2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

MolmoAct: Action Reasoning Models that can Reason in Space

Jason Lee[†1,2*], Jiafei Duan[†1,2*], Haoquan Fang[†1,2*]
Yuquan Deng[†1], Shuo Liu[†1,2], Boyang Li[†2], Bohan Fang[†2], Jieyu Zhang[†1,2], Yi Ru Wang[†1,2]
Sangho Lee[1], Winson Han[1], Wilbert Pumacay[1], Angelica Wu[2], Rose Hendrix[†1], Karen Farley[1], Eli Vanderbilt[1]
Ali Farhadi[1,2], Dieter Fox[†1,2], Ranjay Krishna[†1,2]

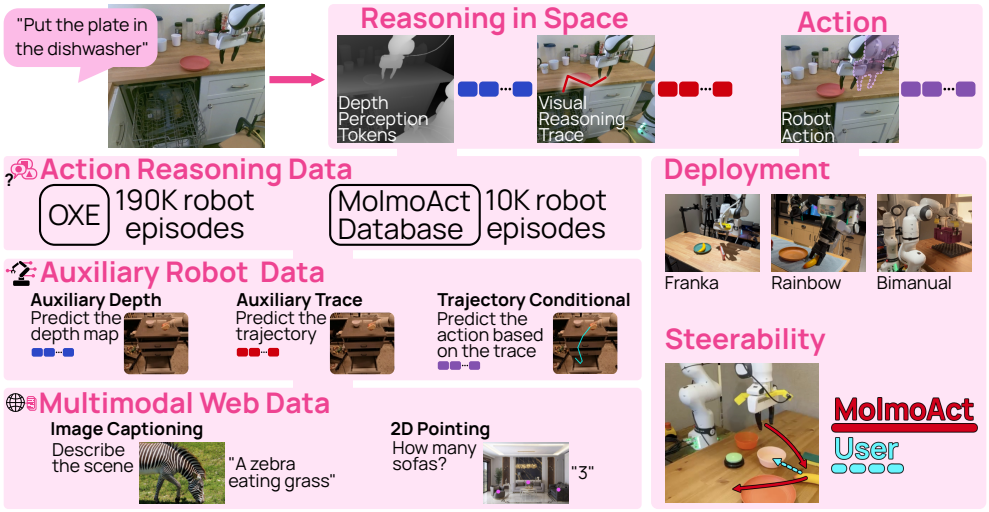

Fig. 1: **Overview**. MOLMOACT is an open Action Reasoning Model (ARM) that reasons in space. Given a language instruction, it autoregressively predicts three structured chains: Depth Perception Tokens to perceive the environment, Visual Reasoning Trace to plan its actions, and Action Tokens as robot control commands. In other words, MOLMOACT generates a latent 2.5D depth map of the scene, a 2D trajectory sketch of its plans, and finally, spatially grounded actions commands.

***Abstract*— Reasoning is essential for purposeful action, yet most robotic foundation models map perception and instructions directly to control, limiting adaptability, generalization, and semantic grounding. We introduce Action Reasoning Models (ARMs), which integrate perception, planning, and control through a structured three-stage pipeline. Our model, MOLMOACT, encodes observations and instructions into depth perception tokens, generates 2D spatial plans, and predicts fine-grained actions, enabling explainable and steerable behavior. MOLMOACT-7B-D achieves 70.5% zero-shot accuracy on SimplerEnv Visual Matching (surpassing π_0 and GR00T N1.5), 86.6% average success on LIBERO, and real-world fine-tuning gains of +10% (single-arm) and +22.7% (bimanual) over π_0-FAST. It further improves out-of-distribution generalization by +23.3% and ranks highest in human-preference evaluations for open-instruction following and trajectory steering. We also release MOLMOACT DATASET, a dataset of 10k diverse robot trajectories that yields an average +5.5% performance boost when used for training. Together with open model weights and code, this establishes MOLMOACT as a state-of-the-art robotic foundation model and an open blueprint for building ARMs that transform perception into grounded, purposeful action. Further experimental details and result with MOLMOACT DATASET and human-preference evaluations included in supplementary video.**

[1]Allen Institute for AI, Seattle WA [2]University of Washington, Seattle WA
[*]Equal contribution. {jason328,duanj1,hqfang}@uw.edu
[†]Core Contributors

I. INTRODUCTION

Reasoning allows us to act with intention [1]. Before reaching for a cup or moving through a room, we subconsciously weigh the context, goals, and constraints—transforming perception into purpose [1]. This process, grounded in our physical experience of the world, makes our actions coherent, adaptable, and explainable [1]. For robots to operate with the same fluency, they must do more than map images and instructions to control commands. They must learn to reason.

Despite rapid advances in language and vision foundation models, robotics still lags [2]–[4]. Vision-Language-Action (VLA) models [5]–[9] aim to close this gap but generalize poorly across tasks, scenes, and embodiments [10], [11] and offer limited interpretability. The bottleneck is not just data but missing structural inductive biases: as language models shifted from brute-force scaling to structured, reasoning-oriented methods [12]–[14], robotics must follow. We introduce MOLMOACT (**M**ultimodal **O**pen **L**anguage **Mo**del for **Act**ion), a family of completely open **A**ction **R**easoning **M**odels (ARM) that integrate perception, planning, and control through a structured reasoning pipeline. MOLMOACT learns to interpret language instructions, perceive its environment, and generate spatial plans, subsequently executing actions conditioned on its preceding perception and planning demonstrated in Figure 1. This structured design yields both strong performance and high interpretability. On simulation benchmarks such as

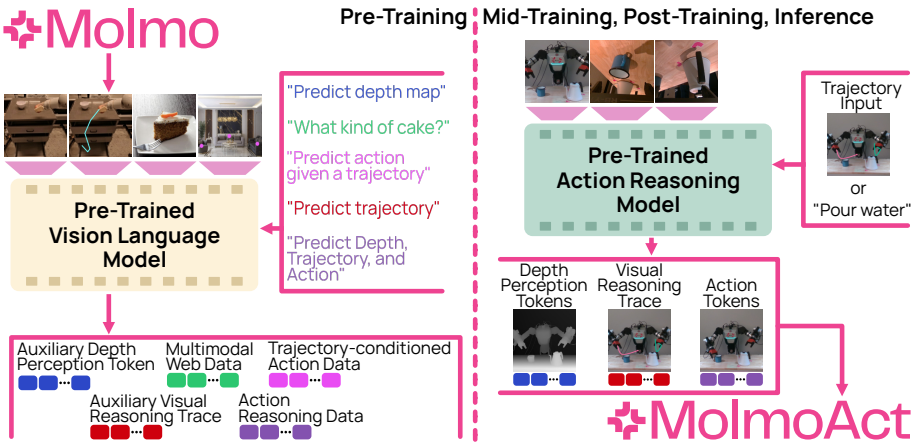

Fig. 2: **Training process of MOLMOACT.** The model training process consists of two stages: **Pre-training** (left) and **Post-training, Mid-training & Inference** (right). In pre-training, the VLM backbone is trained on multimodal and robot reasoning data for tasks including robot control, 2D pointing, trajectory prediction, visual question answering, and perception token prediction. In post-training, the action reasoning model takes multi-view images and either language instructions or trajectory sketches as input, then generates depth tokens, visual reasoning trace tokens, and action tokens for execution.

LIBERO and SimplerEnv, MOLMOACT surpasses state-of-the-art baselines including GR00T N1.5 [8], π_0 and π_0-FAST [5], RT-1 [15], and TraceVLA [16]. In human evaluations for open-ended instruction following, it achieves significantly higher Elo ratings. MOLMOACT also adapts rapidly to new tasks and robot embodiments, demonstrating robust generalization in both simulation and the real world. Its visual reasoning traces make the decision process interpretable and allow users to directly steer behavior by editing trajectories—offering a more reliable interface than language-only prompts.

MOLMOACT is fully open-source: we release model weights, training code, and our action reasoning dataset, which includes 10k teleoperated Franka trajectories across 93 tasks. Together, these contributions establish MOLMOACT as both a state-of-the-art robotic foundation model and an open blueprint for building ARMs that transform perception into purposeful action through grounded reasoning.

II. RELATED WORK

A. Generalist robot manipulation policies

Robotic manipulation is moving from single-task policies to large, diverse datasets spanning scenes, tasks, and embodiments [15], [17]–[20], improving generalization to new environments, objects, and instructions [11], [21]. VLAs leverage LLM/VLM backbones [22]–[25] and are fine-tuned for control via flow matching, diffusion, or tokenized action heads [5]–[7], [15], [16], [26]–[29]. Their drawback is heavy reliance on teleop data; MOLMOACT uses spatial reasoning to reach competitive performance with far less data.

B. Robot reasoning and planning with language

Adding high-level reasoning via LLMs/VLMs—either integrated into policies or used to condition them—improves long-horizon control and generalization [30]–[34]. Other work decouples perception/reasoning from control, producing intermediate plans, graphs, or spatial layouts executed by low-level policies [35]–[39]. MOLMOACT offers language and interactive visual trace steering for interpretable diagnosis and control; unlike RT-Trajectory [34], HAMSTER [36], or inference-time steering [33], [34], [36], [40], it generalizes to new layouts, unseen objects, and ambiguous instructions.

C. Embodied reasoning for robotic manipulation

CoT prompting boosts multi-step reasoning [12] and extends to multimodal settings [41], inspiring embodied variants such as ECoT [42], CoT-VLA [26], and ThinkAct [43]. Closest are Emma-X [44] and [45], which explore mid-level (incl. depth-aware) representations in smaller regimes. MOLMOACT differs by grounding each reasoning step in depth tokens and visual traces—explicit, decodable, and 3D-visualizable—improving explainability and action prediction.

III. MOLMOACT

A. Preliminaries –Vision Language Action Models

To equip an action model with visual and linguistic world knowledge, we build upon vision–language models (VLMs). Current VLMs follow a three-component structure: (i) a visual encoder that transforms an image into patch-level embeddings, (ii) a projection module that maps these visual features into the input space of a language model, and (iii) a large language model (LLM) backbone. These components are trained with a next-token prediction objective on paired or interleaved image–text data. Our work builds on Molmo [25], the Multimodal Open Language Model. We train 2 versions of MOLMOACT, MOLMOACT-7B-D with a backbone based on ViT-SO400M/14 384px SigLIP2 [46] and Qwen2.5-7B [47], and a completely open version, MOLMOACT-7B-O with a VLM backbone based on OpenAI ViT-L/14 336px CLIP [48] and OLMo2-7B [49]. For full details of our model architecture and implementation, please refer to the supplementary video.

We follow prior work [6], [15] in formulating action prediction as a vision–language sequence modeling task. Each action dimension is normalized by dataset quantiles and discretized into 256 uniform bins between the first and ninety-ninth percentiles. Previous approaches map these bins to arbitrary language tokens from the vocabulary tail, ignoring their ordinal structure and yielding a poor initialization for learning an action codebook. We propose a simple, geometry-aware alternative. We select the final 256 tokens from the Qwen2 tokenizer, extract their byte-level BPE symbols, and assign them monotonically to the discretized bins so that adjacent bins map to adjacent symbols. This produces our action token vocabulary V_{action}, whose embeddings share character-level similarity across neighboring bins. We notice that these BPE symbols have a better initialization for action token embeddings by sharing similar characters between adjacent action bins which substantially shortens training time. Compared to GR00T N1.5 's [8] 50,000 GPU-hour pre-training cost, MOLMOACT's pre-training took only 9,216 GPU hours more than fivefold reduction.

B. Action Reasoning Model

a) Depth Perception Tokens: Prior work [41] has shown that depth perception tokens are effective in enhancing chain-of-thought reasoning for visual–spatial tasks. Building on this, we integrate depth estimation for fine-grained control in 3D environments. We define an auxiliary vocabulary of depth perception tokens $V_{\text{depth}} = \{\langle \text{DEPTH}_k \rangle\}_{k=1}^{128}$. For each image, a pre-trained VQVAE [50] encodes the depth map into a sequence of 100 codebook indices, each mapped to its corresponding token in V_{depth}. The resulting token sequence serves as the supervision target and conditions subsequent reasoning and control stages. The pre-trained depth estimator quantizes each depth map into a sequence of indices, each mapped one-to-one to depth tokens, yielding a discrete and interpretable summary of scene geometry. We adopt a specialist-to-generalist distillation approach: the depth specialist provides this sequence as ground truth, and MOLMOACT is trained to autoregressively predict it from the RGB image. Implementation details are in Section III-D.

b) Visual Reasoning Trace: Planning is a critical component in robotics. Rather than decomposing tasks into language-based subtasks, we predict an intermediate 2D trajectory representation that aligns visual inputs with control outputs across diverse robots and tasks. Specifically, we generate the end-effector trajectory and train the model to jointly predict this trajectory and the next action command, a strategy shown effective in prior work [16], [34], [36], [51]. These predicted waypoints align each action to precise end-effector locations, improving fine-grained localization and action-prediction accuracy. We call this representation Visual Reasoning Trace. Given an image observation, visual reasoning trace is a polyline $\boldsymbol{\tau} = (p_1, \ldots, p_L)$ with $1 \le L \le 5$, where each point $p_i = (u_i, v_i)$ is normalized to image resolution, $u_i, v_i \in 0, \ldots, 255$. The first point denotes the current end-effector location, and the remaining points indicate its future positions, evenly subsampled between the current and terminal frames.

c) Action Reasoning Procedure: With depth perception tokens and visual reasoning traces, the model performs spatially grounded action reasoning. Given an RGB observation I and language instruction T, it autoregressively generates three token sequences: (i) depth perception tokens $\mathbf{d}$, (ii) visual reasoning trace $\boldsymbol{\tau}$, and (iii) action tokens $\mathbf{a} = (a_1, \ldots, a_D) \in V_{\text{action}}^D$, where D is the number of control dimensions. Conditioning each stage on the preceding tokens grounds action prediction in both depth and the planned trajectory, yielding spatially consistent, precise control.

C. Action Steerability via Visual Reasoning Trace

We define *steerability* as the ability to guide a policy at test time to perform different behaviors through user-provided instructions. Prior VLAs rely on language-only steering, which suffers from data requirements, linguistic ambiguity, and brittleness to out-of-distribution prompts, leading to inconsistent control. MOLMOACT enables the users sketch a path $\boldsymbol{\tau} = (p_1, \ldots, p_L)$, $1 \le L \le 5$ directly on the image, producing $I^+ = I \oplus \boldsymbol{\tau}$. This representation is precise, editable, and task-agnostic. At test time, conditioning on I^+ and T, the model autoregressively predicts action tokens $\mathbf{a}$, yielding closed-loop execution that accurately follows the user's sketch and remains robust to language variation.

D. Action Reasoning Data Curation

This section details the process for generating ground-truth labels for the depth perception tokens and visual reasoning traces, and explains how these are combined with action labels to train MOLMOACT. A robot episode consists of a sequence of timesteps, where each timestep is a tuple $(I, T, \mathbf{a})_t$: an RGB observation I, a language instruction T, and a ground-truth action $\mathbf{a}$ specified in either end-effector or joint space. This format is common in existing datasets such as DROID and Open-X-Embodiment. However, MOLMOACT extends the dataset with Action Reasoning. To convert any robot dataset into the action reasoning format, we generate ground-truth *Depth Perception Tokens* and *Visual Reasoning Traces* for each timestep. The following sections explain the details of this generation process.

a) Depth Perception Tokens: To generate *Depth Perception Tokens* for each demonstration frame, we first train a VQVAE on 10 million depth maps from RT-1, BridgeData V2, and BC-Z datasets. Depth maps are obtained from observation RGB images using DepthAnything-v2. The VQVAE is trained for 20 epochs with a standard reconstruction loss between input images and their depth maps. After training, each observation image is encoded into latent embeddings, which are quantized using a learned 128-dimensional codebook with a one-to-one index–token mapping. All images are resized to 320×320 px, yielding exactly 100 tokens per image. This enables each depth map to be represented as a token sequence of length 100, which we use as ground-truth labels for our depth perception tokens.

b) Visual Reasoning Trace: To generate a *Visual Reasoning Traces* for each frame of a demonstration, we use Molmo, a vision–language model trained on diverse 2D

pointing datasets, to synthesize gripper trajectories. At each timestep t, we prompt Molmo with "point to the robot gripper" for single-arm robots or "point to the robot gripper on the left/right" for bimanual setups. Molmo predicts a normalized coordinate $(x_t, y_t) \in [0, 100]^2$, which we rescale to integer pixel coordinates $(u_t, v_t) \in [0, 255]^2$. Repeating this for every frame yields a full trajectory $\boldsymbol{\tau}$, or $\boldsymbol{\tau}_L$ and $\boldsymbol{\tau}_R$ for bimanual grippers. For each timestep t, we construct a visual reasoning trace from t to the episode end e. This subsequence includes the current point (u_t, v_t), the final point (u_e, v_e), and up to three uniformly spaced intermediate points. If fewer than three intermediates exist (i.e., $e - t < 4$), all available points are used; if $t = e$, the trace contains only one point. Each trace thus contains 1–5 points, compactly representing the anticipated motion of the end effector.

c) Auxiliary Robot Data: To further enhance MOLMOACT 's spatial reasoning, we extend the data generation pipeline for depth perception tokens and visual reasoning traces to curate three auxiliary supervision dataset: (i) Auxiliary Depth Data—given an RGB observation and language instruction, the model predicts only the depth perception token sequence; (ii) Auxiliary Trace Data—given an RGB observation and instruction, the model predicts only the corresponding visual reasoning trace; and (iii) Trajectory-conditioned Action Data—given $o_t = (I, T, \boldsymbol{\tau})_t$, where I is the current image, T the instruction, and $\boldsymbol{\tau} = (p_1, \ldots, p_L)$ the ground-truth trace, the model predicts the next action using the language T and a trace-overlaid image $I^+ = I \oplus \boldsymbol{\tau}$. This dataset enables MOLMOACT 's steerability feature. After generating ground-truth labels for each frame, we construct the action reasoning dataset by sequentially aligning depth perception tokens, visual reasoning traces, and actions for instruction tuning. The same procedure is applied to curate auxiliary robot data for additional supervision.

E. MolmoAct Dataset

MOLMOACT DATASET is a real-world dataset for general manipulation and spatial reasoning, comprising 10,689 single-arm Franka trajectories (avg. 112 timesteps) across 93 tasks in homes and tabletops. Collected over two months by five trained operators under standardized protocols, the home split (7,730 trajectories; 73 tasks; 20 verbs) was gathered with a DROID-like mobile platform [19] across kitchens, living rooms, bathrooms, and bedrooms, with long-horizon chores decomposed into skill-level subtasks (e.g., "clean up the dishes" becomes "put bowl in dishwasher," "put fork in sink," "cover pot"). The tabletop split contributes 2,959 trajectories over 20 atomic tasks with diverse objects, factorized into reusable motion primitives (open, pick, flip, close) to promote compositional learning and robustness. Further details of MOLMOACT DATASET included in supplementary video.

F. Training

In MOLMOACT, we initialize the vision and language components from publicly available checkpoints and pre-train the VLM using the procedure from Molmo [25] on dense captioning data. After vision–language alignment, we fine-tune MOLMOACT on a subset of the Open X-Embodiment (OXE) mixture and MOLMOACT DATASET, training end-to-end with a next-token prediction objective and computing loss only on action tokens. Please refer to the supplementary video for complete training details.

During pre-training, MOLMOACT is trained on a mixture of action reasoning data, auxiliary robot data, and multimodal web data, totaling 26.3M samples. For robot data, we use 10.5M samples from RT-1, BridgeData V2, and BC-Z (subsampled at 20%, 12.5%, and 7.5%, respectively) and convert them into action reasoning data using our formulation in III-D. Auxiliary supervision includes 1.5M auxiliary depth data samples, 1.5M auxiliary trace data samples, and 10.5M trajectory-conditioned action data samples, co-trained with 2M multimodal web samples at 5% sampling rate. Training is conducted on 256 H100 GPUs for 100k gradient steps with a batch size of 512, requiring 9,728 GPU hours. At each step, batches are sampled proportionally to their defined rates.

The second stage, mid-training, focuses on high-quality action reasoning data from MOLMOACT DATASET, targeting household manipulation tasks. We create 1M action reasoning and 1M trajectory-conditioned action samples. Each episode consists of two side-mounted and one wrist camera view, which we convert into paired-view training examples (side + wrist). Depth perception tokens and visual reasoning traces are generated from the side view, with the wrist view serving as context. We train for 50k gradient steps with a batch size of 128 on 128 H100 GPUs, taking 2,304 GPU hours.

The final stage, post-training, enable us to rapidly adapt MOLMOACT to new tasks and embodiments. For each task, we collect 30–50 teleoperated demonstrations, generate depth perception tokens and visual reasoning traces, and convert them into action reasoning and trajectory-conditioned data. Unlike earlier stages, we apply action chunking [57] with chunk size $N = 8$, tokenizing each chunk identically to single actions and training the model autoregressively on all chunks. Post-training is performed via parameter-efficient LoRA fine-tuning (rank=32, $\alpha = 16$), preserving pre-trained capabilities. We use a batch size of 128 for simulation benchmarks (e.g., LIBERO) and 64 for real-world tasks, with the number of gradient steps varying by task. Most post-training data pairs a front- or side-view with a wrist view, and bimanual setups include multiple wrist views.

IV. EXPERIMENTAL EVALUATION

We rigorously evaluate MOLMOACT (and its MOLMOACT-7B-D variant) against strong baselines. Our evaluation spans (i) out-of-the-box performance after pre-training, (ii) adaptability via lightweight post-training across tasks, domains, and embodiments, and (iii) interactive, steerable action reasoning. Across simulation and real-world settings, we ask these questions for MOLMOACT: **A)** How strong is pre-training performance? **B)** How well does post-training enable generalization to new tasks, domains, and embodiments? **C)** How robust is the model beyond its training distribution? **D)** How steerable is it, and how does steerability enhance

TABLE I: **SimplerEnv evaluation across different policies on Google Robot tasks.** The zero-shot and fine-tuning results denote performance of OXE dataset [22] pre-trained models and RT-1 dataset [15] fine-tuned models, respectively.

Model	Visual Matching			Avg	Variant Aggregation			Avg
	Pick Coke Can	Move Near	Open/Close Drawer		Pick Coke Can	Move Near	Open/Close Drawer	
HPT [52]	56.0%	60.0%	24.0%	46.0%		—		—
TraceVLA [16]	28.0%	53.7%	57.0%	42.0%	60.0%	56.4%	31.0%	45.0%
RT-1-X [15]	56.7%	31.7%	59.7%	53.4%	49.0%	32.3%	29.4%	39.6%
RT-2-X [53]	78.7%	77.9%	25.0%	60.7%	82.3%	79.2%	35.3%	64.3%
Octo-Base [54]	17.0%	4.2%	22.7%	16.8%	0.6%	3.1%	1.1%	1.1%
OpenVLA [6]	16.3%	46.2%	35.6%	27.7%	54.5%	47.7%	17.7%	39.8%
RoboVLM (zero-shot) [10]	72.7%	66.3%	26.8%	56.3%	68.3%	56.0%	8.5%	46.3%
RoboVLM (fine-tuned)	77.3%	61.7%	43.5%	63.4%	75.6%	60.0%	10.6%	51.3%
Emma-X [44]	2.3%	3.3%	18.3%	8.0%	5.3%	7.3%	20.5%	11.0%
Magma [9]	56.0%	65.4%	83.7%	68.4%	53.4%	65.7%	68.8%	62.6%
π_0 (fine-tuned) [5]	72.7%	65.3%	38.3%	58.7%	75.2%	63.7%	25.6%	54.8%
π_0-FAST (fine-tuned)	75.3%	67.5%	42.9%	61.9%	77.6%	68.2%	31.3%	59.0%
GR00T-N1.5 (fine-tuned) [8]	69.3%	68.7%	35.8%	52.4%	46.7%	62.9%	17.5%	42.4%
SpatialVLA [28]	81.0%	69.6%	59.3%	70.0%	89.5%	71.7%	36.2%	65.8%
MOLMOACT (zero-shot)	71.3%	73.8%	66.5%	70.5%	57.8%	43.8%	76.7%	59.3%
MOLMOACT (fine-tuned)	77.7%	77.1%	60.0%	**71.6%**	76.1%	61.3%	78.8%	**72.1%**

TABLE II: **LIBERO benchmark success rates** across four task categories (Spatial, Object, Goal, and Long-horizon) along with the average performance. MOLMOACT achieves the highest overall average success rate of 86.6%, outperforming all baselines, with strong performance across all categories, particularly in long-horizon tasks.

Baseline	Spatial	Object	Goal	Long	Avg
TraceVLA [16]	84.6%	85.2%	75.1%	54.1%	74.8%
Octo-Base [54]	78.9%	85.7%	84.6%	51.1%	75.1%
OpenVLA [6]	84.7%	88.4%	79.2%	53.7%	76.5%
SpatialVLA [28]	88.2%	89.9%	78.6%	55.5%	78.1%
CoT-VLA [26]	87.5%	91.6%	87.6%	69.0%	83.9%
NORA-AC [55]	85.6%	89.4%	80.0%	63.0%	79.5%
WorldVLA [56]	87.6%	96.2%	83.4%	60.0%	79.1%
π_0-FAST [5]	96.4%	96.8%	88.6%	60.2%	85.5%
ThinkAct [43]	88.3%	91.4%	87.1%	70.9%	84.4%
MOLMOACT-7B-D	87.0%	95.4%	87.6%	77.2%	**86.6%**

TABLE III: **Real-world evaluation** of OpenVLA, π_0-FAST, and MOLMOACT on single-arm and bimanual Franka tasks. Bar plots show mean task progress with standard error over 25 trials per task.

Single-arm Franka Tasks			
Task	**OpenVLA** [6]	**π_0-FAST** [5]	**MOLMOACT-7B-D**
Put Bowl in Sink	0.25 ± 0.09	0.70 ± 0.09	**0.83** ± 0.08
Wipe Table	0.26 ± 0.09	0.82 ± 0.08	**1.00** ± 0.00
Table Bussing	0.53 ± 0.08	**0.85** ± 0.07	0.84 ± 0.07
Bimanual Franka Tasks			
Task	**OpenVLA** [6]	**π_0-FAST** [5]	**MOLMOACT-7B-D**
Set Table	0.30 ± 0.09	0.61 ± 0.10	**0.77** ± 0.08
Lift Tray	**1.00** ± 0.00	0.74 ± 0.09	**1.00** ± 0.00
Fold Towel	0.32 ± 0.09	0.52 ± 0.10	**0.80** ± 0.08

user interaction? Note that we have also answered two more questions in the supplementary video: **E)** How does mid-training on the MOLMOACT DATASET improve MOLMOACT 's performance? **F)** How effectively does MOLMOACT follow language commands?

A. MOLMOACT *After Pre-training*

Evaluation Setup and Baselines.

In our first set of experiments, we evaluate MOLMOACT directly after pre-training on the SimplerEnv benchmark to evaluate how well our base model perform out-of-the-box. SimplerEnv consists of both visual-matching and variant-aggregation tasks across WidowX and Google Robot platforms (details in the supplementary video). We focused on the Google Robot Visual Matching task suite, as MOLMOACT's pre-training data mixture primarily consists of Google Robot datasets—specifically BC-Z [18] and RT-1 [15]—which contribute approximately 20% and 7.5% of the data, respectively. We compared MOLMOACT-7B-D-PRETRAIN against a set of generalist policies. Most baselines were evaluated out-of-the-box, with some additionally fine-tuned on a portion of the RT-1 dataset before evaluation. We additionally fine-tuned MOLMOACT-7B-D-PRETRAIN on the RT-1 dataset evaluate it's performance after fine-tuning.

Evaluation Results. MOLMOACT-7B-D-PRETRAIN achieved strong zero-shot performance on the SimplerEnv visual-matching suite, reaching 70.5% success rate and outperforming baselines such as GR00T N1.5, π_0, π_0-FAST, and Magma. With fine-tuning on the same RT-1 subset of OXE, MOLMOACT-7B-D improved to 71.6%, exceeding Magma by 3.2% as shown in Table I. We attribute the strong performance of MOLMOACT to it's ability to reason with *Depth Perception Tokens* and *Visual Reasoning Trace*. These results indicate that MOLMOACT is both an effective zero-shot generalist and a strong initialization for fine-tuned deployment.

B. Fast Adaptation of MOLMOACT *in Post-training*

Evaluation Setups and Baselines. We evaluate MOLMOACT in both simulation and real-world settings to see how well MOLMOACT can adapt to new tasks and embodiments after post-training. In simulation, we evaluate on the LIBERO benchmark [58], which consists of a Franka Panda arm with demonstrations containing front and wrist view camera images (256×256 px), language instructions, and delta end-effector pose actions. We follow prior works ([6]) and

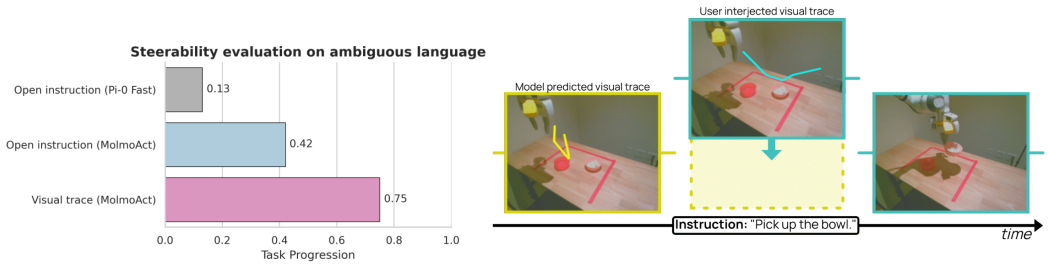

Fig. 3: **Steerability evaluation with open instructions and visual traces. Left:** Success rates for different steering modes, showing that MOLMOACT with visual trace steering achieves the highest success rate (0.75), outperforming its open-instruction variant and π_0-FAST. **Right:** Example of the *"Pick up the bowl"* task: the model-predicted trajectory (yellow) is adjusted via a user-provided steering trajectory (cyan), resulting in the corrected task completion.

evaluate on four task suites – LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long – each with 500 demonstrations across 10 tasks. Following ([6]), we trained on a modified dataset that filtered out no-op actions and unsuccessful demonstrations. Moreover, we set action chunk size to $K = 8$ for evaluation on each task suite and execute full chunks before redoing action reasoning. We fine-tune MOLMOACT-7B-D using Low-Rank Adaptation (LoRA) and compared to state-of-the-art generalist autoregressive policies.

In the real world, we evaluate MOLMOACT on six tasks across single-arm and bimanual Franka setups. The single-arm tasks include `put_bowl_in_sink`, `wipe_table`, and `table_bussing`. The bimanual tasks include `set_table`, `lift_tray`, and `fold_towel`. For each task, we collected 50 human tele-operated demonstrations and post-trained both MOLMOACT-7B-D and baseline models. We evaluate the task progress over 25 trials per task. We detail the task progress scores in the supplementary video. This setup enables a comprehensive comparison of adaptation efficiency across tasks and embodiments.

Evaluation Results. On the LIBERO benchmark, MOLMOACT-7B-D achieves an average success rate of 86.6%, the highest among all discrete action methods (Table II). We benchmark only against discrete action baselines, as both OpenVLA-OFT [59] and our experiments find that continuous action representations yield 5% absolute higher success rates due to higher action prediction precision. Hence we evaluate MolmoAct against π_0's discrete action space version π_0-FAST for equal comparisons. It performs well on LIBERO-Long, a challenging long-horizon suite, where it exceeds the performance of ThinkAct—the second-best method in this setting—by 6.3%. In the real world, MOLMOACT demonstrates effective fine-tuning and generalization across different embodiments (Table III). It outperforms π_0-FAST by an average of 10% in task progression on single-arm tasks and by 22.7% on bimanual tasks.

C. *Effectiveness of* MOLMOACT *in Out-of-Distribution*

We evaluate MOLMOACT in both simulated and real-world settings to assess its ability to generalize beyond the training distribution, both out-of-the-box and after post-training. In simulation, we adopt SimplerEnv Variant-Aggregation, which introduces environment and task perturbations such as distractors, lighting, backgraound and table texture changes. We compare MOLMOACT-7B-D-PRETRAIN against several state-of-the-art generalist policies—TraceVLA, RT-1X, OpenVLA, RoboVLM, Emma-X, π_0-FAST, and SpatialVLA. In the real-world, we evaluated MOLMOACT-7B-D using a single Franka arm after multi-task post-training on a multi-task setup involving three objects and two different-colored plates arranged on a tabletop.

We collect over 300 tele-operated demonstrations for all three tasks, then post-train MOLMOACT-7B-D and each baselines in a multi-task setting. During evaluation, we test generalization in four aspects: (1) **Language Variation**: rephrased instructions, (2) **Spatial Variation**: changes in target object position, (3) **Distractors**: additional distractor objects, and (4) **Novel Objects**: substituting target objects with novel objects unseen during post-training. We benchmark MOLMOACT-7B-D against π_0-FAST and OpenVLA, evaluating three task variants per benchmarked task and conducting four trials per variant. Full task and variation details are presented in the supplementary video.

Evaluation Results. In simulation, MOLMOACT-7B-D-PRETRAIN after fine-tuning achieves 72.1% on the variant aggregation tasks as shown in Table I, outperforming all baselines and exceeding the second-best model, RT-2-X, by 7.8%. The performance difference between variant aggregation and visual matching is less than 1%, highlighting MOLMOACT's robustness to visual and distributional shifts. In the real world, MOLMOACT-7B-D consistently surpasses all baselines across all generalization axes (Table IV), achieving a 23.3% average improvement in task progression over π_0-FAST.

D. *Steerability of* MOLMOACT

Evaluation Setups and Baselines. We evaluate MOLMOACT's ability to steer robot actions, particularly when language instructions are ambiguous. Specifically, we investigate the effectiveness of different interaction mediums in guiding MOLMOACT toward user-intended targets during task execution. For this purpose, we set up a `pick_up_bowl` task, post-training MOLMOACT-7B-D and the baseline model (π_0-FAST) with 100 collected demonstrations, each annotated with two distinct language instructions: one specifying the clean bowl and the other the dirty bowl. During evaluation,

TABLE IV: **MOLMOACT outperforms baselines across generalization settings.** Task progression scores for OpenVLA, π_0-FAST, and MOLMOACT across in-distribution, language variation, spatial variation, distractors, and novel object conditions, showing consistent gains for MOLMOACT.

Category	OpenVLA [6]	π_0-FAST [5]	MOLMOACT-7B-D
In-distribution	0.38	0.65	**0.79**
Language Variations	0.23	0.29	**0.67**
Spatial Variations	0.40	0.46	**0.54**
Distractors	0.29	0.54	**0.75**
Novel Objects	0.29	0.29	**0.65**
Average	0.32	0.45	**0.68**

we first provide ambiguous instructions such as "pick up (the) bowl," prompting MOLMOACT-7B-D to predict an initial trajectory towards one of the bowls. Subsequently, we test two steering methods: visual trace sketches to instruct the model toward the alternative bowl, and open-ended natural language instructions provided by participants (N=10) which are different from the ground-truth instruction. For comparison, we also attempt to steer the actions of π_0-FAST by changing language instructions at test-time. Each model is evaluated in 15 trials, and the performance is evaluated according to the progression of the task. For more details about the setting, please refer to the supplementary video.

Evaluation Results. Based on our experiments, we observed that MOLMOACT-7B-D is notably more steerable via visual trace inputs, achieving a success rate of 75%. Additionally, steering using visual traces significantly outperforms steering via open-ended natural language instructions by a margin of 33%. Lastly, we demonstrate that MOLMOACT-7B-D exhibits superior instruction-following capabilities compared to the baseline model, π_0-FAST. Specifically, when steering robot actions using open-ended language instructions, MOLMOACT surpasses π_0-FAST by a substantial margin of 29%, highlighting its enhanced instruction-following capabilities to user commands.

V. CONCLUSION

We introduced MOLMOACT, a fully open family of action reasoning models that unify perception, planning, and control through spatial reasoning. Combining depth tokens, visual reasoning traces, and action prediction, MOLMOACT produces explainable, steerable behaviors and consistently outperforms VLA baselines, adapting efficiently to new tasks and generalizing robustly. We release model weights, code, and the MOLMOACT DATASET dataset to foster reproducibility and advance research on foundation models that transform perception into purposeful action.

ACKNOWLEDGEMENTS

AI tools (Claude, ChatGPT) were used in a limited capacity (grammar enhancement, code autocompletion). The authors are responsible for all content in this article. This project is largely funded and supported by the Allen Institute for Artificial Intelligence.

REFERENCES

[1] B. Tversky, "Your body thinks as much as your mind," *IAI News*, Aug. 2025, institute of Art and Ideas. [Online]. Available: https://iai.tv/articles/your-body-thinks-as-much-as-your-mind-auid-3282

[2] J. Duan, S. Yu, H. L. Tan, H. Zhu, and C. Tan, "A survey of embodied ai: From simulators to research tasks," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 6, no. 2, pp. 230–244, 2022.

[3] Z. Xu, K. Wu, J. Wen, J. Li, N. Liu, Z. Che, and J. Tang, "A survey on robotics with foundation models: toward embodied ai," *arXiv preprint arXiv:2402.02385*, 2024.

[4] R. Firoozi, J. Tucker, S. Tian, A. Majumdar, J. Sun, W. Liu, Y. Zhu, S. Song, A. Kapoor, K. Hausman, *et al.*, "Foundation models in robotics: Applications, challenges, and the future," *The International Journal of Robotics Research*, vol. 44, no. 5, pp. 701–739, 2025.

[5] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, *et al.*, "π0: A vision-language-action flow model for general robot control. corr, abs/2410.24164, 2024. doi: 10.48550," *arXiv preprint ARXIV.2410.24164*.

[6] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, *et al.*, "Openvla: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

[7] G. R. Team, S. Abeyruwan, J. Ainslie, J.-B. Alayrac, M. G. Arenas, T. Armstrong, A. Balakrishna, R. Baruch, M. Bauza, M. Blokzijl, *et al.*, "Gemini robotics: Bringing ai into the physical world," *arXiv preprint arXiv:2503.20020*, 2025.

[8] NVIDIA, :, J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. J. Fan, Y. Fang, D. Fox, F. Hu, S. Huang, J. Jang, Z. Jiang, J. Kautz, K. Kundalia, L. Lao, Z. Li, Z. Lin, K. Lin, G. Liu, E. Llontop, L. Magne, A. Mandlekar, A. Narayan, S. Nasiriany, S. Reed, Y. L. Tan, G. Wang, Z. Wang, J. Wang, Q. Wang, J. Xiang, Y. Xie, Y. Xu, Z. Xu, S. Ye, Z. Yu, A. Zhang, H. Zhang, Y. Zhao, R. Zheng, and Y. Zhu, "Gr00t n1: An open foundation model for generalist humanoid robots," 2025. [Online]. Available: https://arxiv.org/abs/2503.14734

[9] J. Yang, R. Tan, Q. Wu, R. Zheng, B. Peng, Y. Liang, Y. Gu, M. Cai, S. Ye, J. Jang, *et al.*, "Magma: A foundation model for multimodal ai agents," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 14 203–14 214.

[10] H. Liu, X. Li, P. Li, M. Liu, D. Wang, J. Liu, B. Kang, X. Ma, T. Kong, and H. Zhang, "Towards generalist robot policies: What matters in building vision-language-action models," 2025.

[11] W. Pumacay, I. Singh, J. Duan, R. Krishna, J. Thomason, and D. Fox, "The colosseum: A benchmark for evaluating generalization for robotic manipulation," *arXiv preprint arXiv:2402.08191*, 2024.

[12] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[13] E. Zelikman, Y. Wu, J. Mu, and N. Goodman, "Star: Bootstrapping reasoning with reasoning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 15 476–15 488, 2022.

[14] J. Huang, S. S. Gu, L. Hou, Y. Wu, X. Wang, H. Yu, and J. Han, "Large language models can self-improve," *arXiv preprint arXiv:2210.11610*, 2022.

[15] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, *et al.*, "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.

[16] R. Zheng, Y. Liang, S. Huang, J. Gao, H. Daumé III, A. Kolobov, F. Huang, and J. Yang, "Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies," *arXiv preprint arXiv:2412.10345*, 2024.

[17] H.-S. Fang, H. Fang, Z. Tang, J. Liu, C. Wang, J. Wang, H. Zhu, and C. Lu, "Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot," *arXiv preprint arXiv:2307.00595*, 2023.

[18] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn, "Bc-z: Zero-shot task generalization with robotic imitation learning," in *Conference on Robot Learning*. PMLR, 2022, pp. 991–1002.

[19] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, *et al.*, "Droid: A large-scale in-the-wild robot manipulation dataset," *arXiv preprint arXiv:2403.12945*, 2024.

[20] H. R. Walke, K. Black, T. Z. Zhao, Q. Vuong, C. Zheng, P. Hansen-Estruch, A. W. He, V. Myers, M. J. Kim, M. Du, *et al.*, "Bridgedata v2: A dataset for robot learning at scale," in *Conference on Robot Learning*. PMLR, 2023, pp. 1723–1736.

[21] A. Xie, L. Lee, T. Xiao, and C. Finn, "Decomposing the generalization gap in imitation learning for visual robotic manipulation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 3153–3160.

[22] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 6892–6903.

[23] G. Team, P. Georgiev, V. I. Lei, R. Burnell, L. Bai, A. Gulati, G. Tanzer, D. Vincent, Z. Pan, S. Wang, *et al.*, "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context," *arXiv preprint arXiv:2403.05530*, 2024.

[24] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *Advances in neural information processing systems*, vol. 36, pp. 34 892–34 916, 2023.

[25] M. Deitke, C. Clark, S. Lee, R. Tripathi, Y. Yang, J. S. Park, M. Salehi, N. Muennighoff, K. Lo, L. Soldaini, *et al.*, "Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models," *arXiv e-prints*, pp. arXiv–2409, 2024.

[26] Q. Zhao, Y. Lu, M. J. Kim, Z. Fu, Z. Zhang, Y. Wu, Z. Li, Q. Ma, S. Han, C. Finn, *et al.*, "Cot-vla: Visual chain-of-thought reasoning for vision-language-action models," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 1702–1713.

[27] Q. Li, Y. Liang, Z. Wang, L. Luo, X. Chen, M. Liao, F. Wei, Y. Deng, S. Xu, Y. Zhang, *et al.*, "Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation," *arXiv preprint arXiv:2411.19650*, 2024.

[28] D. Qu, H. Song, Q. Chen, Y. Yao, X. Ye, Y. Ding, Z. Wang, J. Gu, B. Zhao, D. Wang, *et al.*, "Spatialvla: Exploring spatial representations for visual-language-action model," *arXiv preprint arXiv:2501.15830*, 2025.

[29] K. Pertsch, K. Stachowicz, B. Ichter, D. Driess, S. Nair, Q. Vuong, O. Mees, C. Finn, and S. Levine, "Fast: Efficient action tokenization for vision-language-action models," *arXiv preprint arXiv:2501.09747*, 2025.

[30] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," *arXiv preprint arXiv:2204.01691*, 2022.

[31] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "Voxposer: Composable 3d value maps for robotic manipulation with language models," *arXiv preprint arXiv:2307.05973*, 2023.

[32] H. Fang, M. Grotz, W. Pumacay, Y. R. Wang, D. Fox, R. Krishna, and J. Duan, "Sam2act: Integrating visual foundation model with a memory architecture for robotic manipulation," 2025. [Online]. Available: https://arxiv.org/abs/2501.18564

[33] Y. Wang, L. Wang, Y. Du, B. Sundaralingam, X. Yang, Y.-W. Chao, C. Perez-D'Arpino, D. Fox, and J. Shah, "Inference-time policy steering through human interactions," *arXiv preprint arXiv:2411.16627*, 2024.

[34] J. Gu, S. Kirmani, P. Wohlhart, Y. Lu, M. G. Arenas, K. Rao, W. Yu, C. Fu, K. Gopalakrishnan, Z. Xu, P. Sundaresan, P. Xu, H. Su, K. Hausman, C. Finn, Q. Vuong, and T. Xiao, "Rt-trajectory: Robotic task generalization via hindsight trajectory sketches," 2023.

[35] J. Duan, W. Yuan, W. Pumacay, Y. R. Wang, K. Ehsani, D. Fox, and R. Krishna, "Manipulate-anything: Automating real-world robots using vision-language models," *arXiv preprint arXiv:2406.18915*, 2024.

[36] Y. Li, Y. Deng, J. Zhang, J. Jang, M. Memmel, R. Yu, C. R. Garrett, F. Ramos, D. Fox, A. Li, *et al.*, "Hamster: Hierarchical action models for open-world robot manipulation," *arXiv preprint arXiv:2502.05485*, 2025.

[37] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei, "Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation," *arXiv preprint arXiv:2409.01652*, 2024.

[38] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," *arXiv preprint arXiv:2209.07753*, 2022.

[39] J. Duan, W. Pumacay, N. Kumar, Y. R. Wang, S. Tian, W. Yuan, R. Krishna, D. Fox, A. Mandlekar, and Y. Guo, "Aha: A vision-language-model for detecting and reasoning over failures in robotic manipulation," *arXiv preprint arXiv:2410.00371*, 2024.

[40] S. Liu, I. S. Singh, Y. Xu, J. Duan, and R. Krishna, "Vls: Steering pretrained robot policies via vision-language models," *arXiv preprint arXiv:2602.03973*, 2026.

[41] M. Bigverdi, Z. Luo, C.-Y. Hsieh, E. Shen, D. Chen, L. G. Shapiro, and R. Krishna, "Perception tokens enhance visual reasoning in multimodal language models," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 3836–3845.

[42] M. Zawalski, W. Chen, K. Pertsch, O. Mees, C. Finn, and S. Levine, "Robotic control via embodied chain-of-thought reasoning," *arXiv preprint arXiv:2407.08693*, 2024.

[43] C.-P. Huang, Y.-H. Wu, M.-H. Chen, Y.-C. F. Wang, and F.-E. Yang, "Thinkact: Vision-language-action reasoning via reinforced visual latent planning," *arXiv preprint arXiv:2507.16815*, 2025.

[44] Q. Sun, P. Hong, T. D. Pala, V. Toh, U. Tan, D. Ghosal, S. Poria, *et al.*, "Emma-x: An embodied multimodal action model with grounded chain of thought and look-ahead spatial reasoning," *arXiv preprint arXiv:2412.11974*, 2024.

[45] J. Yang, C. K. Fu, D. Shah, D. Sadigh, F. Xia, and T. Zhang, "Bridging perception and action: Spatially-grounded mid-level representations for robot generalization," *arXiv preprint arXiv:2506.06196*, 2025.

[46] M. Tschannen, A. Gritsenko, X. Wang, M. F. Naeem, I. Alabdulmohsin, N. Parthasarathy, T. Evans, L. Beyer, Y. Xia, B. Mustafa, *et al.*, "Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features," *arXiv preprint arXiv:2502.14786*, 2025.

[47] Qwen, :, A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, H. Lin, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Lin, K. Dang, K. Lu, K. Bao, K. Yang, L. Yu, M. Li, M. Xue, P. Zhang, Q. Zhu, R. Men, R. Lin, T. Li, T. Tang, T. Xia, X. Ren, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Wan, Y. Liu, Z. Cui, Z. Zhang, and Z. Qiu, "Qwen2.5 technical report," 2025. [Online]. Available: https://arxiv.org/abs/2412.15115

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

[49] T. OLMo, P. Walsh, L. Soldaini, D. Groeneveld, K. Lo, S. Arora, A. Bhagia, Y. Gu, S. Huang, M. Jordan, *et al.*, "2 olmo 2 furious," *arXiv preprint arXiv:2501.00656*, 2024.

[50] A. Van Den Oord, O. Vinyals, *et al.*, "Neural discrete representation learning," *Advances in neural information processing systems*, vol. 30, 2017.

[51] D. Niu, Y. Sharma, G. Biamby, J. Quenum, Y. Bai, B. Shi, T. Darrell, and R. Herzig, "Llarva: Vision-action instruction tuning enhances robot learning," *arXiv preprint arXiv:2406.11815*, 2024.

[52] L. Wang, X. Chen, J. Zhao, and K. He, "Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers," *Advances in neural information processing systems*, vol. 37, pp. 124 420–124 450, 2024.

[53] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *Conference on Robot Learning*. PMLR, 2023, pp. 2165–2183.

[54] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, *et al.*, "Octo: An open-source generalist robot policy," *arXiv preprint arXiv:2405.12213*, 2024.

[55] C.-Y. Hung, Q. Sun, P. Hong, A. Zadeh, C. Li, U. Tan, N. Majumder, S. Poria, *et al.*, "Nora: A small open-sourced generalist vision language action model for embodied tasks," *arXiv preprint arXiv:2504.19854*, 2025.

[56] J. Cen, C. Yu, H. Yuan, Y. Jiang, S. Huang, J. Guo, X. Li, Y. Song, H. Luo, F. Wang, *et al.*, "Worldvla: Towards autoregressive action world model," *arXiv preprint arXiv:2506.21539*, 2025.

[57] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," *arXiv preprint arXiv:2304.13705*, 2023.

[58] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone, "Libero: Benchmarking knowledge transfer for lifelong robot learning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 44 776–44 791, 2023.

[59] M. J. Kim, C. Finn, and P. Liang, "Fine-tuning vision-language-action models: Optimizing speed and success," 2025. [Online]. Available: https://arxiv.org/abs/2502.19645