MolmoPoint: Better Pointing for VLMs with Grounding Tokens

Christopher Clark[♥1], Yue Yang[♥1], Jae Sung Park[♥1], Zixian Ma[1,2], Jieyu Zhang[1,2], Rohun Tripathi[1], Mohammadreza Salehi[1,2], Sangho Lee[1], and Ranjay Krishna[1,2]

[1] Allen Institute for AI, Seattle WA, USA
[2] University of Washington, Seattle WA, USA
molmo@allenai.org

Abstract. Grounding has become a fundamental capability of vision-language models (VLMs). Most existing VLMs point by generating coordinates as part of their text output, which requires learning a complicated coordinate system and results in a high token count. Instead, we propose a more intuitive pointing mechanism that directly selects the visual tokens that contain the target concept. Our model generates a special pointing token that cross-attends to the input image or video tokens and selects the appropriate one. To make this model more fine-grained, we follow these pointing tokens with an additional special token that selects a fine-grained subpatch within the initially selected region, and then a third token that specifies a location within that subpatch. We further show that performance improves by generating points sequentially in a consistent order, encoding the relative position of the previously selected point, and including a special *no-more-points* class when selecting visual tokens. Using this method, we set a new state-of-the-art on image pointing (70.7% on PointBench), set a new state-of-the-art for fully open models on GUI pointing (61.1% on ScreenSpotPro), substantially improve VLM video tracking (62.5 on $\mathcal{J}\&\mathcal{F}$ vs 56.7 for Molmo2 on Molmo2Track), and improve video pointing (59.1% human preference win rate vs. Molmo2). We additionally show that our method significantly improves learning efficiency and discuss the qualitative differences that emerge with this design change. Our model weights, new datasets, and source code are available at https://allenai.org/blog/molmopoint.

Keywords: VLM · Visual Grounding · Token Referencing

1 Introduction

Grounding through pointing is a key capability for vision-language models (VLMs). Pointing has direct applications to robotics, where points have been shown to be an effective way for VLMs to build plans for grasping or navigation [37, 66, 86]. Computer user agents have increasingly used pointing to determine how to interact with graphical user interfaces (GUIs) [58, 69, 74, 100]. Pointing can also

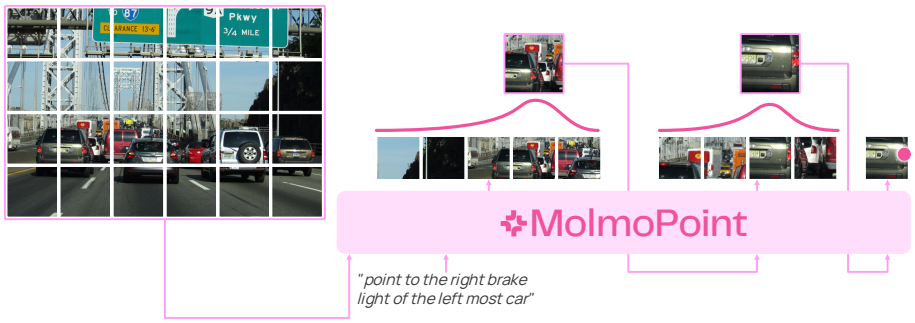

Fig. 1: MolmoPoint. To point, our model scores coarse-grained image patches using the LLM's hidden states, then scores finer-grained subpatches using the ViT patch features, and finally selects a point within the highest-scoring subpatch.

be used with chain-of-thought to improve performance on tasks like counting [19,22], and it can be used to refer back to visual input when communicating with users to provide clearer and more interpretable responses.

VLMs typically point in one of two ways: by directly generating text coordinates [22, 28, 83], or by generating special tokens that correspond to discretized coordinate bins [14,45]. Instead, as shown in Figure 1, we propose to use *grounding tokens* that directly select visual tokens from the input video or image.

To predict a point, the model emits three special grounding tokens, `<PATCH>`, `<SUBPATCH>`, and `<LOCATION>`, that generate a point in a coarse-to-fine manner. The `<PATCH>` token is used to select a coarse-grained patch in the input image (or video) by attending to the hidden states of the LLM's visual tokens. The `<SUBPATCH>` token selects a subpatch within that patch by attending to the ViT features of finer-grained patches within that patch. Finally, the `<LOCATION>` token selects a point within the subpatch. When used as input, the `<PATCH>` token and `<SUBPATCH>` token use embeddings derived from the selected patch and subpatch. This allows the model to carry forward location information as it generates future tokens.

To give the model additional awareness of what it has already pointed to, we apply rotary embeddings (RoPE) [64] when selecting a patch to encode how far candidate patches are from the patch selected by the *previous* `<PATCH>` token. This encoding makes it easier for the model to generate consistent, ordered points and avoid double-pointing. We also allow `<PATCH>` tokens to emit a no-more-points class instead of selecting a token to indicate that the model should stop pointing. We show that this prevents degenerate behavior where the model generates an excessive number of points.

Our approach has several practical advantages. First, the model no longer needs to learn or memorize a coordinate system, which we show makes learning faster and improves generalization across image resolutions unseen during training. Second, it reduces the number of output tokens required to represent each point, lowering the decoding cost and improving inference latency. Third,

it more tightly couples visual recognition and pointing: if the model has already encoded an object, action, or part in the hidden state of a visual token, it becomes trivial to point to that content by generating a query vector that matches its embedding. We show that this leads to stronger pointing performance and shows signs of improving transfer to tasks beyond grounding.

To explore this approach, we train three models: (1) **MolmoPoint-8B**: a general-purpose image and video VLM following the Molmo2 pipeline, (2) **MolmoPoint-Img-8B**: a model specialized for GUI pointing, and (3) **MolmoPoint-Vid-4B**: a lighter-weight model specialized for video pointing. To train MolmoPoint-Img-8B, we construct **MolmoPoint-PointSyn**, a new synthetic dataset of high-resolution GUI grounding examples by extending the code-guided data generation method of CoSyn [89]. To improve tracking in MolmoPoint-8B, we also contribute **MolmoPoint-Track**, a dataset of human-annotated and synthetic tracks for broader object and scene coverage.

We evaluate these models across many pointing tasks. For natural images, MolmoPoint-8B sets a new SoTA on PointBench [16] and PixMo-Points [22], beating the previous methods by 2 points and 4 points, respectively. For GUI pointing, we find MolmoPoint-Img-8B achieves over 5 points better on ScreenSpot-Pro [38] and 4 points better on OSWorldG [80] compared to a baseline using text coordinates, and is SoTA among models of a similar size that have open data. For video pointing, MolmoPoint-8B shows a several-point gain on counting metrics and better human preference scores compared to Molmo2, despite being trained on the same data, and MolmoPoint-Vid-4B further improves these metrics. For video tracking, MolmoPoint-8B reaches 62.5 on $\mathcal{J}\&\mathcal{F}$ vs 56.7 for Molmo2 and shows large gains from both our new data and model design. We also show that our approach improves training and sample efficiency and has notable qualitative effects on the pointing behavior. Our models, code, and data are available at https://allenai.org/blog/molmopoint.

2 Related Work

Generating Coordinates. Generating text coordinates or discrete tokens for grounding is an old approach for VLMs [14, 45, 46, 71]. Large-scale pointing datasets such as PixMo-Points [22] have allowed VLMs to handle pointing across a wide range of objects and images [22], and many recent VLMs have adopted this capability [8, 28, 44, 73, 83, 93]. MolmoPoint-8B shows that using grounding tokens can provide a stronger and more efficient way to learn this skill.

GUI Grounding. Pointing for GUIs has gained interest due to its use for computer-use agents [42, 74, 76]. Existing methods often try to improve performance by enhancing data generation [15, 24, 58, 81] or by using reinforcement learning [67, 69, 94, 100]. Other works have improved GUI grounding through agentic, multi-step strategies such as zooming in and cropping the input screenshot [96, 100], although this comes at the expense of higher compute costs. Our work shows that improving the point representation can also significantly enhance GUI grounding.

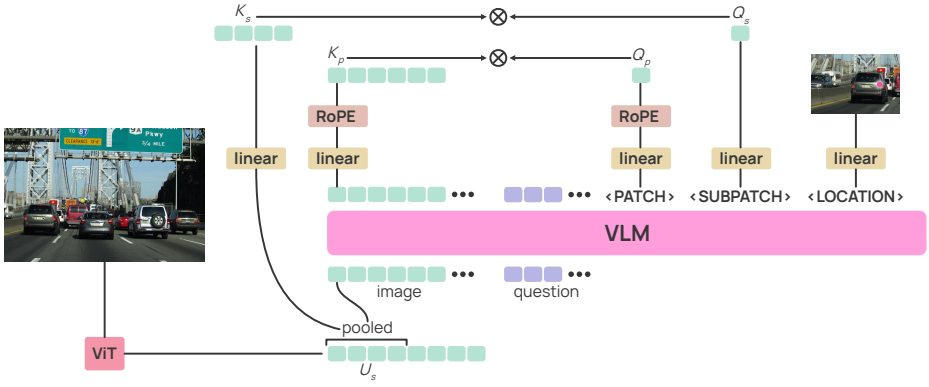

Fig. 2: Pointing with grounding tokens. Keys are built from image tokens and ViT patch features, and queries are built from the `<PATCH>` token and `<SUBPATCH>` token hidden states, to score patches and subpatches. The `<LOCATION>` token predicts the final output points within the highest scoring subpatch.

GUI Grounding Datasets. Existing GUI grounding datasets have been built both purely synthetically [29, 29, 77, 89, 90] and with humans [11, 24, 32]. Our MolmoPoint-PointSyn differs in that it focuses on high-resolution images with greater diversity across domains, resolutions, and aspect ratios.

Video Grounding. Open-vocabulary video grounding is still generally done by specialized models [2, 7, 41, 82], with only a few VLMs supporting this capability [19, 28]. We believe that grounding should not be limited to images, which is partly why we build on top of the Molmo2 models that support video pointing. Our results suggest that token referencing can help in this domain as well.

Grounding Tokens. Grounding tokens have been used for tasks such as image segmentation [7, 8, 35, 59] or depth estimation [9, 36]. These methods typically require a pre-trained decoding model, whereas our method only uses a few lightweight projectors.

More similar to our work, PaDT [65] builds embeddings from the hidden states of the visual tokens and adds them to the output vocabulary so they can be directly selected. However, their approach still uses a separate decoder instead of our coarse-to-fine approach, and is not applied to GUIs and videos.

GUI-Actor [76] also allows cross-attention between a special `<ACTOR>` token and visual patches; however, it does not add refinement stages to allow high-precision pointing and only applies their methods to GUIs and single points.

3 Method

Our approach trains the model to point by directly selecting which visual token contains the target object and then refining that location by generating additional tokens. We describe it in more detail below.

3.1 Patch Selection

First, we add a special <PATCH> token to the model's vocabulary. When this token is generated, a query vector is built from its hidden state:

$$q_p = W_{pq}\text{Norm}(h_p)$$

Here W_{pq} is a learned parameter with shape $M \times D$, h_p is the hidden state of the <PATCH> token with shape $D \times 1$, Norm is a layer norm layer, D is the model's hidden size, and M is a hyper-parameter. We also generate key vectors for each <IMAGE> token that embeds visual input as:

$$K_p = W_{pk}\text{Norm}(H_i)$$

where W_{pk} is another learned parameter with shape $M \times D$, H_i are the hidden states of the <IMAGE> tokens with shape $D \times I$, and I is the number of image tokens. Finally, we score each image token as

$$s_p = K_p^\top q_p / \sqrt{M}$$

The score vector s_p has shape $I \times 1$. During training, we compute the loss of this selection process as:

$$L_p = \text{cross\_entropy}(\text{softmax}(s_p), t_p)$$

where t_p is the ground truth target token. L_p is added directly to the token-level loss from the LLM before that loss is averaged by the number of tokens.

During inference, we select the highest scoring token $p^* = \text{argmax}(s_p)$. During training, we instead use $p^* = t_p$. Then, when <PATCH> token is an input, we add the input embedding of the <IMAGE> token that was selected to its embedding: $q_p + E_i[:, p^*]$, where E_i are input embeddings of the <IMAGE> tokens. This is important so the model is aware of which token it pointed to.

During training, we sort ground truth points so that the <IMAGE> tokens the <PATCH> tokens select are ordered based on where they appear in the input sequence. We mask out <IMAGE> tokens that come before previously selected <IMAGE> tokens during both training and inference to enforce this pattern.

3.2 Location Refinement

In most VLMs, image tokens are constructed by pooling multiple patches from the underlying ViT. For example, in Molmo2 models, each <IMAGE> token is built from 4 ViT patches that each cover 14x14 pixels, so it represents a 28x28 pixel area. To get a more fine-grained location, our model emits a <SUBPATCH> token that selects one of the ViT patches that were pooled to build p^*, also using dot-product scoring. The hidden state of the <SUBPATCH> token h_s is projected to create a query vector q_s, and key vectors K_s are built by projecting the ViT features for the subpatches U_s, where U_s is a $T \times K$ matrix, K is the number of subpatches, and T is the dimensionality of the ViT.

We similarly use the ground-truth subpatch location to compute a loss L_s for this component during training and select a subpatch index s^*. When a <SUBPATCH> token is used as input, its embedding is built from the hidden state of the selected ViT patch: $q_s + W_{se}U_s[:, s^*]$ where W_{se} with shape $D \times T$ projects the ViT patch feature to the LLM's dimension. Adding this embedding indicates to the LLM which subpatch was selected and gives the model access to the unpooled features of the selected patch, which we find important when trying to further refine the location.

This gives us a 14x14 resolution, which can still be too coarse-grained. To produce a precise point, we emit a final <LOCATION> token. The hidden state of the <LOCATION> token is used to predict one of 9 locations within the subpatch (arranged in a 3x3 grid) using a single linear layer. With 14x14 ViT patches, this results in a precision of about 4.7 pixels. Unlike pointing with text coordinates, this method maintains a 4-pixel resolution regardless of input size, potentially enabling fine-grained pointing even with ultra-HD images.

3.3 Rotary Embedding

We add rotary embeddings to better encode how <IMAGE> tokens are positioned relative to the previously selected <PATCH> token. This is important to help the model follow the sorted order of points or to track what frames the previous points were generated for when doing video pointing. This is implemented by rotating the <PATCH> token key and query vectors:

$$s_p = \text{Rot}(K_p, p_i)^\top \text{Rot}(q_p, p_q)/\sqrt{M}$$

Where p_i contains the <IMAGE> token position $[0, 1, 2, ..., I]$ and p_q is the image position selected by the previous <PATCH> token if there is one, or 0.

3.4 No-More-Points Class

One issue with this approach is that if the model chooses to generate a <PATCH> token, it is forced to select a point, even if none of the scores in s_p are high. We observe that this can sometimes lead to degenerate output, where the model generates an excessive number of points. To solve this, we add a special *no-more-points* class with a fixed key embedding that the <PATCH> token can attend to, meaning we have:

$$K_p = [W_{pk}\text{Norm}(h_i); h_{done}]$$

Where h_{done} is a learned $M \times 1$ vector. We use a position of 0 for h_{done} when applying rotary embeddings. If the no-more-points class is selected, the model is prevented from generating a <SUBPATCH> token and stops pointing.

4 Training and Inference

We train three models using this proposed method. We present high-level details of how they are trained but leave the specifics to the appendix.

4.1 Implementation

During pre-processing, we map input points to the corresponding target <IMAGE> token index, ViT patch index, and location index, and use those triples as additional input to the model. Our text input for points follows the Molmo2 [19] format, but replaces the string coordinates with the grounding tokens, including an additional <PATCH> token at the end of each list of points that is assigned the no-more-points class. This reduces the number of tokens per coordinate from 8 (6 digits and 2 spaces) to 3. For video, we also remove the text timestamps used by Molmo2 since they can be recovered based on which <IMAGE> token was selected, further reducing the token count.

As with Molmo2, we also give an integer object ID for each point, but place it after the coordinates instead of before. Following Molmo2, we use a separate learning rate and gradient norm for the new pointing parameters. In general, we set the learning rate to match that used for the image-text connector parameters. We set $M = 512$ for all experiments. In all training runs, we use packing and message-trees to support training on multiple examples per sequence [19].

4.2 Inference

During inference, we cache the keys of the image tokens and ViT patches during prefilling. This adds additional memory overhead, but the low-dimensionality of the keys means this uses roughly the same memory as the cached keys and values for 1-2 LLM layers, and it is only required for the image tokens.

We constrain the model to generate a <SUBPATCH> token and <LOCATION> token after each <PATCH> token, and to only select <IMAGE> tokens that are the same, or after, any <IMAGE> token it has already selected in the input sequence, so output points are ordered correctly. We also prevent the model from generating multiple points with the same <IMAGE> token and ViT subpatch since we observe that this is almost always a case of the model pointing to the same thing twice. If the model selects the no-more-points class, we constrain the model to generate the "> token, which ends a list of points in the Molmo2 pointing format.

To convert the selected patches back into coordinates, we retain a map of token_id $\rightarrow$ coordinates for every ViT patch during pre-processing and combine it with the location predictions to get the output point.

4.3 Models

MolmoPoint-8B. We conduct a full end-to-end training run following the pipeline of Molmo2-8B. To improve tracking, we also incorporate MolmoPoint-Track, a new dataset of human-annotated and synthetic tracks (see below). We also slightly adjust the training mixture to better exploit the improved learning efficiency of the pointing data, and make minor adjustments to training setup to better utilize the hardware we have available (see the appendix for details).

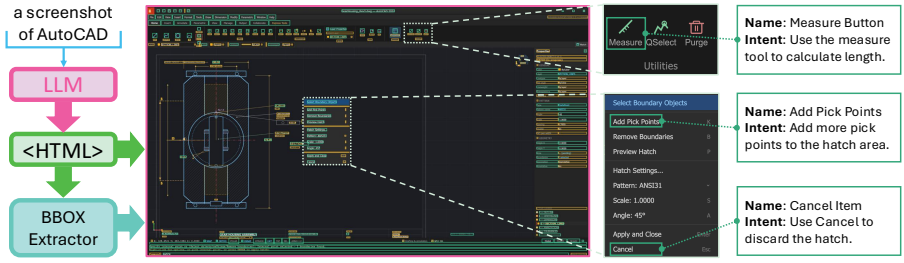

Fig. 3: Generating MolmoPoint-PointSyn. We prompt an LLM to generate the HTML for the screenshot and extract all bounding boxes of its UI elements. Then we use LLMs to annotate each bounding box with its interaction intents.

MolmoPoint-Img-8B. The image pointing data in the Molmo2 mixture does not contain many instructional/GUI examples. To train a model better optimized for this task, we build MolmoPoint-PointSyn, a synthetic GUI instructional dataset (see below for details), and fine-tune on it for 2000 steps with a batch size of 128 while increasing the image resolution to 48 crops per image.

MolmoPoint-Vid-4B. As in Molmo2, we find specialized models on video grounding can outperform generalist models. We therefore train a specialized video grounding model by training a 4B version of MolmoPoint with the same pre-training stage, but only fine-tuned on video-grounding data.

4.4 MolmoPoint-PointSyn

As shown in Figure 3, we extend the code-guided synthetic data generation framework (CoSyn) [89] to screenshot generation, in which we prompt the language model to generate HTML code that mimics digital environments for web, desktop, and mobile. Given access to the underlying HTML code in each screenshot, we use the Playwright library with custom JavaScript to automatically extract bounding boxes for all elements in the screenshot. We then feed the bounding box information to the language model to generate 5 pointing instructions per element that a user may ask when interacting with it. In total, we synthesize 36K screenshots, with 2M densely annotated points and over 10M pointing instructions. Qualitative examples are in Figure 8 in the appendix.

4.5 MolmoPoint-Track

We contribute MolmoPoint-Track, consisting of (1) **MolmoPoint-TrackAny**, human-annotated tracks on videos with diverse objects and (2) **MolmoPoint-TrackSyn**, synthetic tracks with diverse motion and occlusion patterns. For MolmoPoint-TrackAny, we extend Molmo2-VideoPoint annotations into full tracks via human annotation (Figure 4). For MolmoPoint-TrackSyn, we generate multi-object tracking videos in Blender with complex occlusion and motion dynamics, paired with automatically generated referring queries (Figure 7). See Appendix I for details and qualitative examples.

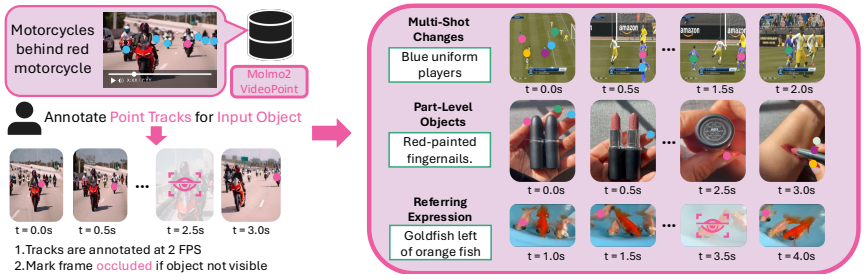

Fig. 4: MolmoPoint-TrackAny. Annotators are given a text query and an object of interest, and provide point tracks on frames where the object is visible.

Table 1: PixMo-Points results. MolmoPoint-8B is SoTA. We collect results for GPT5.2, Gemini3, and Qwen3 ourselves.

Model	R	P	F1
API call only			
GPT-5.2 [56]	31.0	32.9	31.6
Gemini3-Pro [28]	77.3	81.3	77.8
Open weights			
Qwen3-VL-8B [83]	54.3	53.5	53.4
Qwen3-VL-4B [83]	45.1	44.2	43.7
Fully open			
Molmo-7B-D [22]	76.4	76.2	75.7
Molmo-72B [22]	74.9	74.9	74.5
Molmo2-4B [19]	83.3	85.1	83.4
Molmo2-8B [19]	85.5	86.4	85.2
MolmoPoint			
MolmoPoint-8B	**90.4**	**89.3**	**89.2**

Table 2: Point-Bench results. Scores are from the Point-Bench leaderboard or [87].

Model	Avg
API call only	
Gemini-Robotics-ER-1.5 [1]	67.1
Gemini-2.5-Pro [20]	62.8
Open weights	
Poivre-7B [87]	67.5
Qwen3VL [83]	58.5
Qwen3-VL-235B [83]	58.3
Fully open	
VisionReasoner-7B [44]	64.7
Molmo-7B-D [22]	61.6
Molmo-72B [22]	63.8
Molmo2-4B [19]	67.7
Molmo2-8B [19]	68.7
MolmoPoint	
MolmoPoint-8B	**70.7**

5 Results

5.1 Image Pointing

We show results on natural image pointing in Table 15 and Table 1. MolmoPoint-8B is state-of-the-art on PointBench [16], surpassing Molmo2 by almost 2 points, including a 5 point gain in reasoning and spatial reasoning. On PixMo-Points [22] MolmoPoint-8B surpasses Molmo2 by 4 points. Molmo2 and MolmoPoint-8B used the same data and training procedure, so these results show that using grounding tokens significantly boosts pointing capabilities on natural images.

5.2 GUI Pointing

We show results on ScreenSpot-V2 [38], ScreenSpot-Pro [38], and OSWorldG [80]. In addition to other models, we also compare to a baseline, Molmo2-GUI-8B,

Table 3: GUI grounding results. 64crops denotes increasing the number of crops to 64 during inference. Scores with * are from evaluations in [58, 69].

Model	ScreenSpot-V2	ScreenSpot-Pro	OSWorldG
API call only			
Claude 3.7 [3]	87.6	27.7	-
OpenAI CUA [55]	87.9	23.4	-
Gemini-3-Pro [28]	93.7	72.7	35.5
Open weights			
Holo2-8B [21]	93.2	58.9	70.1
UI-TARS 1.5-7B [58]	94.2	61.6	64.2*
UI-Venus-1.5-8B [69]	95.9	68.4	69.7
Qwen3-VL-8B [6]	92.1*	52.7*	57.5*
MAI-UI-8B [100]	95.2	65.8	60.1
Fully open			
GUI-Actor [76]	90.9	41.8	-
JEDI-7B [81]	91.7	50.2	54.1
GroundCUA-7B [24]	89.3	50.2	67.2
OpenCUA-7B [74]	92.3	50.0	55.3
GTA1-7B [88]	92.4	50.1	60.1
Molmo2-8B [19]	89.5	30.4	54.1
Molmo2-GUI-8B	88.8	52.3	66.1
MolmoPoint			
MolmoPoint-8B	89.8	36.4	54.9
MolmoPoint-8B (64crops)	89.8	39.4	56.5
MolmoPoint-Img-8B	93.4	60.2	**70.0**
MolmoPoint-Img-8B (64crops)	**93.9**	**61.1**	**70.0**

built by fine-tuning Molmo2-8B on the same data MolmoPoint-Img-8B was trained on. The Molmo2 data mixture does not contain instruction-point pairs, so MolmoPoint-8B sometimes does not point when given them as input. To fix this, we use constrained decoding for both the Molmo2 models and MolmoPoint-8B (but not MolmoPoint-Img-8B) to force the model to generate exactly one point. We also show results when we increase the number of crops to 64 during inference. This breaks the ability of models with text coordinates to map patches to coordinates (*e.g.*, Molmo2-GUI-8B loses 30-60 points on high-res images, see the appendix), therefore we do not use it for other models.

Results are shown in Table 3. Compared to Molmo2, MolmoPoint-8B shows significant improvements on ScreenSpotPro and OSWorldG. Fine-tuning with instruction-image data makes MolmoPoint-Img-8B SoTA among fully open models on all tasks. Open-weight models UI-Venus and MAI-UI show better results, likely because both models utilize large-scale proprietary data collection efforts as well as more elaborate training pipelines that include RL.

We observe a gap of 2 to 9 points between MolmoPoint-Img-8B and our baseline, showing that our model design is critical for this high performance. We hypothesize the 9-point on ScreenSpotPro comes from grounding tokens being particularly important for high-resolution input.

Table 4: Video counting and pointing results. MolmoPoint-8B scores highest on BURST-VC and MolmoPoint-8B-VP and second highest on MolmoPoint-8B-VC's close accuracy, slightly behind Gemini 2.5 Pro. Best open model results are **bold**.

	BURST VC (test) [5]		Molmo2-VC		Molmo2-VP		
Model	Acc.	Close acc.	Acc.	Close acc.	F1	Recall	Precision
API call only							
GPT-5 [56]	43.1	73.7	35.8	50.3	4.1	4.4	4.2
GPT-5 mini [56]	46.0	73.0	29.8	49.3	2.2	2.2	2.2
Gemini 3 Pro [28]	44.0	71.7	37.1	53.1	20.0	27.4	19.8
Gemini 2.5 Pro [20]	41.6	70.0	35.8	56.5	13.0	14.5	13.6
Gemini 2.5 Flash [20]	38.7	70.0	31.9	48.2	11.1	11.2	12.2
Claude Sonnet 4.5 [4]	42.4	72.6	27.2	45.1	3.5	3.7	4.3
Open weights							
Qwen3-VL-4B [6]	38.9	74.7	25.3	44.3	0.0	0.0	0.0
Qwen3-VL-8B [6]	42.0	74.4	29.6	47.7	1.5	1.5	1.5
Fully open							
Molmo2-4B	61.5	76.1	34.3	56.1	**39.9**	**42.7**	**39.4**
Molmo2-8B	60.8	75.0	35.5	53.3	38.4	39.3	38.7
Molmo2-O-7B	61.6	76.0	33.2	50.5	35.8	35.8	37.9
MolmoPoint							
MolmoPoint-8B	61.6	**76.9**	35.6	54.6	36.2	35.7	37.8
MolmoPoint-Vid-4B	**62.0**	76.3	**36.0**	**58.7**	38.8	39.8	38.8

5.3 Video Pointing

We evaluate video pointing on BURST-VideoCount [5], Molmo2-VideoCount, and Molmo2-VideoPoint [19]. For the counting datasets, we report accuracy as well as close accuracy, which measures if the number of points is almost correct (computed as $|pred - gt| \leq \Delta$, where $\Delta = 1 + \lfloor 0.05 \times gt \rfloor$). For Molmo2-VideoPoint, we report F1, recall, and precision metrics when matching points to ground-truth segmentation masks. Baseline numbers come from Molmo2 [19].

For MolmoPoint-8B, we see an improvement on both Burst-VC and Molmo2-VC compared to Molmo2-8B, although we also see a drop in Molmo2-VP. To get a more definitive result, we conduct a human preference evaluation using predictions from both models on 470 video pointing queries (See appendix for details). We find 152 ties, 130 wins for Molmo2, and 188 wins for MolmoPoint-8B. Excluding ties, MolmoPoint-8B has a 59.1% win rate, showing humans prefer MolmoPoint-8B's output. MolmoPoint-Vid-4B sees a more consistent gain, including a full 5 point gain on Molmo2-VC close, surpassing Gemini 3 Pro.

5.4 Tracking Results

We evaluate on several tracking benchmarks and report Jaccard and F-measure ($\mathcal{J}\&\mathcal{F}$) and HOTA [48] following Molmo2 [19]. Results are in Table 12, F1 and per-category scores are in the appendix. MolmoPoint-8B shows substantial gains over Molmo2-8B. We see especially large gains on MeViS valid-u (+1.4 $\boldsymbol{\mathcal{J}\&\mathcal{F}}$),

Table 5: Tracking results. $\mathcal{J}\&\mathcal{F}$ evaluates segmentation masks generated using the points with Sam2, and HOTA [48] rewards consistent ID associations.

	MeViS [23]	MeViS [23]		Ref-YT-VOS [61]		Ref-Davis [34]		ReasonVOS [7]		Molmo2Track	
	valid	valid-u		valid		valid		test		overall	
Model	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	HOTA	$\mathcal{J}\&\mathcal{F}$	HOTA	$\mathcal{J}\&\mathcal{F}$	HOTA	$\mathcal{J}\&\mathcal{F}$	HOTA	$\mathcal{J}\&\mathcal{F}$	HOTA
API call only											
GPT-5 [56]	23.4	26.5	14.0	30.9	18.4	25.2	11.6	24.7	10.7	23.5	7.5
GPT-5 mini [56]	15.7	15.4	6.8	16.2	6.2	8.4	2.3	14.6	3.4	12.7	2.1
Gemini 3 Pro [28]	42.5	51.1	36.0	55.0	45.5	66.6	55.7	52.6	42.1	44.6	29.1
Gemini 2.5 Pro [20]	40.7	52.8	35.0	45.1	40.5	45.6	56.6	44.0	42.4	47.9	27.8
Gemini 2.5 Flash [20]	27.6	31.8	19.9	36.0	30.0	31.6	30.0	26.5	21.0	36.2	19.8
Open weights only											
Qwen3-VL-4B [83]	29.7	30.6	18.7	32.1	26.5	44.4	26.9	26.5	13.5	28.5	6.7
Qwen3-VL-8B [83]	35.1	34.4	23.8	48.3	37.6	41.0	33.2	24.9	17.5	28.7	16.5
Specialized open models											
VideoLISA [7]	44.4	53.2	–	63.7	–	68.8	–	47.5	–	43.3	–
VideoGLaMM [59]	45.2	50.6	–	66.8	–	69.5	–	33.9	–	37.9	–
Sa2VA-8B [92]	46.9	57.0	–	**70.7**	–	75.2	–	55.5	–	46.9	–
Sa2VA-Qwen3-VL-4B [92]	36.7	57.1	–	68.1	–	**76.0**	–	50.0	–	46.7	–
Fully open											
Molmo [22] + SAM 2 [60]	46.9	51.5	–	64.6	–	65.2	–	45.7	–	54.2	–
VideoMolmo-7B [2]	53.9	57.0	–	67.3	–	72.5	–	51.1	–	51.3	–
Molmo2-4B	63.3	70.0	72.4	70.2	78.8	73.5	81.1	61.9	64.0	56.7	57.6
Molmo2-8B	62.3	70.8	72.6	70.2	77.3	72.7	78.7	**65.8**	**68.6**	56.2	57.5
Molmo2-O-7B	58.4	69.7	72.3	67.9	76.1	70.4	76.0	62.6	65.1	53.7	54.2
MolmoPoint											
MolmoPoint-8B	**63.5**	**72.2**	**73.8**	70.5	**80.5**	73.6	**82.2**	64.7	67.0	**62.5**	**60.0**

which has more multi-object tracking queries, and Molmo2Track (+6.3 $\mathcal{J}\&\mathcal{F}$), which focuses on diverse in-the-wild videos. One exception is ReasonVOS [7], which targets queries that require semantic reasoning rather than precise spatial grounding, and therefore sees less benefit from grounding tokens.

Ablations show that removing grounding tokens leads to a **drop of 4.6 F1** and **4.0 HOTA** on Molmo2Track. Removing MolmoPoint-Track drops performance on both metrics by 1 point overall and by 7 points in the miscellaneous object category, showing it has significantly expanded the coverage of our training data. See the appendix for details.

5.5 MultiTask Results

To evaluate how well MolmoPoint-8B works as a general-purpose VLM, Table 6 reports average performance on 11 image benchmarks [8, 22, 30, 33, 47, 51–53, 63, 78, 95], 3 multi-image benchmarks [26, 54, 70], 6 short-video benchmarks [31, 39, 43, 57, 62, 79], and 7 long video benchmarks [25, 49, 50, 72, 75, 101] following the evaluation protocol Molmo2. Full details are in the appendix.

Overall, we see only small changes compared to Molmo2-8B: a small gain on images, a small loss on multi-images and short videos, and no difference on long videos. Since the data and training protocol are mostly unchanged between the two models, better transfer from image pointing data to image tasks might explain the gain in images. Some interference between grounding and QA might likewise explain the drop in short videos. We do observe another sign of positive transfer during pre-training, where captioning performance is slightly better for MolmoPoint-8B after pre-training (55.9 vs 55.6 F1 on the Molmo captioning metric [22]). However, in either case, the effect is small.

Table 6: Image and video QA results. Averaged results on the QA benchmarks in [19], see the appendix for details. Some results were computed by [19].

Model	Image Avg.	Multi-Image Avg.	Short Video Avg.	Long Video Avg.
API call only				
GPT-5 [56]	83.7	72.1	73.1	76.3
GPT-5 mini [56]	81.9	68.2	66.8	69.8
Gemini 3 Pro [28]	86.2	81.9	71.0	78.8
Gemini 2.5 Pro [20]	81.3	72.4	71.1	80.4
Gemini 2.5 Flash [20]	79.3	68.3	67.0	74.5
Claude Sonnet 4.5 [4]	76.3	59.5	62.8	66.4
Open weights				
InternVL3.5-4B [73]	77.2	53.5	62.0	56.5
InternVL3.5-8B [73]	78.2	54.9	63.0	57.1
Qwen3-VL-4B [6]	78.4	57.6	63.7	62.7
Qwen3-VL-8B [6]	81.2	56.3	65.3	63.5
Keye-VL-1.5-8B [85]	79.8	52.1	60.1	60.4
GLM-4.1V-9B [68]	77.0	**67.4**	64.2	60.5
MiniCPM-V-4.5-8B [91]	77.7	47.3	62.1	60.1
Eagle2.5-8B [13]	81.2	52.0	67.0	**65.2**
Fully open				
PLM-3B [18]	75.9	40.6	66.3	53.5
PLM-8B [18]	78.7	35.7	68.5	56.2
LLaVA-Video-7B [98]	-	-	59.4	56.2
VideoChat-Flash-7B [40]	-	-	66.4	58.1
Molmo2-4B [19]	80.4	57.8	69.3	64.5
Molmo2-8B [19]	81.7	56.4	**69.9**	64.1
Molmo2-O-7B [19]	79.7	53.5	68.1	59.2
MolmoPoint				
MolmoPoint-8B	**82.2**	56.0	69.5	64.2

Table 7: Ablations. Results when removing rotary embeddings, the no-more-points class, or the constraint that the points must be generated in sorted order.

Model	PixMoPoint F1	PointBench Avg.	Molmo2-VC Correct	Close	Overcount ↓
Molmo2-P-Ablation	85.2	67.8	58.0	36.6	3.6
w/o rotary	84.5	67.6	56.8	33.0	4.5
w/o no-more-points	84.7	66.6	52.3	32.8	10.3
w/o point sorting	83.6	71.2	40.0	24.4	3.2

5.6 Modeling Ablations

Modeling ablations are shown in Table 7. Due to limited compute resources, we do ablations with a lighter-weight training pipeline that starts from a pre-trained captioning model (without any pointing capabilities) and then fine-tunes on the

Fig. 5: Sample efficiency. Left: Performance when using a very limited number of pointing training examples. Right: Pointing performance during full-scale pre-training.

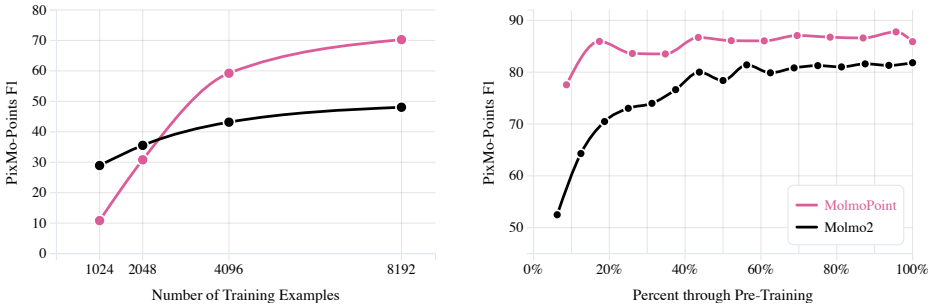

image and video pointing data from the Molmo2 data mixture. We train for 6000 steps with a batch size of 64. For video input, we observe that models can produce degenerate outputs with an excessive number of points. To capture this, we add an *overcount* metric, defined as how often predictions are $>$ 10 points and $>=$ twice the number of ground truth points.

Removing rotary embeddings decreases performance a bit on images and has a more significant effect on video. Removing the no-more-points token hurts performance and more than doubles the amount of overcounting. Randomizing the order of the points shows a significant drop for video, but a surprising gain on PointBench. We hypothesize that this might improve performance by allowing models to generate points in an "easiest point first" order [12], but more work will be needed to take advantage of this without degrading video.

5.7 Sample Efficiency

Figure 5 (left) shows performance when fine-tuning a base captioning model on a small number of pointing examples. Models were tuned for 6 epochs to ensure they were fully saturated. MolmoPoint-8B initially performs worse, likely due to needing to learn new parameters from scratch, but quickly improves to a 20 point gain when using 8192 examples (the full pointing dataset has close to half a million examples). Figure 5 (right) shows MolmoPoint reaches peak performance faster during pre-training. Both of these results show that grounding tokens make pointing easier and more efficient to learn than text coordinates.

5.8 Qualitative results

Despite being trained on the same data, we observe significant qualitative differences between MolmoPoint-8B and Molmo2-8B. MolmoPoint-8B is less likely to produce degenerate output on videos, is better at finding small objects, and is more precise when pointing. However, it occasionally produces off-by-one errors when counting high-frequency objects. See Figure 6 for qualitative examples.

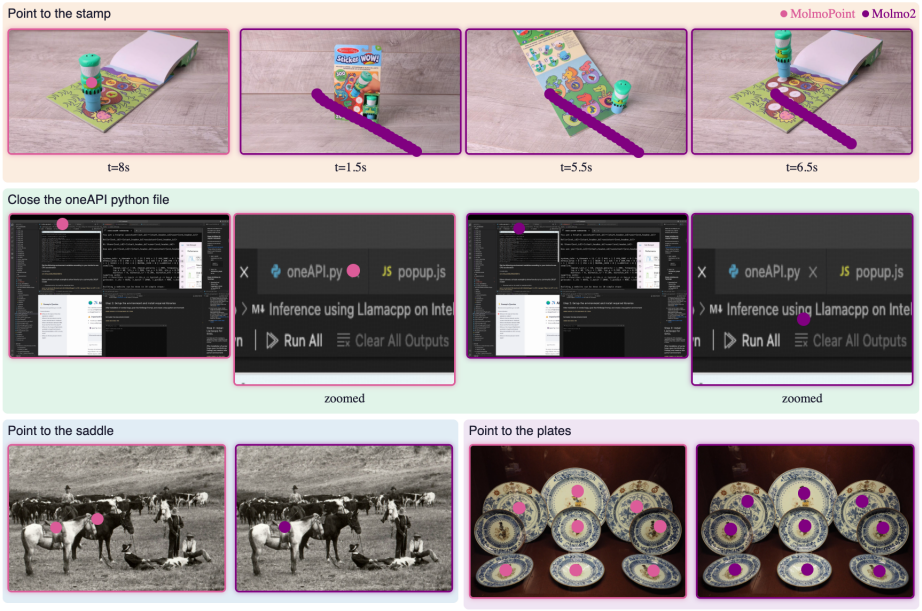

Fig. 6: Qualitative examples. Molmo2 generates lines of incorrect points in multiple video frames (top), is unable to localize the "x" exactly (middle), misses the second, partly occluded saddle (bottom left), but doesn't miss one of the plates (bottom right). The middle row shows results from MolmoPoint-Img-8B and Molmo2-GUI-8B, and the others show MolmoPoint-8B and Molmo2-8B.

6 Conclusion

We have shown that using grounding tokens significantly improves pointing across multiple domains. Future work could extend this approach to pointing to text tokens or audio tokens to enable pointing in additional modalities.

Acknowledgements

We thank David Albright, Cailin Brashear, Crystal Nam, Kyle Wiggers, and Will Smith for their important work for the MolmoPoint-8B public release. We also the Prolific team for their support and our annotators on Prolific for providing us with high-quality data that is crucial to MolmoPoint-8B. Finally, thanks to the other members of the PRIOR team for providing advice and feedback on various aspects of MolmoPoint-8B. This material is based upon work supported by the National Science Foundation under Award No. 2413244.

References

1. Abdolmaleki, A., Abeyruwan, S., Ainslie, J., Alayrac, J.B., Arenas, M.G., Balakrishna, A., Batchelor, N., Bewley, A., Bingham, J., Bloesch, M., et al.: Gemini robotics 1.5: Pushing the frontier of generalist robots with advanced embodied reasoning, thinking, and motion transfer. arXiv preprint arXiv:2510.03342 (2025)
2. Ahmad, G.S., Heakl, A., Gani, H., Shaker, A., Shen, Z., Khan, F.S., Khan, S.: Videomolmo: Spatio-temporal grounding meets pointing. arXiv preprint arXiv:2506.05336 (2025)
3. Anthropic: The claude 3 model family: Opus, sonnet, haiku (2024), https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf
4. Anthropic: Claude sonnet 4.5 system card (2025), https://assets.anthropic.com/m/12f214efcc2f457a/original/Claude-Sonnet-4-5-System-Card.pdf
5. Athar, A., Luiten, J., Voigtlaender, P., Khurana, T., Dave, A., Leibe, B., Ramanan, D.: Burst: A benchmark for unifying object recognition, segmentation and tracking in video. In: WACV (2023)
6. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., Ge, W., Guo, Z., Huang, Q., Huang, J., Huang, F., Hui, B., Jiang, S., Li, Z., Li, M., Li, M., Li, K., Lin, Z., Lin, J., Liu, X., Liu, J., Liu, C., Liu, Y., Liu, D., Liu, S., Lu, D., Luo, R., Lv, C., Men, R., Meng, L., Ren, X., Ren, X., Song, S., Sun, Y., Tang, J., Tu, J., Wan, J., Wang, P., Wang, P., Wang, Q., Wang, Y., Xie, T., Xu, Y., Xu, H., Xu, J., Yang, Z., Yang, M., Yang, J., Yang, A., Yu, B., Zhang, F., Zhang, H., Zhang, X., Zheng, B., Zhong, H., Zhou, J., Zhou, F., Zhou, J., Zhu, Y., Zhu, K.: Qwen3-vl technical report. arXiv preprint arXiv:2511.21631 (2025)
7. Bai, Z., He, T., Mei, H., Wang, P., Gao, Z., Chen, J., Liu, L., Zhang, Z., Shou, M.Z.: One token to seg them all: Language instructed reasoning segmentation in videos. In: NeurIPS (2024)
8. Beyer, L., Steiner, A., Pinto, A.S., Kolesnikov, A., Wang, X., Salz, D., Neumann, M., Alabdulmohsin, I., Tschannen, M., Bugliarello, E., Unterthiner, T., Keysers, D., Koppula, S., Liu, F., Grycner, A., Gritsenko, A., Houlsby, N., Kumar, M., Rong, K., Eisenschlos, J., Kabra, R., Bauer, M., Bošnjak, M., Chen, X., Minderer, M., Voigtlaender, P., Bica, I., Balazevic, I., Puigcerver, J., Papalampidi, P., Henaff, O., Xiong, X., Soricut, R., Harmsen, J., Zhai, X.: PaliGemma: A versatile 3B VLM for transfer. arXiv preprint arXiv:2407.07726 (2024)
9. Bigverdi, M., Luo, Z., Hsieh, C.Y., Shen, E., Chen, D., Shapiro, L.G., Krishna, R.: Perception tokens enhance visual reasoning in multimodal language models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 3836–3845 (2025)
10. Carion, N., Gustafson, L., Hu, Y.T., Debnath, S., Hu, R., Suris, D., Ryali, C., Alwala, K.V., Khedr, H., Huang, A., et al.: Sam 3: Segment anything with concepts. arXiv preprint arXiv:2511.16719 (2025)
11. Chai, Y., Huang, S., Niu, Y., Xiao, H., Liu, L., Zhang, D., Gao, P., Ren, S., Li, H.: Amex: Android multi-annotation expo dataset for mobile gui agents (2024), https://arxiv.org/abs/2407.17490
12. Chang, H., Zhang, H., Jiang, L., Liu, C., Freeman, W.T.: Maskgit: Masked generative image transformer. In: CVPR (2022)
13. Chen, G., Li, Z., Wang, S., Jiang, J., Liu, Y., Lu, L., Huang, D.A., Byeon, W., Le, M., Ehrlich, M., Lu, T., Wang, L., Catanzaro, B., Kautz, J., Tao, A., Yu, Z.,

Liu, G.: Eagle 2.5: Boosting long-context post-training for frontier vision-language models. In: NeurIPS (2025)
14. Chen, T., Saxena, S., Li, L., Fleet, D.J., Hinton, G.: Pix2seq: A language modeling framework for object detection. arXiv preprint arXiv:2109.10852 (2021)
15. Cheng, K., Sun, Q., Chu, Y., Xu, F., YanTao, L., Zhang, J., Wu, Z.: Seeclick: Harnessing gui grounding for advanced visual gui agents. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 9313–9332 (2024)
16. Cheng, L., Duan, J., Wang, Y.R., Fang, H., Li, B., Huang, Y., Wang, E., Eftekhar, A., Lee, J., Yuan, W., et al.: Pointarena: Probing multimodal grounding through language-guided pointing. arXiv preprint arXiv:2505.09990 (2025)
17. Chiang, W.L., Zheng, L., Sheng, Y., Angelopoulos, A.N., Li, T., Li, D., Zhang, H., Zhu, B., Jordan, M., Gonzalez, J.E., Stoica, I.: Chatbot arena: An open platform for evaluating LLMs by human preference. In: ICML (2024)
18. Cho, J.H., Madotto, A., Mavroudi, E., Afouras, T., Nagarajan, T., Maaz, M., Song, Y., Ma, T., Hu, S., Rasheed, H., Sun, P., Huang, P.Y., Bolya, D., Jain, S., Martin, M., Wang, H., Ravi, N., Jain, S., Stark, T., Moon, S., Damavandi, B., Lee, V., Westbury, A., Khan, S., Krähenbühl, P., Dollár, P., Torresani, L., Grauman, K., Feichtenhofer, C.: Perceptionlm: Open-access data and models for detailed visual understanding. arXiv preprint arXiv:2504.13180 (2025)
19. Clark, C., Zhang, J., Ma, Z., Park, J.S., Salehi, M., Tripathi, R., Lee, S., Ren, Z., Kim, C.D., Yang, Y., et al.: Molmo2: Open weights and data for vision-language models with video understanding and grounding. arXiv preprint arXiv:2601.10611 (2026)
20. Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., et al.: Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261 (2025)
21. Company, H.: Holo2 - open foundation models for navigation and computer use agents (2025), https://huggingface.co/collections/Hcompany/holo2
22. Deitke, M., Clark, C., Lee, S., Tripathi, R., Yang, Y., Park, J.S., Salehi, M., Muennighoff, N., Lo, K., Soldaini, L., Lu, J., Anderson, T., Bransom, E., Ehsani, K., Ngo, H., Chen, Y., Patel, A., Yatskar, M., Callison-Burch, C., Head, A., Hendrix, R., Bastani, F., VanderBilt, E., Lambert, N., Chou, Y., Chheda, A., Sparks, J., Skjonsberg, S., Schmitz, M., Sarnat, A., Bischoff, B., Walsh, P., Newell, C., Wolters, P., Gupta, T., Zeng, K.H., Borchardt, J., Groeneveld, D., Nam, C., Lebrecht, S., Wittlif, C., Schoenick, C., Michel, O., Krishna, R., Weihs, L., Smith, N.A., Hajishirzi, H., Girshick, R., Farhadi, A., Kembhavi, A.: Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In: CVPR (2025)
23. Ding, H., Liu, C., He, S., Jiang, X., Loy, C.C.: Mevis: A large-scale benchmark for video segmentation with motion expressions. In: ICCV (2023)
24. Feizi, A., Nayak, S., Jian, X., Lin, K.Q., Li, K., Awal, R., Lù, X.H., Obando-Ceron, J., Rodriguez, J.A., Chapados, N., Vazquez, D., Romero-Soriano, A., Rabbany, R., Taslakian, P., Pal, C., Gella, S., Rajeswar, S.: Grounding computer use agents on human demonstrations. arXiv preprint arXiv:2511.07332 (2025)
25. Fu, C., Dai, Y., Luo, Y., Li, L., Ren, S., Zhang, R., Wang, Z., Zhou, C., Shen, Y., Zhang, M., et al.: Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In: CVPR (2025)

26. Fu, X., Hu, Y., Li, B., Feng, Y., Wang, H., Lin, X., Roth, D., Smith, N.A., Ma, W.C., Krishna, R.: Blink: Multimodal large language models can see but not perceive. In: ECCV (2024)
27. Ge, T., Chan, X., Wang, X., Yu, D., Mi, H., Yu, D.: Scaling synthetic data creation with 1,000,000,000 personas. arXiv preprint arXiv:2406.20094 (2024)
28. Google: Gemini 3 Pro model card (2025), https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Pro-Model-Card.pdf
29. Gou, B., Wang, R., Zheng, B., Xie, Y., Chang, C., Shu, Y., Sun, H., Su, Y.: Navigating the digital world as humans do: Universal visual grounding for GUI agents. In: The Thirteenth International Conference on Learning Representations (2025), https://openreview.net/forum?id=kxnoqaisCT
30. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In: CVPR (2017)
31. Hong, W., Cheng, Y., Yang, Z., Wang, W., Wang, L., Gu, X., Huang, S., Dong, Y., Tang, J.: Motionbench: Benchmarking and improving fine-grained video motion understanding for vision language models. In: CVPR (2025)
32. Kapoor, R., Butala, Y.P., Russak, M., Koh, J.Y., Kamble, K., AlShikh, W., Salakhutdinov, R.: Omniact: A dataset and benchmark for enabling multimodal generalist autonomous agents for desktop and web. In: European Conference on Computer Vision. pp. 161–178. Springer (2024)
33. Kembhavi, A., Salvato, M., Kolve, E., Seo, M., Hajishirzi, H., Farhadi, A.: A diagram is worth a dozen images. In: ECCV (2016)
34. Khoreva, A., Rohrbach, A., Schiele, B.: Video object segmentation with language referring expressions. In: ACCV (2018)
35. Lai, X., Tian, Z., Chen, Y., Li, Y., Yuan, Y., Liu, S., Jia, J.: Lisa: Reasoning segmentation via large language model. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9579–9589 (2024)
36. Lee, J., Duan, J., Fang, H., Deng, Y., Liu, S., Li, B., Fang, B., Zhang, J., Wang, Y.R., Lee, S., Han, W., Pumacay, W., Wu, A., Hendrix, R., Farley, K., VanderBilt, E., Farhadi, A., Fox, D., Krishna, R.: Molmoact: Action reasoning models that can reason in space. arXiv preprint arXiv:2508.07917 (2025)
37. Lee, J., Duan, J., Fang, H., Deng, Y., Liu, S., Li, B., Fang, B., Zhang, J., Wang, Y.R., Lee, S., et al.: Molmoact: Action reasoning models that can reason in space. arXiv preprint arXiv:2508.07917 (2025)
38. Li, K., Meng, Z., Lin, H., Luo, Z., Tian, Y., Ma, J., Huang, Z., Chua, T.S.: Screenspot-pro: Gui grounding for professional high-resolution computer use. In: MM (2025)
39. Li, K., Wang, Y., He, Y., Li, Y., Wang, Y., Liu, Y., Wang, Z., Xu, J., Chen, G., Luo, P., et al.: Mvbench: A comprehensive multi-modal video understanding benchmark. In: CVPR (2024)
40. Li, X., Wang, Y., Yu, J., Zeng, X., Zhu, Y., Huang, H., Gao, J., Li, K., He, Y., Wang, C., Qiao, Y., Wang, Y., Wang, L.: Videochat-flash: Hierarchical compression for long-context video modeling. arXiv preprint arXiv:2501.00574 (2024)
41. Li, Y., Zhang, J., Teng, X., Zhang, H., Liu, X., Lan, L.: Refsam: Efficiently adapting segmenting anything model for referring video object segmentation. Neural Networks (2025)
42. Lin, K.Q., Li, L., Gao, D., Yang, Z., Wu, S., Bai, Z., Lei, W., Wang, L., Shou, M.Z.: Showui: One vision-language-action model for gui visual agent (2024), https://arxiv.org/abs/2411.17465

43. Liu, Y., Li, S., Liu, Y., Wang, Y., Ren, S., Li, L., Chen, S., Sun, X., Hou, L.: Tempcompass: Do video llms really understand videos? In: ACL (2024)
44. Liu, Y., Qu, T., Zhong, Z., Peng, B., Liu, S., Yu, B., Jia, J.: Visionreasoner: Unified visual perception and reasoning via reinforcement learning. arXiv preprint arXiv:2505.12081 (2025)
45. Lu, J., Clark, C., Lee, S., Zhang, Z., Khosla, S., Marten, R., Hoiem, D., Kembhavi, A.: Unified-io 2: Scaling autoregressive multimodal models with vision language audio and action. In: CVPR (2024)
46. Lu, J., Clark, C., Zellers, R., Mottaghi, R., Kembhavi, A.: Unified-io: A unified model for vision, language, and multi-modal tasks. arXiv preprint arXiv:2206.08916 (2022)
47. Lu, P., Bansal, H., Xia, T., Liu, J., Li, C., Hajishirzi, H., Cheng, H., Chang, K.W., Galley, M., Gao, J.: MathVista: Evaluating mathematical reasoning of foundation models in visual contexts. In: ICLR (2024)
48. Luiten, J., Osep, A., Dendorfer, P., Torr, P., Geiger, A., Leal-Taixé, L., Leibe, B.: Hota: A higher order metric for evaluating multi-object tracking. IJCV (2021)
49. Ma, W., Ren, W., Jia, Y., Li, Z., Nie, P., Zhang, G., Chen, W.: Videoeval-pro: Robust and realistic long video understanding evaluation. arXiv preprint arXiv:2505.14640 (2025)
50. Mangalam, K., Akshulakov, R., Malik, J.: Egoschema: A diagnostic benchmark for very long-form video language understanding. In: NeurIPS Track on Datasets and Benchmarks (2023)
51. Masry, A., Long, D., Tan, J.Q., Joty, S., Hoque, E.: ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In: ACL (2022)
52. Mathew, M., Bagal, V., Tito, R., Karatzas, D., Valveny, E., Jawahar, C.: InfographicVQA. In: WACV (2022)
53. Mathew, M., Karatzas, D., Jawahar, C.: DocVQA: A dataset for VQA on document images. In: WACV (2021)
54. Meng, F., Wang, J., Li, C., Lu, Q., Tian, H., Liao, J., Zhu, X., Dai, J., Qiao, Y., Luo, P., Zhang, K., Shao, W.: Mmiu: Multimodal multi-image understanding for evaluating large vision-language models. In: ICLR (2025)
55. OpenAI: Collaborative user agreement (2024), https://openai.com/policies/service-terms/, accessed: 2025-03-01
56. OpenAI: GPT-5 system card (2025), https://openai.com/index/gpt-5-system-card/
57. Patraucean, V., Smaira, L., Gupta, A., Recasens, A., Markeeva, L., Banarse, D., Koppula, S., Malinowski, M., Yang, Y., Doersch, C., et al.: Perception test: A diagnostic benchmark for multimodal video models. NeurIPS (2023)
58. Qin, Y., Ye, Y., Fang, J., Wang, H., Liang, S., Tian, S., Zhang, J., Li, J., Li, Y., Huang, S., et al.: Ui-tars: Pioneering automated gui interaction with native agents. arXiv preprint arXiv:2501.12326 (2025)
59. Rasheed, H., Maaz, M., Shaji, S., Shaker, A., Khan, S., Cholakkal, H., Anwer, R.M., Xing, E., Yang, M.H., Khan, F.S.: Glamm: Pixel grounding large multimodal model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13009–13018 (2024)
60. Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K.V., Carion, N., Wu, C.Y., Girshick, R., Dollár, P., Feichtenhofer, C.: Sam 2: Segment anything in images and videos. In: ICLR (2025)
61. Seo, S., Lee, J.Y., Han, B.: Urvos: Unified referring video object segmentation network with a large-scale benchmark. In: ECCV (2020)

62. Shangguan, Z., Li, C., Ding, Y., Zheng, Y., Zhao, Y., Fitzgerald, T., Cohan, A.: Tomato: Assessing visual temporal reasoning capabilities in multimodal foundation models. In: ICLR (2025)
63. Singh, A., Natarjan, V., Shah, M., Jiang, Y., Chen, X., Parikh, D., Rohrbach, M.: Towards VQA models that can read. In: CVPR (2019)
64. Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y.: Roformer: Enhanced transformer with rotary position embedding. Neurocomputing (2024)
65. Su, Y., Zhang, H., Li, S., Liu, N., Liao, J., Pan, J., Liu, Y., Xing, X., Sun, C., Li, C., et al.: Patch-as-decodable-token: Towards unified multi-modal vision tasks in mllms. arXiv preprint arXiv:2510.01954 (2025)
66. Sun, Q., Hong, P., Pala, T.D., Toh, V., Tan, U.X., Ghosal, D., Poria, S.: Emma-x: An embodied multimodal action model with grounded chain of thought and look-ahead spatial reasoning. In: ACL (2025)
67. Tang, J., Xia, Y., Wu, Y.F., Hu, Y., Chen, Y., Chen, Q., Xu, X., Wu, X., Lu, H., Ma, Y., Lu, S., Chen, Q.: Lpo: Towards accurate gui agent interaction via location preference optimization. ArXiv **abs/2506.09373** (2025), https://api.semanticscholar.org/CorpusID:279306118
68. Team, V., Hong, W., Yu, W., Gu, X., Wang, G., Gan, G., Tang, H., Cheng, J., Qi, J., Ji, J., Pan, L., Duan, S., Wang, W., Wang, Y., Cheng, Y., He, Z., Su, Z., Yang, Z., Pan, Z., Zeng, A., Wang, B., Chen, B., Shi, B., Pang, C., Zhang, C., Yin, D., Yang, F., Chen, G., Xu, J., Zhu, J., Chen, J., Chen, J., Chen, J., Lin, J., Wang, J., Chen, J., Lei, L., Gong, L., Pan, L., Liu, M., Xu, M., Zhang, M., Zheng, Q., Yang, S., Zhong, S., Huang, S., Zhao, S., Xue, S., Tu, S., Meng, S., Zhang, T., Luo, T., Hao, T., Tong, T., Li, W., Jia, W., Liu, X., Zhang, X., Lyu, X., Fan, X., Huang, X., Wang, Y., Xue, Y., Wang, Y., Wang, Y., An, Y., Du, Y., Shi, Y., Huang, Y., Niu, Y., Wang, Y., Yue, Y., Li, Y., Zhang, Y., Wang, Y., Wang, Y., Zhang, Y., Xue, Z., Hou, Z., Du, Z., Wang, Z., Zhang, P., Liu, D., Xu, B., Li, J., Huang, M., Dong, Y., Tang, J.: Glm-4.5v and glm-4.1v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. arXiv preprint arXiv:2507.01006 (2025)
69. Team, V., Gao, C., Gu, Z., Liu, Y., Qiu, X., Shen, S., Wen, Y., Xia, T., Xu, Z., Zeng, Z., Zhou, B., Zhou, X., Chen, W., Dai, S., Dou, J., Gong, Y., Guo, Y., Guo, Z., Li, F., Li, Q., Lin, J., Zhou, Y., Zhu, L., Chen, L., Guo, Z., Meng, C., Wang, W.: Ui-venus-1.5 technical report. arXiv preprint arXiv:2602.09082 (2026)
70. Wang, F., Fu, X., Huang, J.Y., Li, Z., Liu, Q., Liu, X., Ma, M.D., Xu, N., Zhou, W., Zhang, K., et al.: Muirbench: A comprehensive benchmark for robust multi-image understanding. In: ICLR (2025)
71. Wang, P., Yang, A., Men, R., Lin, J., Bai, S., Li, Z., Ma, J., Zhou, C., Zhou, J., Yang, H.: Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In: International conference on machine learning. pp. 23318–23340. PMLR (2022)
72. Wang, W., He, Z., Hong, W., Cheng, Y., Zhang, X., Qi, J., Ding, M., Gu, X., Huang, S., Xu, B., et al.: Lvbench: An extreme long video understanding benchmark. In: ICCV (2025)
73. Wang, W., Gao, Z., Gu, L., Pu, H., Cui, L., Wei, X., Liu, Z., Jing, L., Ye, S., Shao, J., et al.: Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. arXiv preprint arXiv:2508.18265 (2025)
74. Wang, X., Wang, B., Lu, D., Yang, J., Xie, T., Wang, J., Deng, J., Guo, X., Xu, Y., Wu, C.H., Shen, Z., Li, Z., Li, R., Li, X., Chen, J., Zheng, B., Li, P., Lei, F., Cao, R., Fu, Y., Shin, D., Shin, M., Hu, J., Wang, Y., Chen, J., Ye, Y., Zhang, D.,

Du, D., Hu, H., Chen, H., Zhou, Z., Yao, H., Chen, Z., Gu, Q., Wang, Y., Wang, H., Yang, D., Zhong, V., Sung, F., Charles, Y., Yang, Z., Yu, T.: Opencua: Open foundations for computer-use agents. arXiv preprint arXiv:2508.09123 (2025)
75. Wu, H., Li, D., Chen, B., Li, J.: Longvideobench: A benchmark for long-context interleaved video-language understanding. In: NeurIPS (2024)
76. Wu, Q., Cheng, K., Yang, R., Zhang, C., Yang, J., Jiang, H., Mu, J., Peng, B., Qiao, B., Tan, R., et al.: Gui-actor: Coordinate-free visual grounding for gui agents. arXiv preprint arXiv:2506.03143 (2025)
77. Wu, Z., Wu, Z., Xu, F., Wang, Y., Sun, Q., Jia, C., Cheng, K., Ding, Z., Chen, L., Liang, P.P., et al.: Os-atlas: A foundation action model for generalist gui agents. arXiv preprint arXiv:2410.23218 (2024)
78. xAI: RealWorldQA. https://huggingface.co/datasets/xai-org/RealworldQA (2024), accessed: 2024-09-24
79. Xiao, J., Shang, X., Yao, A., Chua, T.S.: Next-qa: Next phase of question-answering to explaining temporal actions. In: CVPR (2021)
80. Xie, T., Deng, J., Li, X., Yang, J., Wu, H., Chen, J., Hu, W., Wang, X., Xu, Y., Wang, Z., Xu, Y., Wang, J., Sahoo, D., Yu, T., Xiong, C.: Scaling computer-use grounding via user interface decomposition and synthesis. arXiv preprint arXiv:2505.13227 (2025)
81. Xie, T., Deng, J., Li, X., Yang, J., Wu, H., Chen, J., Hu, W., Wang, X., Xu, Y., Wang, Z., Xu, Y., Wang, J., Sahoo, D., Yu, T., Xiong, C.: Scaling computer-use grounding via user interface decomposition and synthesis. arXiv preprint arXiv:2505.13227 (2025)
82. Yan, C., Wang, H., Yan, S., Jiang, X., Hu, Y., Kang, G., Xie, W., Gavves, E.: Visa: Reasoning video object segmentation via large language models. In: ECCV (2024)
83. Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Zhou, J., Lin, J., Dang, K., Bao, K., Yang, K., Yu, L., Deng, L., Li, M., Xue, M., Li, M., Zhang, P., Wang, P., Zhu, Q., Men, R., Gao, R., Liu, S., Luo, S., Li, T., Tang, T., Yin, W., Ren, X., Wang, X., Zhang, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Zhang, Y., Wan, Y., Liu, Y., Wang, Z., Cui, Z., Zhang, Z., Zhou, Z., Qiu, Z.: Qwen3 technical report. arXiv preprint arXiv:2505.09388 (2025)
84. Yang, A., Yang, B., Hui, B., Zheng, B., Yu, B., Zhou, C., Li, C., Li, C., Liu, D., Huang, F., Dong, G., Wei, H., Lin, H., Tang, J., Wang, J., Yang, J., Tu, J., Zhang, J., Ma, J., Xu, J., Zhou, J., Bai, J., He, J., Lin, J., Dang, K., Lu, K., Chen, K., Yang, K., Li, M., Xue, M., Ni, N., Zhang, P., Wang, P., Peng, R., Men, R., Gao, R., Lin, R., Wang, S., Bai, S., Tan, S., Zhu, T., Li, T., Liu, T., Ge, W., Deng, X., Zhou, X., Ren, X., Zhang, X., Wei, X., Ren, X., Fan, Y., Yao, Y., Zhang, Y., Wan, Y., Chu, Y., Liu, Y., Cui, Z., Zhang, Z., Fan, Z.: Qwen2 technical report. arXiv preprint arXiv:2407.10671 (2024)
85. Yang, B., Wen, B., Ding, B., Liu, C., Chu, C., Song, C., Rao, C., Yi, C., Li, D., Zang, D., et al.: Kwai keye-vl 1.5 technical report. arXiv preprint arXiv:2509.01563 (2025)
86. Yang, J., Fu, C.K., Shah, D., Sadigh, D., Xia, F., Zhang, T.: Bridging perception and action: Spatially-grounded mid-level representations for robot generalization. arXiv preprint arXiv:2506.06196 (2025)
87. Yang, W., Huang, Z.: Poivre: Self-refining visual pointing with reinforcement learning. arXiv preprint arXiv:2509.23746 (2025)

88. Yang, Y., Li, D., Yang, Y., Luo, Z., Dai, Y., Chen, Z., Xu, R., Pan, L., Xiong, C., Li, J.: Grpo for gui grounding done right (2025), https://huggingface.co/HelloKKMe/GTA1-7B
89. Yang, Y., Patel, A., Deitke, M., Gupta, T., Weihs, L., Head, A., Yatskar, M., Callison-Burch, C., Krishna, R., Kembhavi, A., et al.: Scaling text-rich image understanding via code-guided synthetic multimodal data generation. In: ACL (2025)
90. Yang, Y., Wang, Y., Li, D., Luo, Z., Chen, B., Huang, C., Li, J.: Aria-ui: Visual grounding for gui instructions. arXiv preprint arXiv:2412.16256 (2024)
91. Yu, T., Wang, Z., Wang, C., Huang, F., Ma, W., He, Z., Cai, T., Chen, W., Huang, Y., Zhao, Y., Xu, B., Cui, J., Xu, Y., Ruan, L., Zhang, L., Liu, H., Tang, J., Liu, H., Guo, Q., Hu, W., He, B., Zhou, J., Cai, J., Qi, J., Guo, Z., Chen, C., Zeng, G., Li, Y., Cui, G., Ding, N., Han, X., Yao, Y., Liu, Z., Sun, M.: Minicpm-v 4.5: Cooking efficient mllms via architecture, data, and training recipe. arXiv preprint arXiv:2509.18154 (2025)
92. Yuan, H., Li, X., Zhang, T., Huang, Z., Xu, S., Ji, S., Tong, Y., Qi, L., Feng, J., Yang, M.H.: Sa2va: Marrying sam2 with llava for dense grounded understanding of images and videos. arXiv preprint arXiv:2501.04001 (2025)
93. Yuan, W., Duan, J., Blukis, V., Pumacay, W., Krishna, R., Murali, A., Mousavian, A., Fox, D.: Robopoint: A vision-language model for spatial affordance prediction for robotics. In: CoRL (2024)
94. Yuan, X., Zhang, J., Li, K., Cai, Z., Yao, L., Chen, J., Wang, E., Hou, Q., Chen, J., Jiang, P.T., Li, B.: Enhancing visual grounding for gui agents via self-evolutionary reinforcement learning. ArXiv **abs/2505.12370** (2025), https://api.semanticscholar.org/CorpusID:278739769
95. Yue, X., Ni, Y., Zhang, K., Zheng, T., Liu, R., Zhang, G., Stevens, S., Jiang, D., Ren, W., Sun, Y., Wei, C., Yu, B., Yuan, R., Sun, R., Yin, M., Zheng, B., Yang, Z., Liu, Y., Huang, W., Sun, H., Su, Y., Chen, W.: MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In: CVPR (2024)
96. Zhang, B., Zhang, B., Wang, B., Zheng, W., Cheng, Y., Tang, L., Yan, Y., Zhou, J., Lu, J.: Manicog: Training-free improvement for gui grounding via manipulation chains (2026)
97. Zhang, Y., Zhang, L., Ma, R., Cao, N.: Texverse: A universe of 3d objects with high-resolution textures. arXiv preprint arXiv:2508.10868 (2025)
98. Zhang, Y., Wu, J., Li, W., Li, B., Ma, Z., Liu, Z., Li, C.: Llava-video: Video instruction tuning with synthetic data. TMLR (2025)
99. Zhao, Y., Gu, A., Varma, R., Luo, L., Huang, C.C., Xu, M., Wright, L., Shojanazeri, H., Ott, M., Shleifer, S., et al.: Pytorch fsdp: Experiences on scaling fully sharded data parallel. arXiv preprint arXiv:2304.11277 (2023)
100. Zhou, H., Zhang, X., Tong, P., Zhang, J., Chen, L., Kong, Q., Cai, C., Liu, C., Wang, Y., Zhou, J., Hoi, S.: Mai-ui technical report: Real-world centric foundation gui agents. arXiv preprint arXiv:2512.22047 (2025)
101. Zhou, J., Shu, Y., Zhao, B., Wu, B., Liang, Z., Xiao, S., Qin, M., Yang, X., Xiong, Y., Zhang, B., et al.: Mlvu: Benchmarking multi-task long video understanding. In: CVPR (2025)