those patches and the patch containing p_s in I_s. Finally, the forward and backward similarity maps are fused to obtain the final semantic correspondence.

4.2.1 Forward Similarity

Given inputs I_s and I_t, the visual encoder produces patch-level features $z_s, z_t \in \mathbb{R}^{H_v W_v \times d_v}$. We then compute the patch-wise correspondence matrix $A \in \mathbb{R}^{H_v W_v \times H_v W_v}$ as

$$A_{ij} = \frac{z_s^i \cdot z_t^j}{\|z_s^i\| \, \|z_t^j\|},$$

where A_{ij} represents the cosine similarity between the i-th patch feature z_s^i of I_s and the j-th patch feature z_t^j of I_t.

Let i_s denote the index of the patch in z_s containing p_s. We define the forward similarity score map $S_{\text{Visual-fwd}} \in \mathbb{R}^{H_v \times W_v}$ for the target image as

$$S_{\underset{j}{\text{Visual-fwd}}} = A_{i_s j}, \quad j = 1, \ldots, H_v W_v,$$

where $S_{\text{Visual-fwd}}$ reflects the similarity between the patch containing p_s and each patch in the target image, with higher scores indicating a stronger correspondence to the same object part.

4.2.2 Backward Similarity

In contrast to the forward similarity, we also compute a backward similarity from the target image to the reference image. For each patch z_t^j in the target image I_t, we first identify its most similar patch in the support image I_s as

$$m(j) = \arg\max_i A_{ij},$$

where A_{ij} is the patch-wise similarity defined above. The index $m(j)$ thus represents the matching patch in I_s that is most similar to the j-th patch in I_t.

We then measure the relationship between this matching patch and the patch containing p_s. Let i_s denote the index of the patch in I_s containing p_s. The backward similarity score map $S^{\text{bwd}} \in \mathbb{R}^{H_v \times W_v}$ is defined as

$$S_j^{\text{bwd}} = \frac{z_s^{i_s} \cdot z_s^{m(j)}}{\|z_s^{i_s}\| \, \|z_s^{m(j)}\|}, \quad j = 1, \ldots, H_v W_v.$$

This design is motivated by the intuition that if a patch in the target image corresponds to an object part, then its most feature-similar patch in the support image should also lie on the same object part (Liu et al., 2024; Zhang et al., 2024a). Since self-supervised visual encoders exhibit strong self-correlation properties (Siméoni et al., 2021; Walmer et al., 2023), patches in the support image that are more similar to the patch containing p_s are more likely to belong to the same object part.

4.2.3 Combine Forward and Backward Similarity

The forward and backward similarities capture complementary notions of patch correspondence: the forward similarity measures how target patches relate to the support patch in feature space, while the backward similarity measures the reverse relation from target to support. As discussed above, combining these two signals helps identify the target patches corresponding to the same object part as specified by the support example.

To integrate these two signals, we compute the final visual semantic correspondence score map by element-wise multiplication of the forward and backward similarity maps:

$$S_{\text{Visual}} = S_{\text{Visual-fwd}} \odot S_{\text{Visual-bwd}},$$

where $\odot$ denotes element-wise multiplication. The resulting vector is then reshaped into $H_v \times W_v$ to form the spatial score map aligned with the target image. As shown in Figure 3, the combined visual score map highlights the part information more effectively. We can observe that both the forward and backward similarities are able to capture part-related regions, while the combined score map further enhances this signal, making the target part more prominent and localized.

4.3 Combining Textual and Visual Cues

After obtaining the language-guided localization score map $S_{\text{Text}} \in \mathbb{R}^{H_l \times W_l}$ and the visual semantic correspondence score map $S_{\text{Visual}} \in \mathbb{R}^{H_v \times W_v}$, we fuse them to achieve robust localization. Since the two maps may differ in resolution (with S_{Visual} typically higher), we first upsample S_{Text} to the size of S_{Visual} using bilinear interpolation. The final score map S_{Final} is then obtained by element-wise multiplication:

$$S_{\text{Final}} = \text{Interp}(S_{\text{Text}}) \odot S_{\text{Visual}},$$

where $\text{Interp}(\cdot)$ denotes interpolation. In the few-shot setting, we simply extend this process by multiplying the text map with each visual map from multiple support images. As shown in Figure 3, the fused score map provides sharper and more reliable localization.

4.4 Final Point Prediction

Given the final score map S_{Final},we first upsample it by a factor of two using bilinear interpolation to reduce quantization errors, then take the center of the highest-scoring patch as the predicted point. Finally, the point is mapped back to the original image coordinates, accounting for the resizing applied before encoding by the visual encoder.

5 Experiments

5.1 Experimental Settings

Datasets. We adapt three part segmentation datasets: PACO-LVIS (Ramanathan et al., 2023), InstructPart (Wan et al., 2025), and PartImageNet++ (Li et al., 2024) to the part-level pointing tasks. For all datasets, we evaluate on the test split and sample support sets from the train split for each query. PACO-LVIS contains 456 object-part categories. For our test set, we use the object instances in the official test split that are paired with referring expressions, along with their corresponding part annotations. We remove cases with extremely small ground-truth masks, resulting in 18,154 samples. InstructPart focuses on object parts in household-task scenarios. The dataset contains 2,400 images across 48 object and 44 object-part categories, with 1,800 supporting and 600 test samples. PartImageNet++ augments ImageNet (Russakovsky et al., 2015) with part annotations across diverse object categories. For each category, we sample 90 support and 10 target images, yielding 26,747 test cases over 3,308 object–part categories.

Since the datasets are constructed for segmentation, we select a representative support point from each mask by computing the innermost point (Borgefors, 1986), i.e., the point farthest from the mask boundary. When choosing the support set of each episode, unless otherwise specified, we perform random sampling. To ensure robustness, we report the average results over five different random seeds. For more details, please refer to Appendix C.1.

Metrics. Following prior work (Team et al., 2025a; Cheng et al., 2025), we assign a score of 1 if a predicted point falls inside the ground-truth part mask and 0 otherwise, and report the average accuracy score across all image-part queries.

Implementation Details. For pointing-capable MLLMs, we evaluate Molmo-7B-D (Deitke et al., 2025), Qwen2.5-VL-7B-Instruct (Bai et al., 2025), and Ovis2.5-9B (Lu et al., 2025). Non-pointing-capable MLLMs include InternVL-3-8B (Zhu et al., 2025) and Kimi-VL-Instruct-A3B (Team et al., 2025b), selected for their strong empirical performance. Following Kang et al. (2025), we use the top-3 localization heads to extract text-to-image attention maps. Unless noted otherwise, DINOv3-ViT-L/16 (Siméoni et al., 2025) serves as the self-supervised vision encoder.

In evaluation, we adopt different prompt designs for different datasets. For PACO-LVIS, we adopt the prompt template: *"For {referring expression}, point to its {part}."* For InstructPart and PartImageNet++, we use the prompt template: *"Point to the {object}'s {part}."* For non-pointing-capable MLLMs, we replace "point to" with "locate" to align with their native prompt format for visual grounding tasks. For more details, please refer to Appendix C.3.

Baselines. To contextualize our few-shot results, we include several baselines. For zero-shot reference, we report the performance of pointing-capable MLLMs and GPT-4.1. We also compare against representative open-vocabulary and reasoning segmentation models, including X-Decoder (Zou et al., 2023a), SEEM (Zou et al., 2023b), VL-Part (Sun et al., 2023)(a part-specialist trained on PACO and related datasets) and LISA (Lai et al., 2024).

For few-shot baselines, we first evaluate pointing-capable MLLMs with in-context learning (Brown et al., 2020), where support examples are provided in the context. We also consider two strong training-free segmentation methods, Matcher (Liu et al., 2024) and GF-SAM (Zhang et al., 2024a), which build on DINOv2-ViT-L/14 (Oquab et al., 2024) and SAM (Kirillov et al., 2023). For fairness, we substitute DINOv2 with DINOv3-ViT-L/16 (Siméoni et al., 2025) at 1024 resolution. These models are evaluated with full support masks, whereas our method only uses a single support point.

For segmentation models outputting masks, we extract a representative point by default from the maximum-logit pixel; if only a binary mask is available, we instead select the innermost point via distance transform (Borgefors, 1986). For more details, please refer to Appendix C.2.

5.2 Main Results

5.2.1 Part Pointing with Pointing-capable MLLMs

Tab. 1 presents the performance of our method on PACO, InstructPart, and PartImageNet++, compared with various baseline models. In 1-shot, POP consistently surpasses the zero-shot pointing performance of original pointing-capable MLLMs. Averaged across the three datasets, it improves accuracy by 8.9 points on Qwen2.5-VL, 6.5 points on Ovis2.5, and 5.5 points on Molmo. It also outperforms part-specialized zero-shot segmentation models such as VL-Part, as well as few-shot baselines, including segmentation methods like Matcher and GF-SAM, and in-context learning (ICL) baselines. Notably, we observe that ICL can sometimes degrade visual grounding performance for pointing-capable MLLMs, consistent with findings in multi-image grounding (Li et al., 2025). In contrast, our training-free method effectively leverages exemplars.

Extending to the few-shot setting, our method scales effectively with more examples. Performance consistently improves from one-shot to three-shot across all datasets. Averaged over the three datasets, it achieves accuracy gains of 16.4 points on Qwen2.5-VL, 13.1 points on Ovis2.5, and 10.6 points on Molmo. Notably, as the number of examples increases, the performance gap between our full method and its visual-only variant, DINOv3, narrows on InstructPart and PartImageNet++, where the images are relatively simple. In contrast, on PACO—which contains complex scenes with

Table 1: **Results of pointing-capable MLLMs.** Our method consistently improves over the original pointing-capable MLLMs with just 1-shot, and achieves further gains with 3-shot. For segmentation models, * indicates that the representative point is obtained via distance transform, as described in Sec. 5.1. For pointing-capable MLLMs, 1-shot and 3-shot use in-context learning; Molmo supports only single-image input and thus has no in-context results. POP (ours) uses attention-based pointing, and DINOv3 relies solely on visual features. Best results per shot are highlighted in bold.

Method	**PACO**			**InstructPart**			**PartImageNet++**		
Zero-shot Segmentation Models									
X-Decoder	17.2			37.3			33.0		
SEEM	15.6			32.2			32.5		
LISA-7B	26.7			57.2			49.9		
VL-Part	38.3*			55.2*			54.4*		
Zero-shot Proprietary MLLMs									
GPT-4.1	19.8			51.5			48.9		
Few-shot Segmentation Models									
		1-shot	3-shot		1-shot	3-shot		1-shot	3-shot
GF-SAM		16.6*	19.8*		53.5*	55.5*		47.9*	49.4*
Matcher		33.3*	40.5*		82.0*	84.3*		71.7*	79.2*
Pointing-capable MLLMs									
	0-shot	1-shot	3-shot	0-shot	1-shot	3-shot	0-shot	1-shot	3-shot
Qwen2.5-VL-7B	47.7	14.1	13.1	78.7	40.3	43.0	66.6	30.8	26.5
Ovis2.5-9B	47.5	37.8	32.8	79.8	74.2	72.7	**76.4**	71.9	70.5
Molmo-7B-D	**51.2**	N/A	N/A	**87.2**	N/A	N/A	75.9	N/A	N/A
*POP (**Ours**): Few-shot Attention-based Pointing*									
	0-shot	1-shot	3-shot	0-shot	1-shot	3-shot	0-shot	1-shot	3-shot
DINOv3	N/A	41.5	55.6	N/A	84.2	91.8	N/A	77.3	87.0
Qwen2.5-VL-7B	31.6	51.6	61.7	68.2	87.9	93.4	58.3	80.1	87.1
Ovis2.5-9B	31.5	53.0	61.8	65.2	88.5	93.0	55.1	81.7	88.2
Molmo-7B-D	42.3	**55.8**	**62.4**	74.8	**90.9**	**94.5**	69.1	**84.1**	**89.1**

Table 2: **Results of non-pointing-capable MLLMs.** Our method can be applied to MLLMs without specialized pointing training. With a single shot, it achieves performance comparable to the zero-shot pointing-capable MLLMs. For non-pointing-capable MLLMs, zero-shot pointing performance is obtained using attention-based method.

Method	**PACO**		**InstructPart**		**PartImageNet++**	
	0-shot	1-shot	0-shot	1-shot	0-shot	1-shot
DINOv3	N/A	41.5	N/A	84.2	N/A	77.3
InternVL-3-8B	23.6	45.7	56.6	86.1	54.0	78.3
Kimi-VL-A3B	24.9	49.7	57.2	90.3	44.6	79.4

multiple objects and fine-grained referring expressions—DINOv3 still falls behind, underscoring the advantage of jointly leveraging language and vision in more challenging scenarios.

5.2.2 Part Pointing with Non-pointing-capable MLLMs

Tab. 2 shows the performance of our method on MLLMs without pointing post-training. Despite lacking specialized supervision, POP leverages both text and visual cues, improving accuracy by 25.3 points on InternVL-3 and 30.9 points on Kimi-VL on average across the three datasets. Remarkably, even with a single example, it achieves performance comparable to the zero-shot pointing-capable MLLMs in Tab. 1, demonstrating that frozen general-purpose MLLMs can serve as foundation models for pointing tasks without point-specialist training.

Table 3: **Results of random versus [CLS]-retrieved support examples.** Selecting semantically similar supports further improves our method's performance. The better result between the two strategies is highlighted in bold.

Method	PACO		InstructPart		PartImageNet++	
	Random	[CLS]	Random	[CLS]	Random	[CLS]
DINOv3	41.5	**49.7**	84.2	**88.2**	77.3	**84.1**
Molmo-7B-D	55.8	**58.7**	90.9	**91.3**	84.1	**86.0**
Qwen2.5-VL-7B	51.6	**54.2**	87.9	**89.3**	80.1	**82.4**
Ovis2.5-9B	53.0	**57.0**	88.5	**89.7**	81.7	**84.6**

Table 4: **Ablation study results on the PACO dataset with Molmo-7B-D.**

(a) Effect of forward and backward similarity for visual semantic correspondence.

Method	Forward	Backward	Accuracy
Molmo-7B-D	–	–	51.2
POP (Ours)	✓	✗	54.5
POP (Ours)	✗	✓	51.8
POP (Ours)	✓	✓	**55.8**

(b) Ablation study of merging strategies. λ denotes the weight for S_{Visual}, i.e., $\lambda \cdot S_{Visual} + (1 - \lambda) \cdot S_{Text}$.

Merging Method	Accuracy			
Sum	$\lambda = 0.45$	$\lambda = 0.5$	$\lambda = 0.55$	$\lambda = 0.6$
	54.7	55.6	55.7	54.8
Product (Ours)	**55.8**			

5.2.3 Effect of Support Example Quality

Previously, we reported average accuracy over five random seeds with randomly selected support examples. Here, we investigate selecting semantically similar supports. Leveraging DINO's strong training-free image retrieval capability(Oquab et al., 2024; Siméoni et al., 2025), we compute similarity using the [CLS] token and select the support most aligned with the target image. As shown in Tab. 3, this careful selection improves few-shot performance for the purely visual baseline (DINOv3) and allows our method to further benefit from combining visual and textual information, achieving even higher accuracy.

5.3 Ablation Study

We conduct ablation studies on the PACO dataset with Molmo-7B-D to analyze the contributions of different design choices. We focus on (i) forward and backward similarity for visual semantic correspondences, and (ii) strategies for merging textual and visual localization cues.

Forward and Backward Similarity. Tab. 4a shows that using forward or backward similarity alone already improves over the zero-shot baseline of 51.2 accuracy points. Combining both raises accuracy to 55.8, indicating that the two directions are complementary and jointly enhance pointing performance with text guidance.

Combining Textual and Visual Cues. For merging textual and visual cues, we compare weighted summation and element-wise product. As shown in Tab. 4b, a weighted sum with $\lambda = 0.55$ reaches 55.7 accuracy points, while the product achieves a similar 55.8 accuracy points without tuning, making it a simpler and more robust choice. We thus adopt the product as our final merging method.

6 Conclusion

We show that part-level pointing can be effectively enabled in MLLMs using **PO**inting at **P**arts (POP), a training-free, plug-and-play approach that integrates attention maps from MLLMs with visual correspondences from self-supervised encoders. On average across the three evaluated datasets, POP improves each of the pointing-capable baselines (Qwen2.5-VL-7B, Ovis2.5-9B, and Molmo-7B-D) by up to 8.9 accuracy points with a single shot, and benefits MLLMs without pointing-specific post-training (InternVL3-8B, Kimi-VL-A3B) by up to 30.9 points with one shot. These results demonstrate that POP provides a simple and effective strategy for few-shot part-level localization and grounding in MLLMs.

REFERENCES

Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *ECCVW What is Motion For?*, 2022.

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. *arXiv preprint arXiv:2502.13923*, 2025.

Michael Batty. Spatial entropy. *Geographical Analysis*, 6(1):1–31, 1974.

Gunilla Borgefors. Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, 34(3):344–371, 1986. ISSN 0734-189X. doi: https://doi.org/10.1016/S0734-189X(86)80047-0. URL https://www.sciencedirect.com/science/article/pii/S0734189X86800470.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9650–9660, October 2021.

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. URL https://arxiv.org/abs/2505.09990.

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, Jiasen Lu, Taira Anderson, Erin Bransom, Kiana Ehsani, Huong Ngo, YenSung Chen, Ajay Patel, Mark Yatskar, Chris Callison-Burch, Andrew Head, Rose Hendrix, Favyen Bastani, Eli VanderBilt, Nathan Lambert, Yvonne Chou, Arnavi Chheda, Jenna Sparks, Sam Skjonsberg, Michael Schmitz, Aaron Sarnat, Byron Bischoff, Pete Walsh, Chris Newell, Piper Wolters, Tanmay Gupta, Kuo-Hao Zeng, Jon Borchardt, Dirk Groeneveld, Crystal Nam, Sophie Lebrecht, Caitlin Wittlif, Carissa Schoenick, Oscar Michel, Ranjay Krishna, Luca Weihs, Noah A. Smith, Hannaneh Hajishirzi, Ross Girshick, Ali Farhadi, and Aniruddha Kembhavi. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 91–104, June 2025.

Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Muhammad Hussain. Yolo-v1 to yolo-v8, the rise of yolo and its complementary nature toward digital manufacturing and industrial defect detection. *Machines*, 2023. URL https://api.semanticscholar.org/CorpusID:259717436.

Haoquan Fang Yuquan Deng Shuo Liu Boyang Li Bohan Fang Jieyu Zhang Yi Ru Wang Sangho Lee Winson Han Wilbert Pumacay Angelica Wu Rose Hendrix Karen Farley Eli VanderBilt Ali Farhadi Dieter Fox Ranjay Krishna Jason Lee, Jiafei Duan. Molmoact: Action reasoning models that can reason in space. *arXiv preprint arXiv:2508.07917*, 2025.

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 (CVPR)*, pp. 9339–9350, June 2025.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. ReferItGame: Referring to objects in photographs of natural scenes. In Alessandro Moschitti, Bo Pang, and Walter Daelemans (eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 787–798, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1086. URL https://aclanthology.org/D14-1086/.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3992–4003, 2023. doi: 10.1109/ICCV51070.2023.00371.

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 (CVPR)*, pp. 9579–9589, June 2024.

Xiao Li, Yining Liu, Na Dong, Sitian Qin, and Xiaolin Hu. Partimagenet++ dataset: Scaling up part-based models for robust recognition. In *European conference on computer vision*. Springer, 2024.

You Li, Heyu Huang, Chi Chen, Kaiyu Huang, Chao Huang, Zonghao Guo, Zhiyuan Liu, Jinan Xu, Yuhua Li, Ruixuan Li, and Maosong Sun. Migician: Revealing the magic of free-form multi-image grounding in multimodal large language models, 2025. URL https://arxiv.org/abs/2501.05767.

Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):978–994, 2011. doi: 10.1109/TPAMI.2010.147.

Yang Liu, Muzhi Zhu, Hengtao Li, Hao Chen, Xinlong Wang, and Chunhua Shen. Matcher: Segment anything with one shot using all-purpose feature matching. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=yzRXdhk2he.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Shiyin Lu, Yang Li, Yu Xia, Yuwei Hu, Shanshan Zhao, Yanqing Ma, Zhichao Wei, Yinglun Li, Lunhao Duan, Jianshan Zhao, Yuxuan Han, Haijun Li, Wanying Chen, Junke Tang, Chengkun Hou, Zhixing Du, Tianli Zhou, Wenjie Zhang, Huping Ding, Jiahe Li, Wen Li, Gui Hu, Yiliang Gu, Siran Yang, Jiamang Wang, Hailong Sun, Yibo Wang, Hui Sun, Jinlong Huang, Yuping He, Shengze Shi, Weihong Zhang, Guodong Zheng, Junpeng Jiang, Sensen Gao, Yi-Feng Wu, Sijia Chen, Yuhui Chen, Qing-Guo Chen, Zhao Xu, Weihua Luo, and Kaifu Zhang. Ovis2.5 technical report, 2025. URL https://arxiv.org/abs/2508.11737.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt. Featured Certification.

Elia Peruzzo, Enver Sangineto, Yahui Liu, Marco De Nadai, Wei Bi, Bruno Lepri, and Nicu Sebe. Spatial entropy as an inductive bias for vision transformers, 2022. URL https://arxiv.org/abs/2206.04636.

Vignesh Ramanathan, Anmol Kalia, Vladan Petrovic, Yi Wen, Baixue Zheng, Baishan Guo, Rui Wang, Aaron Marquez, Rama Kovvuri, Abhishek Kadian, Amir Mousavi, Yiwen Song, Abhimanyu Dubey, and Dhruv Mahajan. Paco: Parts and attributes of common objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7141–7151, June 2023.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Oriane Siméoni, Gilles Puy, Huy V. Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce. Localizing objects with self-supervised transformers and no labels. In *Proceedings of the British Machine Vision Conference (BMVC)*, November 2021.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3, 2025. URL https://arxiv.org/abs/2508.10104.

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 (ICCV)*, pp. 15453–15465, October 2023.

Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, Steven Bohez, Konstantinos Bousmalis, Anthony Brohan, Thomas Buschmann, Arunkumar Byravan, Serkan Cabi, Ken Caluwaerts, Federico Casarini, Oscar Chang, Jose Enrique Chen, Xi Chen, Hao-Tien Lewis Chiang, Krzysztof Choromanski, David D'Ambrosio, Sudeep Dasari, Todor Davchev, Coline Devin, Norman Di Palo, Tianli Ding, Adil Dostmohamed, Danny Driess, Yilun Du, Debidatta Dwibedi, Michael Elabd, Claudio Fantacci, Cody Fong, Erik Frey, Chuyuan Fu, Marissa Giustina, Keerthana Gopalakrishnan, Laura Graesser, Leonard Hasenclever, Nicolas Heess, Brandon Hernaez, Alexander Herzog, R. Alex Hofer, Jan Humplik, Atil Iscen, Mithun George Jacob, Deepali Jain, Ryan Julian, Dmitry Kalashnikov, M. Emre Karagozler, Stefani Karp, Chase Kew, Jerad Kirkland, Sean Kirmani, Yuheng Kuang, Thomas Lampe, Antoine Laurens, Isabel Leal, Alex X. Lee, Tsang-Wei Edward Lee, Jacky Liang, Yixin Lin, Sharath Maddineni, Anirudha Majumdar, Assaf Hurwitz Michaely, Robert Moreno, Michael Neunert, Francesco Nori, Carolina Parada, Emilio Parisotto, Peter Pastor, Acorn Pooley, Kanishka Rao, Krista Reymann, Dorsa Sadigh, Stefano Saliceti, Pannag Sanketi, Pierre Sermanet, Dhruv Shah, Mohit Sharma, Kathryn Shea, Charles Shu, Vikas Sindhwani, Sumeet Singh, Radu Soricut, Jost Tobias Springenberg, Rachel Sterneck, Razvan Surdulescu, Jie Tan, Jonathan Tompson, Vincent Vanhoucke, Jake Varley, Grace Vesom, Giulia Vezzani, Oriol Vinyals, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Fei Xia, Ted Xiao, Annie Xie, Jinyu Xie, Peng Xu, Sichun Xu, Ying Xu, Zhuo Xu, Yuxiang Yang, Rui Yao, Sergey Yaroshenko, Wenhao Yu, Wentao Yuan, Jingwei Zhang, Tingnan Zhang, Allan Zhou, and Yuxiang Zhou. Gemini robotics: Bringing ai into the physical world, 2025a. URL https://arxiv.org/abs/2503.20020.

Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, Congcong Wang, Dehao Zhang, Dikang Du, Dongliang Wang, Enming Yuan, Enzhe Lu, Fang Li, Flood Sung, Guangda Wei, Guokun Lai, Han Zhu, Hao Ding, Hao Hu, Hao Yang, Hao Zhang, Haoning Wu, Haotian Yao, Haoyu Lu, Heng Wang, Hongcheng Gao, Huabin Zheng, Jiaming Li, Jianlin Su, Jianzhou Wang, Jiaqi Deng, Jiezhong Qiu, Jin Xie, Jinhong Wang, Jingyuan Liu, Junjie Yan, Kun Ouyang, Liang Chen, Lin Sui, Longhui Yu, Mengfan Dong, Mengnan Dong, Nuo Xu, Pengyu Cheng, Qizheng Gu, Runjie Zhou, Shaowei Liu, Sihan Cao, Tao Yu, Tianhui Song, Tongtong Bai, Wei Song, Weiran He, Weixiao Huang, Weixin Xu, Xiaokun Yuan, Xingcheng Yao, Xingzhe Wu, Xinhao Li, Xinxing Zu, Xinyu Zhou, Xinyuan Wang, Y. Charles, Yan Zhong, Yang Li, Yangyang Hu, Yanru Chen, Yejie Wang, Yibo Liu, Yibo Miao, Yidao Qin, Yimin Chen, Yiping Bao, Yiqin Wang, Yongsheng Kang, Yuanxin Liu, Yuhao Dong, Yulun Du, Yuxin Wu, Yuzhi Wang, Yuzi Yan, Zaida Zhou, Zhaowei Li, Zhejun Jiang,

Zheng Zhang, Zhilin Yang, Zhiqi Huang, Zihao Huang, Zijia Zhao, Ziwei Chen, and Zongyu Lin. Kimi-vl technical report, 2025b. URL https://arxiv.org/abs/2504.07491.

Michael Tomasello, Malinda Carpenter, and Ulf Liszkowski. A new look at infant pointing. *Child development*, 78 3:705–22, 2007. URL https://api.semanticscholar.org/CorpusID:12990844.

Matthew Walmer, Saksham Suri, Kamal Gupta, and Abhinav Shrivastava. Teaching matters: Investigating the role of supervision in vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7486–7496, 2023.

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. URL https://openreview.net/forum?id=IMEr4XgJSZ.

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 (CVPR)*, pp. 24710–24721, June 2025.

Jianwei Yang, Chunyuan Li, Xiyang Dai, and Jianfeng Gao. Focal modulation networks, 2022.

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 in robotics. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=GVX6jpZOhU.

Anqi Zhang, Guangyu Gao, Jianbo Jiao, Chi Harold Liu, and Yunchao Wei. Bridge the points: Graph-based few-shot segment anything semantically. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a. URL https://openreview.net/forum?id=jYypS5VIPj.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Luisa Polania Cabrera, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. A tale of two features: Stable diffusion complements DINO for zero-shot semantic correspondence. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=lds9D17HRd.

Kaidong Zhang, Pengzhen Ren, Bingqian Lin, Junfan Lin, Shikui Ma, Hang Xu, and Xiaodan Liang. PIVOT-r: Primitive-driven waypoint-aware world model for robotic manipulation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=gnXTDQyxlU.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, Nanyun Peng, Lijuan Wang, Yong Jae Lee, and Jianfeng Gao. Generalized decoding for pixel, image, and language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15116–15127, June 2023a.

Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023b. URL https://openreview.net/forum?id=UHBrWeFWlL.