EGOLAP: LEARNING FROM EGOCENTRIC HUMAN DATA THROUGH LANGUAGE-ACTION REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Egocentric human data offer a path to scaling robot learning beyond costly robot demonstrations, yet the embodiment gap makes raw human trajectories a poor supervisory target for control. Our key insight is that, although low-level actions are embodiment-specific, their underlying *motion intent* can capture task-relevant structure that transfers across humans and robots. We introduce EGOLAP, a VLA pre-training framework that jointly learns from human and robot trajectories through a shared language-based action chain-of-thought. EgoLAP expresses motion intent as structured, temporally abstracted language actions and pairs them with motion-level reasoning grounded in scene geometry, physics, and object affordances. Across extensive real-world and simulated experiments, EgoLAP transfers human experience to robot control more effectively than alternative action representations and reaches 80.1% mean real-world task progress, a $2.3\times$ performance gain over alternative action representations. Motion-level reasoning also outperforms a composite reasoning format that combines subtask, object-box, and visual-trace reasoning.

1 INTRODUCTION

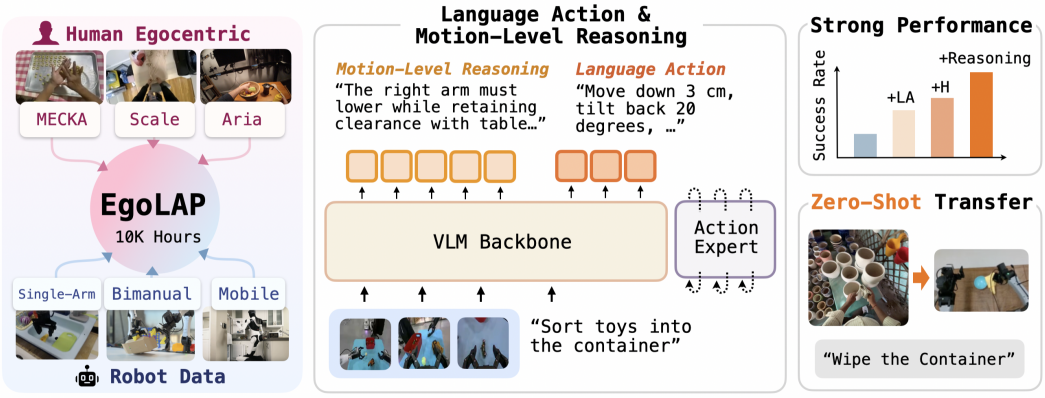


Figure 1: **EgoLAP turns egocentric human experience into transferable robot control.** Human hands and robots differ in embodiment and low-level execution, yet share the same underlying motion intent when performing a task. EgoLAP captures this intent through motion-level reasoning and language actions, providing a shared supervision target across both data sources and leading to a $2.3\times$ performance gain over alternative action representations.

Robot foundation models have advanced rapidly by scaling robot demonstrations across tasks, environments, and embodiments (Collaboration et al., 2025; Khazatsky et al., 2025; Intelligence et al., 2025; Brohan et al., 2023; Kim et al., 2024; Ye et al., 2026), yet such demonstrations remain costly to collect and narrow relative to the breadth of human activity. Egocentric human video offers the scale and behavioral diversity that robot data lacks, capturing long-horizon manipulation in everyday environments and growing independently of any robot platform.

To leverage human data for robot learning, prior work extracts intermediate representations through hand-engineered pipelines, such as interaction hotspots and affordances (Nagarajan et al., 2019;

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Srirama et al., 2024), hand poses or image-space tracks (Bharadhwaj et al., 2023; Wen et al., 2024; Bharadhwaj et al., 2024b; Ren et al., 2025), latent plans (Wang et al., 2023; Jain et al., 2024; Bharadhwaj et al., 2024a), or visually edited embodiments (Bahl et al., 2022; Lepert et al., 2026; 2025). These representations capture useful facets of human behavior, but inherit the inductive biases and failure modes of their extraction pipelines that restricts scalability. Moreover, differences in morphology, kinematics, and control create an embodiment gap that limits the direct transfer of action-level knowledge from human data to robot policies (Cai et al., 2025).

Our key insight is that humans and robots need not share low-level actions; instead, they need to share *motion intent*: what motion needs to be produced, and why it's appropriate given the current physical state. We represent the *what* using *language actions*: structured, temporally abstract descriptions of changes in pose and gripper state, such as "move forward 3 cm, rotate clockwise 14 degrees, and close gripper." Unlike embodiment-specific trajectories or non-linguistic action tokens (Pertsch et al., 2025; Liu et al., 2026), language actions reuse the vocabulary and compositional structure of a vision–language model (VLM), thereby providing control-relevant supervision within its pretrained semantic space (Zha et al., 2026; Hancock et al., 2025; Grover et al., 2025). However, the same task may require different motions across embodiments, so specifying *what* motion to execute is insufficient. We therefore introduce *motion-level reasoning* to represent the *why*: local, physically grounded rationales that relate contact, geometry, and object affordances to the intended effect of the next motion. Unlike subtask plans or visual traces, which primarily specify what action to take or where to move (Zawalski et al., 2025; Sun et al., 2024; 2026), motion-level reasoning captures the physical relationship between the current state and the motion required to achieve the intended effect, providing a form of supervision that can connect directly to low-level actions, yet transfer across embodiments. Together, motion-level reasoning and language actions define a physically grounded "action chain-of-thought".

Statement of contributions. We introduce EGOLAP, a VLA pre-training framework that co-trains on egocentric human and robot trajectories through motion-level reasoning and language actions, achieving 44.9 points of real-world performance more than the strongest action representation. Unlike previous embodied reasoning methods (Zawalski et al., 2025; Sun et al., 2024), EgoLAP only runs a lightweight action expert during inference time, where the textual targets provide auxiliary supervision and require no autoregressive generation. Across extensive zero-shot real-world and simulation experiments on a bimanual robot, language actions transfer human experience better than non-linguistic action representations (Pertsch et al., 2025; Liu et al., 2026), and motion-level reasoning contributes 44.7 points of real-world task progress more to EgoLAP than the baselines. Motion-level reasoning also strongly outperforms a strong composite of subtask, object-box, and visual-trace reasoning supervision. Finally, EgoLAP adapts effectively through reinforcement learning over language actions and motion-level reasoning.

2 RELATED WORK

Learning Robot Policies from Egocentric Human Data. Human data provides scalable supervision for robot learning, but leveraging them requires bridging the embodiment gap. Prior methods rely on method-specific representations: interaction hotspots and affordances (Nagarajan et al., 2019; Srirama et al., 2024), hand poses or image-space tracks (Bharadhwaj et al., 2023; Wen et al., 2024; Bharadhwaj et al., 2024b; Ren et al., 2025), latent plans (Wang et al., 2023; Jain et al., 2024; Bharadhwaj et al., 2024a), or visually edited embodiments (Bahl et al., 2022; Lepert et al., 2026; 2025). Although effective, these hand-designed interfaces require modular processing (e.g., hand-pose estimation, point tracking, or embodiment editing) and inherit its inductive biases and errors, limiting their scalability. The gap is evident in direct policy learning: despite the 3,000 hours of egocentric video available in Ego4D (Grauman et al., 2022), MimicPlay and Motion Tracks train on only about 10 minutes of task-specific human video, ATM evaluates transfer from 100 human videos, and Phantom trains on 250–350 demonstrations per task (Wang et al., 2023; Ren et al., 2025; Wen et al., 2024; Lepert et al., 2026).

Another line directly trains policies on large-scale egocentric human datasets (Punamiya et al., 2026; Hoque et al., 2026) along with robot data (Kareer et al., 2024; Zheng et al., 2026; Punamiya et al., 2025; Li et al., 2026; Qiu et al., 2025; Kim et al., 2026b; Yang et al., 2025; Luo et al., 2026). These systems typically retain embodiment-specific pose targets or encode trajectories with non-linguistic

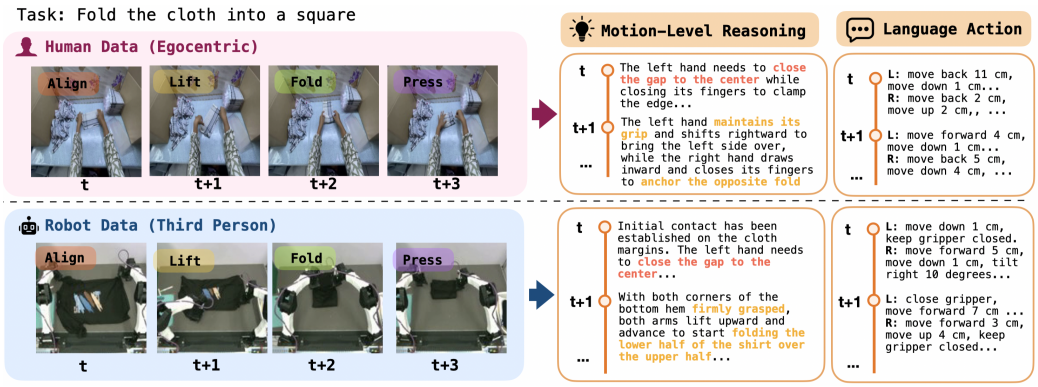


Figure 2: **EgoLAP places human and robot motion in a shared, physically grounded language space.** By capturing the physical relationship between the current state and the motion needed to produce the intended effect, motion-level reasoning provides supervision that connects directly to low-level actions that transfers across embodiments.

action tokens as training targets (Pertsch et al., 2025; Kareer et al., 2025; Wei et al., 2026), which impede cross-embodiment transfer (Zha et al., 2026). Language-Action Pre-training (LAP) (Zha et al., 2026) instead expresses motion as structured natural language, showing improved zero-shot transfer among robots. EgoLAP extends this representation to human–robot co-training and pairs it with motion-level reasoning, which provides both embodiments with shared language-based targets and improves human-to-robot transfer.

Action Representations for VLAs. VLAs commonly supervise a VLM with discretized action tokens (Brohan et al., 2023; Kim et al., 2024), sometimes alongside a separate continuous action expert (Intelligence et al., 2025). The discrete target is especially consequential when gradients from the action expert do not reach the VLM backbone (Driess et al., 2025). Learned tokenizers such as FAST, OAT, ActionCodec, and X-Tokenizer optimize action compression or token structure (Pertsch et al., 2025; Liu et al., 2026; Dong et al., 2026; Kang et al., 2026); LAP instead describes motion in natural language to remain compatible with the VLM's pretrained semantic space (Zha et al., 2026). EgoLAP applies this representation to bimanual human–robot co-training and shows that motion-level reasoning yields larger gains with language actions than with non-linguistic action tokens.

Reasoning for Visuomotor Control. Prior work augments visuomotor policies with textual reasoning over goals, subtask plans, and spatial cues, or with visual intermediates such as affordances and 2D point traces (Zawalski et al., 2025; Sun et al., 2024; Zhao et al., 2025; Lee et al., 2025; Yuan et al., 2026; Chen et al., 2025b; Sun et al., 2026). With human video, plans, reasoning labels, and dense annotations improve policy performance under distribution shift (Mu et al., 2023; Fu et al., 2025; Kim et al., 2026a; Clark et al., 2025). These results establish the value of reasoning supervision, but leave unresolved which reasoning content best supports cross-embodiment motion transfer. For example, DeMiAn finds that no single annotation type performs best across tasks (Kim et al., 2026a), while a large-scale study finds no benefit from conditioning actions on chain-of-thought traces (Lin et al., 2026).

This gap motivates *motion-level reasoning*: supervision that captures the embodiment-invariant physical rationale underlying human and robot motions, thereby bridging the two data sources. Unlike prior approaches (Zawalski et al., 2025; Chen et al., 2025b), EgoLAP requires neither multi-stage distillation nor text generation at inference, simplifying training and ensuring fast inference.

3 METHOD

EgoLAP is a VLA pre-training framework for transferring motion knowledge from large-scale egocentric human experience to robot manipulation. Its central design choice is to map human-hand and robot-end-effector trajectories to the same language-based interface. We first formalize this interface (Sec. 3.1) and describe the model architecture and training objective (Sec. 3.2). We then detail the construction of language actions (Sec. 3.3) and motion-level reasoning targets (Sec. 3.4).

3.1 Problem Formulation

Let $\mathcal{D}_R$ contain robot samples $(o_t, \ell, s_t, a_{t:t+K})$, where o_t contains one or more RGB observations, ℓ is the task instruction, s_t is the proprioceptive state, and $a_{t:t+K}$ is an action chunk of horizon K. Let $\mathcal{D}_H$ contain egocentric human samples $(o_t, \ell, q_{t:t+K})$, where q denotes one or two tracked hand poses in the head-camera frame. For human samples, the current pose q_t supplies the state input s_t, and the relative motion between successive poses defines the continuous action chunk $a_{t:t+K}$.

Each sample provides up to three training signals. First, a deterministic canonicalizer G_e, specialized only by embodiment metadata e — single arm robot, bimanual robot, or human data, produces a shared discrete action-token target m_t:

$$m_t = G_e(q_{t:t+K}) \quad \text{or} \quad m_t = G_e(a_{t:t+K}, s_t)\,. \tag{1}$$

In EgoLAP, m_t is a language action summarizing one second of hand or end-effector motion in an egocentric frame (Fig. 2; Sec. 3.3). Controlled baselines replace these discrete action tokens with FAST or OAT tokens while preserving the remaining training recipe. Second, a subset of samples includes a motion rationale r_t that grounds m_t in visible spatial relations, contact evidence, object affordances, and the intended physical effect of the motion (Fig. 2; Sec. 3.4). Third, the continuous action chunk $a_{t:t+K}$ directly supervises real-time control.

The model consists of a vision–language backbone f_θ and a continuous action expert v_ϕ. The backbone predicts m_t and, when available, r_t on both human and robot data, thereby aligning their motion semantics. The action expert learns $a_{t:t+K}$ through flow matching. We stop action-loss gradients at the backbone boundary (Driess et al., 2025), insulating the language-pretrained representation from the randomly initialized expert. At inference, only the action expert is rolled out to generate control commands; neither a language action nor a rationale is decoded.

3.2 Model Architecture

EgoLAP uses a Mixture-of-Transformers (MoT) architecture (Liang et al., 2025), pairing a PaliGemma-2B vision–language backbone (Beyer et al., 2024) with a 300M-parameter action expert initialized from scratch. Appendix A.1 provides the full architecture diagram and tensor shapes.

Inputs. The VLM receives up to three RGB images—one base-camera view and two wrist-camera views—together with the task instruction and proprioceptive state. Missing camera views are zero-padded. Following $\pi_{0.5}$, the normalized state is discretized into 256 uniform bins and appended to the prompt as text. For our Cartesian policies, the state encodes end-effector position, continuous 6D rotation (Zhou et al., 2020), and gripper state.



Figure 3: **Attention and gradient structure.** Orange marks allowed attention, gray blocked attention, and blue attention whose gradients are stopped at the VLM boundary. Reasoning and language action tokens are causally masked and condition on the observation prefix; action-expert tokens never attend to them, so control requires no text generation.

Attention and gradient structure. Figure 3 shows the structured attention mask used by the policy. Image, instruction, and state tokens form a bidirectional prefix. Reasoning and language action tokens attend to this prefix and are casual within themselves. The action expert attends to its own tokens and to the observation–instruction–state prefix, but it cannot attend to either reasoning or language action tokens. Consequently, during inference time, no text generation is required. Knowledge insulation stops gradients from the action expert before they enter the VLM (Driess et al., 2025); the backbone is therefore only optimized by the textual targets, while the action expert learns to consume its representations.

Training objective. Let D denote the action dimension. The action expert, parameterized by flow matching (Intelligence et al., 2025), predicts $a_{t:t+K} \in \mathbb{R}^{K \times D}$ in delta end-effector pose space,

and is optimized by a flow matching loss $\mathcal{L}_{\mathrm{FM}}$. Let $\mathcal{L}_{\mathrm{text}}$ denote the expected masked token-level objective for language action and motion-level reasoning, and let λ_{text} weight the two objectives. The complete training loss is

$$\mathcal{L} = \mathbb{E}_{(x_t, a_{t:t+K}, m_t, r_t)\sim p_{\mathrm{mix}}} \left[\lambda_{\mathrm{text}}\mathcal{L}_{\mathrm{text}} + \mathcal{L}_{\mathrm{FM}}\right] \tag{2}$$

where p_{mix} is the fixed human–robot sampling distribution. In our model, $K{=}30$ is the action horizon. We set $\lambda_{\mathrm{text}}{=}0.8$, slightly downweighting the autoregressive VLM objective relative to continuous flow matching to align their convergence speed.

At inference, the action expert starts from Gaussian noise at $\tau{=}1$ and integrates its learned velocity field to $\tau{=}0$, conditioned only on the current observation, instruction, and state. Because the textual targets are absent from the expert's attention context, EgoLAP retains real-time continuous control without test-time language generation. Appendix A.3 gives the complete flow-matching and inference specification, while Appendix A.4 defines the cross-entropy loss $\mathcal{L}_{\mathrm{text}}$.

3.3 Language Actions

Following LAP (Zha et al., 2026), we supervise the VLM backbone with language actions: deterministic natural-language descriptions of hand or end-effector motion over a fixed horizon. Any trajectory with known Cartesian poses can be converted automatically; this conversion requires no manual labeling, VLM labeling, or learned action tokenizer.

Egocentric canonicalization. Raw trajectories cannot share a representation unless their coordinate conventions are aligned. For each sample, we define an egocentric action frame $F_t = (R_t^F, p_t^F)$ at the start of the prediction window. We use the end-effector frame for single-arm robots with a wrist camera, the base frame for bimanual robots and single-arm robots without a wrist camera, and the head-mounted camera frame for egocentric human data. Each action frame is thus approximately aligned with its corresponding egocentric image frame—the wrist camera for single-arm robots, the base camera for bimanual robots, and the head-mounted camera for humans. Because these camera-centric frames follow broadly similar viewing conventions, motion directions are more consistent across embodiments than they would be in embodiment-specific world coordinates.

For an end-effector or hand pose (R_τ, p_τ), we express the pose in this frame as

$$\bar{p}_\tau = (R_t^F)^\top (p_\tau - p_t^F), \qquad \bar{R}_\tau = (R_t^F)^\top R_\tau. \tag{3}$$

We compute the net translation $\Delta p = \bar{p}_{t+1\mathrm{s}} - \bar{p}_t$ and relative rotation $\Delta R = \bar{R}_t^\top \bar{R}_{t+1\mathrm{s}}$ over a one-second window. For bimanual samples, we apply the same procedure independently to the left and right trajectories and retain arm identity. Rotations are converted to extrinsic XYZ Euler angles before serialization.

Deterministic language actions. We serialize $(\Delta p, \Delta R, \Delta c)$—where Δc denotes a gripper or hand-state transition—using the structured grammar from LAP:

$$m_t^{\mathrm{lang}} = \text{“}\langle\text{verb}\rangle\ \langle\text{direction}\rangle\ \langle\text{magnitude}\rangle\ \langle\text{unit}\rangle\text{”}. \tag{4}$$

Here, *move* denotes translation, *tilt* denotes roll or pitch, and *rotate* denotes yaw. Translational magnitudes are rounded to integer centimeters and rotational magnitudes to multiples of five degrees; components that round to zero are omitted. We append an open, close, or maintain-state phrase for the gripper or hand and use explicit *left arm* and *right arm* clauses for bimanual motion. For example: "Left arm: move forward 5 cm, move left 2 cm, tilt left 10 degrees, rotate clockwise 10 degrees, close gripper. Right arm: move down 10 cm, tilt forward 15 degrees, open gripper."

Reinforcement learning on language actions. Because G_e is deterministic and the grammar is machine-parseable, a language action is a *verifiable* target: any sampled completion can be parsed and scored against the demonstrated motion with no learned reward model. We use this to post-train EgoLAP with GRPO (Shao et al., 2024), updating only the language backbone while the vision encoder and action expert stay frozen. Appendix A.9 gives the reward and full protocol, and results are presented in Sec.4.5.

3.4 Motion-Level Reasoning

A language action specifies *what motion occurs*, but not *why that motion is appropriate in the current physical state*. We therefore annotate subsets of human and robot trajectories with a motion rationale

r_t. Given the task instruction, ordered trajectory frames, recent action history, and the ground-truth language action at the current frame, we produce a rationale that identifies a concrete end-effector–object relation, relevant contact or affordance evidence, and the immediate physical effect of the labeled motion. For example, a rationale for kneading dough might read:

> *The dough mass rests at the base of the metal bowl, with both palms directly contacting the top and sides of the dough. Kneading requires deforming the dough against the bowl surface. Both hands translate dominantly backward while tilting wrists up and opening fingers, pulling across the compliant dough toward the body before the next pressing stroke.*

Motion-level reasoning is deliberately local and physics-focused. It differs from the high-level task or subtask reasoning studied in prior work (Zawalski et al., 2025; Chen et al., 2025b; Sun et al., 2026; 2024): it lives on a lower hierarchy and directly connects to low-level actions, describing the underlying motion intent shared between embodiments. Appendix A.5 describes the annotation procedure, and Appendix D provides generated robot and human examples together with the labeling prompt. We apply reasoning dropout and reasoning loss dropout during training, which is discussed in Appendix A.4.

4 EXPERIMENTS

We organize the evaluation around four questions: 1. Which action representation is most effective when co-training with human data? 2. Does motion-level reasoning improve transfer from human data to robots, and are motion-level reasoning and language actions mutually reinforcing? 3. How does motion-level reasoning compare to established reasoning formats? 4. Can EgoLAP be effectively adapted to downstream tasks using supervised fine-tuning and reinforcement learning?

4.1 EXPERIMENTAL DESIGN

Zero-shot evaluation. In the real world, we evaluate our models and baselines on a customized bimanual YAM robot (Fig. 4 (b)) whose arm placement (e.g., distance between the two arms) and camera configuration differ from every training embodiment. We test five behaviors spanning rigid, articulated, and deformable manipulation: Pick Place (Simple), Fold Pants, Handover & Place, Wipe Object, and Pick Place (Complex). We run 10 trials per task, except Pick Place (Complex), for which we run 12 trials, and report mean task progress. Appendix B describes each task and gives the complete scoring rubric. Error bar is reported as ± 1 SEM.

We complement the real-world experiments with a bimanual simulation benchmark built upon ABC (Allshire et al., 2026), including 14 tasks that span picking, placing, open-ended language following, and cloth folding. Representative tasks are visualized in Fig. 4 (a). Notably, none of our models is trained on simulation data, so this is a zero-shot simulation evaluation. We evaluate each model for 100 episodes per task, 1,400 episodes in total, and report the mean success rate for each task group. Appendix B reports the suite composition and full evaluation protocol.

Baselines. We instantiate three policy families differing only in action representation. EgoLAP uses language actions and motion-level reasoning; EgoFAST replaces language actions with FAST tokens (Pertsch et al., 2025) and uses the same motion-level reasoning; and EgoOAT instead uses OAT tokens (Liu et al., 2026) plus motion-level reasoning. We re-train the OAT tokenizer on our data mixture. In addition, we compare to LAP (Zha et al., 2026), which is EgoLAP trained without reasoning and without human data, as shown in Fig. 5. In Sec.4.4, we compare EgoLAP with our reimplemented ECoT baselines, which replaces motion-level reasoning with subtask plan, bounding box and visual trace reasoning. Appendix A.8 discusses the implementation details for baselines.

Unless stated otherwise, all variants share everything except the action representation — same backbone, same architecture, same data mixture, same data pre-processing (e.g., egocentric canonicalization in Sec. 3.3), same optimization budget, and same evaluation protocol. All action representations correspond to the same one-second action chunk. We train every variant for 40k steps and select its checkpoint with the same validation procedure before the final evaluation; the only exception is EgoLAP, which converges faster and we train it up to 30k steps (Appendix C). Appendix A reports the shared training recipe and data mixture. Notably, within our matched implementation, EgoFAST reproduces the discrete action-token interface of $\pi_{0.5}$ (Intelligence et al., 2025), differing primarily

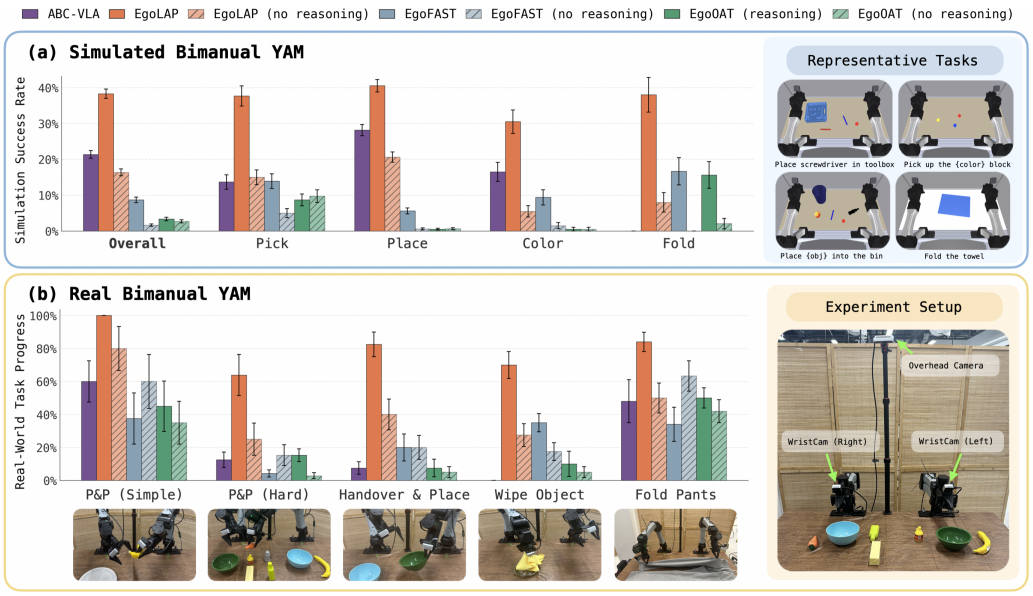


Figure 4: **Zero-shot evaluation on bimanual YAM robots.** EgoLAP achieves 80.1% mean progress on real-world tasks—a 2.3× gain over alternative action representations—and a 38.3% success rate on simulated tasks, outperforming ABC-VLA (Allshire et al., 2026) by 1.8×. When paired with language actions, motion-level reasoning yields a 35.6-point improvement, substantially greater than the gains achieved with other action representations.

in its training mixture. We also compare with the released checkpoint for ABC-VLA (Allshire et al., 2026), which is trained directly on the ABC YAM platform.

4.2 Language Actions Transfer Human Experience Most Effectively

We first show that EgoLAP benefits from human data. Holding the representation fixed, we compare EgoLAP (no reasoning) trained on human plus robot data against the same policy trained on robot data alone, i.e., LAP Zha et al. (2026) trained on our data mixture. Adding human data lifts simulation success from 7.2% to 16.4%, a 2.3× gain (Fig. 5) The simulation suite is the harder test of transfer, since neither policy has ever seen the simulator, and it is there that human data pays off most. Appendix A.8 specifies the matched robot-only training setup.

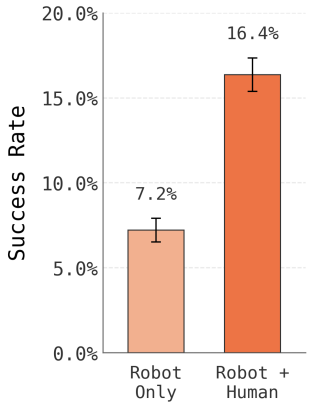


Figure 5: **Human data helps with EgoLAP.** Adding egocentric human data more than doubles simulation success for the same language action policy.

We next compare action representations. Changing only the action representation and using no reasoning, EgoLAP reaches 16.4% success in simulation and 44.5% progress in the real world, against 1.6% and 35.2% for EgoFAST and 2.7% and 18.0% for EgoOAT. The margin is largest exactly where the domain gap is biggest: on the unseen simulator, language actions are an order of magnitude better than FAST and OAT. This strongly suggests that language actions transfer human experience to robot tasks better than non-linguistic action tokenizers do, even without any reasoning supervision. We believe this is because language actions align with the VLM's pre-training distribution, so human and robot motions that share an intent are described by the same words and land in the same region of a representation space the backbone already understands. The full EgoLAP policy also beats ABC-VLA, which is trained directly on the data from the same simulation environment, by 16.9 points in simulation and 54.5 points in the real world. This strongly suggests EgoLAP overcomes an in-domain data advantage and generalizes better by transferring human data knowledge.

4.3 Motion-Level Reasoning and Language Actions Are Mutually Reinforcing

We further compare each EgoX (no reasoning) policy with its reasoning-supervised counterpart. Motion-level reasoning raises EgoLAP from 44.5% to 80.1% mean progress in the real world, a 35.6-point gain that holds on every one of the five tasks (Fig. 4 (b)), and from 16.4% to 38.3% in simulation, a 21.9-point gain that holds on every task family (Fig. 4 (a)). This first suggests that motion-level reasoning is very helpful on its own. The same supervision, however, adds only 7.1 points to EgoFAST in simulation and in fact *hurts* it in the real world, dropping it from 35.2% to 26.1%. EgoOAT benefits more modestly, gaining 0.7 points in simulation and 7.6 points in the real world.

This gap strongly suggests a synergy between language actions and motion-level reasoning. As shown in Fig. 2, the two form a single textual "action chain-of-thought", moving from reasoning about *why* a motion is appropriate, grounded in visuals and physics, to the actual motion to execute. Non-linguistic action tokens like FAST and OAT lack this semantic correspondence, leaving the backbone to infer the correlation from co-occurrence alone, without transferring the knowledge already in the VLM backbone.

4.4 Motion-Level Reasoning Outperforms Established Reasoning Formats

We next ask whether the benefit comes from motion-level reasoning specifically or from reasoning tokens in general. We compare against a strong composite target in the style of ECoT (Zawalski et al., 2025), which predicts the current subtask, visible-object bounding boxes, and a visual trace before the language action. Motion-level reasoning helps every target we tried (Fig. 6): it adds 7.1 points to FAST, 9.1 points to the ECoT composite, and 21.9 points to plain language actions. The composite itself, however, does not help. On its own it reaches 5.3%, well below the 16.4% of language actions with no reasoning at all, and stacking it in front of motion-level reasoning caps the policy at 14.4%, less than half of the 38.3% obtained by motion-level reasoning alone.

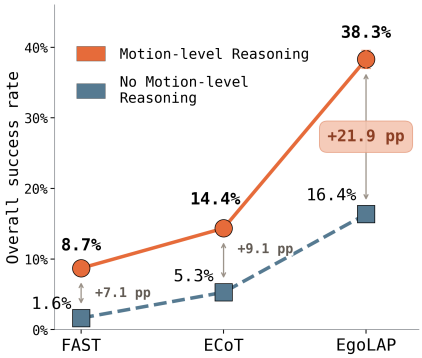


Figure 6: **Motion-level reasoning is what carries the gain.** Motion-level reasoning outperforms ECoT-style reasoning (Zawalski et al., 2025), and augments it by a large margin.

Because all action-representation variants are trained on the same trajectories with identical action-conditioned rationales, and the ECoT-style control retains the same language-action target while adding subtask, box, and future-trace supervision, EgoLAP's substantially larger gain cannot be attributed to privileged action information or merely to additional auxiliary supervision; it specifically demonstrates the effectiveness of pairing motion-level rationales with a language-based action interface.

4.5 Adapting EgoLAP to Downstream Tasks

Following Sec. 3.4, we post-train EgoLAP with GRPO (Shao et al., 2024) over the language actions on ABC (Allshire et al., 2026), keeping the vision and action expert frozen. We construct the language action reward by summing the negative squared translation, geodesic rotation, and a binary gripper error of the parsed language action. This can be seen as an indirect behavior cloning alignment reward of the language actions. We test three variations: reasoning pre-training without reasoning generation (i.e., only generates language actions during sampling), reasoning pre-training with reasoning generation, and the non-reasoning baseline (w/o reasoning). Appendix A.9 gives the reward, objective, and full protocol.

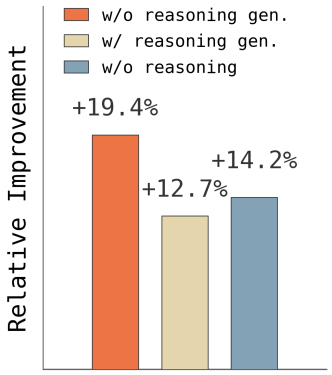


Figure 7: GRPO over language actions over simulation tasks.

Even with a frozen expert, RL improves simulation success in all three arms (Fig. 7). Comparing the orange bar and the blue

bar suggests reasoning pre-training supplies a stronger initialization for a stronger post-RL policy. Secondly, generating reasoning during RL sampling *hurts* performance relative to only sampling language actions ($+12.7\%$ vs. $+19.4\%$): ABC rationales were never seen in training, so the sampled rationales are out of distribution, which makes the subsequent language actions generation also worse. We leave investigating how to better leverage motion-level reasoning for RL in future works.

We further evaluate downstream adaptation by supervised fine-tuning on RoboTwin 2.0 (Chen et al., 2025a) across four tasks with increasing difficulty (Tab. 1). FAST benefits less from fine-tuning and only performs well under one of the seen tasks. In contrast, EgoLAP adapts more effectively, achieving higher success rates on most seen and unseen task instances. The gains are particularly consistent on unseen instances, suggesting that EgoLAP's structured language action space supports better generalization during adaptation. This indicates that EgoLAP is not only effective for RL post-training, but also provides a more favorable representation for downstream fine-tuning. See details in Appendix A.10.

	Seen		Unseen	
Task	**FAST**	**EgoLAP**	**FAST**	**EgoLAP**
Handover Mic	**70**	58	26	**44**
Handover Block	22	**40**	13	**34**
Shake Bottle	80	**96**	80	**97**
Turn Switch	34	**64**	26	**31**

Table 1: **Downstream fine-tuning on RoboTwin 2.0.** Success rate ($\%$) after supervised fine-tuning on the RoboTwin 2.0 dataset (Chen et al., 2025a), evaluated on seen and unseen task instances.

4.6 Reasoning Aligns Skill Structure Across Embodiments

We test matched checkpoints differing only in reasoning supervision. We mean-pool final-layer features from $1{,}002$ balanced human and robot frames, subtract each embodiment mean, and perform cross-embodiment cosine retrieval. In Tab. 2, R$\rightarrow$H means a robot query ranks human frames; H$\rightarrow$R is the reverse. "3w" and "2w" denote three- and two-skill splits; R@1 is the fraction whose nearest frame has the same skill; MRR is the mean reciprocal rank of the first same-skill match; "w/o" is the matched checkpoint without reasoning; and Δ is the absolute gain from reasoning in percentage points. We report paired task-bootstrap confidence intervals.

Table 2: **Motion-level reasoning improves cross-embodiment skill retrieval.** All values are percentages.

Retrieval	w/o	EgoLAP	Δ
R$\rightarrow$H · 3w R@1	51	**76**	$\mathbf{+25}$
H$\rightarrow$R · 3w R@1	42	**63**	$\mathbf{+21}$
R$\rightarrow$H · 2w R@1	69	**92**	$\mathbf{+23}$
R$\rightarrow$H · 3w MRR	61	**80**	$\mathbf{+19}$

Robot-to-human R@1 rises from 51% to 76% (+25 points, 95% CI $[16, 39]$, $p < .005$), while the reverse direction rises from 42% to 63% (+21 points, $p = .04$). The two-skill split improves from 69% to 92% (+23 points, 95% CI $[16, 30]$), and MRR from 61% to 80% (Tab. 2). Because the embodiments remain linearly separable, reasoning preserves domain identity but gives both domains the same internal organization—fold near fold, place near place.

5 CONCLUSION

We introduce EgoLAP, a VLA pre-training framework that transfers egocentric human experience to robots through a shared language of motion. By pairing language actions that describe *what* motion to produce with physically grounded reasoning that explains *why*, EgoLAP reaches 80.1% real-world task progress and 38.3% simulation success on bimanual manipulation, substantially outperforming alternative action representations and reasoning formats. Beyond policy performance, motion-level reasoning improves cross-embodiment skill retrieval by up to 25 points, suggesting that it organizes human and robot experience by shared skill structure without erasing embodiment identity. EgoLAP also adapts effectively through supervised fine-tuning and reinforcement learning, while its textual targets remain training-only and add no generation cost during control.

AI USE STATEMENT

We used large language models to edit and polish the manuscript. We also used Google Gemini to generate offline motion-level reasoning labels: `gemini-3.5-flash` for DROID and `gemini-3.7-flash` for MECKA. These labels serve as training targets for the reasoning-enabled policies; Appendix A.5 documents the inputs and annotation procedure, and Appendix D provides the prompts and representative outputs. The authors designed and checked the labeling pipeline, reviewed the AI-assisted manuscript edits, and take responsibility for all content, claims, results, and AI-assisted artifacts in this work.

REPRODUCIBILITY STATEMENT

We provide the methodological and experimental details needed to reproduce our results. Section 3 specifies the model, objectives, shared motion representation, and motion-level reasoning formulation, while Section 4.1 describes the matched comparison protocol. Appendix A documents the architecture, prompt and action serialization, flow-matching objective, reasoning-annotation pipeline, training hyperparameters, data mixture, and baseline implementations. Appendix B specifies the evaluation tasks and number of episodes, Appendix A.9 gives the reinforcement-learning setup, and Appendix C reports additional results and checkpoint-selection details. The complete Gemini labeling prompts and representative annotations are included in Appendix D.

ETHICS STATEMENT

All authors have read and adhered to the ICLR Code of Ethics. This work uses previously collected human and robot datasets and does not introduce new human-subject data collection. Because egocentric video can contain privacy-sensitive information and reflect demographic, environmental, and behavioral biases in its source datasets, researchers should follow the datasets' licenses, consent conditions, and privacy safeguards and should not assume that the reported results generalize beyond the represented settings. The motion-level reasoning labels are generated by Gemini and may inherit factual errors or social biases from the annotator; we disclose this use, document the prompts and procedure, and treat the labels as imperfect supervision. Finally, learned robot policies can fail unpredictably or be misused, so the models should undergo task-specific safety testing and human oversight before deployment around people or outside controlled research environments.

REFERENCES

Arthur Allshire, Himanshu Gaurav Singh, Ritvik Singh, Adam Rashid, Hongsuk Choi, David McAllister, Justin Yu, Yiyuan Chen, Huang Huang, Pieter Abbeel, Xi Chen, Rocky Duan, Phillip Isola, Jitendra Malik, Fred Shentu, Guanya Shi, Philipp Wu, and Angjoo Kanazawa. Scalable behavior cloning with open data, training, and evaluation, 2026. URL https://arxiv.org/abs/2606.27375.

Shikhar Bahl, Abhinav Gupta, and Deepak Pathak. Human-to-robot imitation in the wild, 2022. URL https://arxiv.org/abs/2207.09450.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. Paligemma: A versatile 3b vlm for transfer, 2024. URL https://arxiv.org/abs/2407.07726.

Homanga Bharadhwaj, Abhinav Gupta, Shubham Tulsiani, and Vikash Kumar. Zero-shot robot manipulation from passive human videos, 2023. URL https://arxiv.org/abs/2302.02011.

Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation, 2024a. URL https://arxiv.org/abs/2409.16283.

Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation, 2024b. URL https://arxiv.org/abs/2405.01527.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023. URL https://arxiv.org/abs/2307.15818.

Xiongyi Cai, Ri-Zhao Qiu, Geng Chen, Lai Wei, Isabella Liu, Tianshu Huang, Xuxin Cheng, and Xiaolong Wang. In-n-on: Scaling egocentric manipulation with in-the-wild and on-task data, 2025. URL https://arxiv.org/abs/2511.15704.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation, 2025a. URL https://arxiv.org/abs/2506.18088.

William Chen, Suneel Belkhale, Suvir Mirchandani, Oier Mees, Danny Driess, Karl Pertsch, and Sergey Levine. Training strategies for efficient embodied reasoning, 2025b. URL https://arxiv.org/abs/2505.08243.

Jaden Clark, Suvir Mirchandani, Dorsa Sadigh, and Suneel Belkhale. Action-free reasoning for policy generalization, 2025. URL https://arxiv.org/abs/2502.03729.

Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homanga Bharadhwaj, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jay Vakil, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin

Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Muhammad Zubair Irshad, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Martín-Martín, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shubham Tulsiani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vikash Kumar, Vincent Vanhoucke, Vitor Guizilini, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open x-embodiment: Robotic learning datasets and rt-x models, 2025. URL https://arxiv.org/abs/2310.08864.

Zibin Dong, Yicheng Liu, Shiduo Zhang, Baijun Ye, Yifu Yuan, Fei Ni, Jingjing Gong, Xipeng Qiu, Hang Zhao, Yinchuan Li, and Jianye Hao. Actioncodec: What makes for good action tokenizers, 2026. URL https://arxiv.org/abs/2602.15397.

Danny Driess, Jost Tobias Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Z. Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, and Sergey Levine. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better, 2025. URL https://arxiv.org/abs/2505.23705.

Yankai Fu, Ning Chen, Junkai Zhao, Shaozhe Shan, Guocai Yao, Pengwei Wang, Zhongyuan Wang, and Shanghang Zhang. Metis: Multi-source egocentric training for integrated dexterous vision-language-action model, 2025. URL https://arxiv.org/abs/2511.17366.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18995–19012, 2022. URL https://arxiv.org/abs/2110.07058.

Shresth Grover, Akshay Gopalkrishnan, Bo Ai, Henrik I. Christensen, Hao Su, and Xuanlin Li. Enhancing generalization in vision-language-action models by preserving pretrained representations, 2025. URL https://arxiv.org/abs/2509.11417.

Asher J. Hancock, Xindi Wu, Lihan Zha, Olga Russakovsky, and Anirudha Majumdar. Actions as language: Fine-tuning vlms into vlas without catastrophic forgetting, 2025. URL https://arxiv.org/abs/2509.22195.

Ryan Hoque, Peide Huang, David J. Yoon, Mouli Sivapurapu, and Jian Zhang. Egodex: Learning dexterous manipulation from large-scale egocentric video, 2026. URL https://arxiv.org/abs/2505.11709.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh,

Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025. URL https://arxiv.org/abs/2504.16054.

Vidhi Jain, Maria Attarian, Nikhil J Joshi, Ayzaan Wahid, Danny Driess, Quan Vuong, Pannag R Sanketi, Pierre Sermanet, Stefan Welker, Christine Chan, Igor Gilitschenski, Yonatan Bisk, and Debidatta Dwibedi. Vid2robot: End-to-end video-conditioned policy learning with cross-attention transformers, 2024. URL https://arxiv.org/abs/2403.12943.

Miracle Kang, Lights Shi, Lucy Liang, Roy Gan, Dongxiu Liu, Pushi Zhang, Sylas Chen, Shawn Qin, Yinan Zheng, Jinliang Zheng, Hao Wang, Xianyuan Zhan, and Hang Su. X-tokenizer: A multimodal action tokenizer for vision-language-action pretraining, 2026. URL https://arxiv.org/abs/2606.14752.

Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. Egomimic: Scaling imitation learning via egocentric video, 2024. URL https://arxiv.org/abs/2410.24221.

Simar Kareer, Karl Pertsch, James Darpinian, Judy Hoffman, Danfei Xu, Sergey Levine, Chelsea Finn, and Suraj Nair. Emergence of human to robot transfer in vision-language-action models, 2025. URL https://arxiv.org/abs/2512.22414.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, Vitor Guizilini, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Muhammad Zubair Irshad, Donovon Jackson, Charlotte Le, Yunshuang Li, Kevin Lin, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. Droid: A large-scale in-the-wild robot manipulation dataset, 2025. URL https://arxiv.org/abs/2403.12945.

Bosung Kim, Ruiyi Wang, David Acuna, Jaehun Jung, Alexander Trevithick, Brandon Cui, Yejin Choi, and Prithviraj Ammanabrolu. How to instruct your robot: Dense language annotations power robot policy learning, 2026a. URL https://arxiv.org/abs/2605.17077.

Ji Woong Kim, Ke Wang, Zipeng Fu, Sirui Chen, Cong Zhao, Jeff Lai, and Chelsea Finn. Ego-pi: Vla fine-tuning for ego-centric human and robot data, 2026b. URL https://arxiv.org/abs/2606.08107.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model, 2024. URL https://arxiv.org/abs/2406.09246.

Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, Winson Han, Wilbert Pumacay, Angelica Wu, Rose Hendrix, Karen Farley, Eli VanderBilt, Ali Farhadi, Dieter Fox, and Ranjay Krishna. Molmoact: Action

reasoning models that can reason in space, 2025. URL https://arxiv.org/abs/2508.07917.

Marion Lepert, Jiaying Fang, and Jeannette Bohg. Masquerade: Learning from in-the-wild human videos using data-editing, 2025. URL https://arxiv.org/abs/2508.09976.

Marion Lepert, Jiaying Fang, and Jeannette Bohg. Phantom: Training robots without robots using only human videos, 2026. URL https://arxiv.org/abs/2503.00779.

Baoyu Li, Xinchen Yin, Mengying Lin, Yixin Zhang, and Danfei Xu. Egowam: World action models beyond pixels with in-the-wild egocentric human data, 2026. URL https://arxiv.org/abs/2607.08436.

Weixin Liang, Lili Yu, Liang Luo, Srinivasan Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen tau Yih, Luke Zettlemoyer, and Xi Victoria Lin. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models, 2025. URL https://arxiv.org/abs/2411.04996.

Fanqi Lin, Kushal Arora, Jean Mercat, Haruki Nishimura, Paarth Shah, Chen Xu, Mengchao Zhang, Mark Zolotas, Maya Angeles, Owen Pfannenstiehl, Andrew Beaulieu, and Jose Barreiros. A systematic study of data modalities and strategies for co-training large behavior models for robot manipulation, 2026. URL https://arxiv.org/abs/2602.01067.

Chaoqi Liu, Yue Zhao, Haonan Chen, Xiaoshen Han, Jiawei Gao, Ehsan Adeli, and Yilun Du. Ordered action tokens for visuomotor policy learning, 2026. URL https://arxiv.org/abs/2607.21670.

Hao Luo, Ye Wang, Wanpeng Zhang, Sipeng Zheng, Ziheng Xi, Chaoyi Xu, Haiweng Xu, Haoqi Yuan, Chi Zhang, Yiqing Wang, Yicheng Feng, and Zongqing Lu. Being-h0.5: Scaling human-centric robot learning for cross-embodiment generalization, 2026. URL https://arxiv.org/abs/2601.12993.

Yao Mu, Qinglong Zhang, Mengkang Hu, Wenhai Wang, Mingyu Ding, Jun Jin, Bin Wang, Jifeng Dai, Yu Qiao, and Ping Luo. Embodiedgpt: Vision-language pre-training via embodied chain of thought, 2023. URL https://arxiv.org/abs/2305.15021.

Tushar Nagarajan, Christoph Feichtenhofer, and Kristen Grauman. Grounded human-object interaction hotspots from video, 2019. URL https://arxiv.org/abs/1812.04558.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models, 2025. URL https://arxiv.org/abs/2501.09747.

Ryan Punamiya, Dhruv Patel, Patcharapong Aphiwetsa, Pranav Kuppili, Lawrence Y. Zhu, Simar Kareer, Judy Hoffman, and Danfei Xu. Egobridge: Domain adaptation for generalizable imitation from egocentric human data, 2025. URL https://arxiv.org/abs/2509.19626.

Ryan Punamiya, Simar Kareer, Zeyi Liu, Josh Citron, Ri-Zhao Qiu, Xiongyi Cai, Alexey Gavryushin, Jiaqi Chen, Davide Liconti, Lawrence Y. Zhu, Patcharapong Aphiwetsa, Baoyu Li, Aniketh Cheluva, Pranav Kuppili, Yangcen Liu, Dhruv Patel, Aidan Gao, Hye-Young Chung, Ryan Co, Renee Zbizika, Jeff Liu, Xiaomeng Xu, Haoyu Xiong, Geng Chen, Sebastiano Oliani, Wenkai Xuan, Chenyu Yang, Xi Wang, James Fort, Richard Newcombe, Josh Gao, Jason Chong, Garrett Matsuda, Aseem Doriwala, Marc Pollefeys, Robert Katzschmann, Xiaolong Wang, Shuran Song, Judy Hoffman, and Danfei Xu. Egoverse: An egocentric human dataset for robot learning from around the world, 2026. URL https://arxiv.org/abs/2604.07607.

Ri-Zhao Qiu, Shiqi Yang, Xuxin Cheng, Chaitanya Chawla, Jialong Li, Tairan He, Ge Yan, David J. Yoon, Ryan Hoque, Lars Paulsen, Ge Yang, Jian Zhang, Sha Yi, Guanya Shi, and Xiaolong Wang. Humanoid policy human policy, 2025. URL https://arxiv.org/abs/2503.13441.

Juntao Ren, Priya Sundaresan, Dorsa Sadigh, Sanjiban Choudhury, and Jeannette Bohg. Motion tracks: A unified representation for human-robot transfer in few-shot imitation learning, 2025. URL https://arxiv.org/abs/2501.06994.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Mohan Kumar Srirama, Sudeep Dasari, Shikhar Bahl, and Abhinav Gupta. Hrp: Human affordances for robotic pre-training, 2024. URL https://arxiv.org/abs/2407.18911.

Nan Sun, Yuan Zhang, Yongkun Yang, Wentao Zhao, Peiyan Li, Jun Guo, Wenxuan Song, Pengxiang Ding, Runze Suo, Yifei Su, Xin Xiao, Xinghang Li, and Huaping Liu. Revisiting embodied chain-of-thought for generalizable robot manipulation, 2026. URL https://arxiv.org/abs/2606.03784.

Qi Sun, Pengfei Hong, Tej Deep Pala, Vernon Toh, U-Xuan Tan, Deepanway Ghosal, and Soujanya Poria. Emma-x: An embodied multimodal action model with grounded chain of thought and look-ahead spatial reasoning, 2024. URL https://arxiv.org/abs/2412.11974.

Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play, 2023. URL https://arxiv.org/abs/2302.12422.

Songlin Wei, Hongyi Jing, Boqian Li, Zhenyu Zhao, Jiageng Mao, Zhenhao Ni, Sicheng He, Jie Liu, Xiawei Liu, Kaidi Kang, Sheng Zang, Weiduo Yuan, Marco Pavone, Di Huang, and Yue Wang. ψ_0: An open foundation model towards universal humanoid loco-manipulation, 2026. URL https://arxiv.org/abs/2603.12263.

Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning, 2024. URL https://arxiv.org/abs/2401.00025.

Ruihan Yang, Qinxi Yu, Yecheng Wu, Rui Yan, Borui Li, An-Chieh Cheng, Xueyan Zou, Yunhao Fang, Xuxin Cheng, Ri-Zhao Qiu, Hongxu Yin, Sifei Liu, Song Han, Yao Lu, and Xiaolong Wang. Egovla: Learning vision-language-action models from egocentric human videos, 2025. URL https://arxiv.org/abs/2507.12440.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi "Jim" Fan, and Joel Jang. World action models are zero-shot policies, 2026. URL https://arxiv.org/abs/2602.15922.

Yifu Yuan, Haiqin Cui, Yaoting Huang, Yibin Chen, Fei Ni, Zibin Dong, Pengyi Li, Yan Zheng, Hongyao Tang, and Jianye Hao. Embodied-r1: Reinforced embodied reasoning for general robotic manipulation, 2026. URL https://arxiv.org/abs/2508.13998.

Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning, 2025. URL https://arxiv.org/abs/2407.08693.

Lihan Zha, Asher J. Hancock, Mingtong Zhang, Tenny Yin, Yixuan Huang, Dhruv Shah, Allen Z. Ren, and Anirudha Majumdar. Lap: Language-action pre-training enables zero-shot cross-embodiment transfer, 2026. URL https://arxiv.org/abs/2602.10556.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, Ankur Handa, Ming-Yu Liu, Donglai Xiang, Gordon Wetzstein, and Tsung-Yi Lin. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models, 2025. URL https://arxiv.org/abs/2503.22020.

Ruijie Zheng, Dantong Niu, Yuqi Xie, Jing Wang, Mengda Xu, Yunfan Jiang, Fernando Castañeda, Fengyuan Hu, You Liang Tan, Letian Fu, Trevor Darrell, Furong Huang, Yuke Zhu, Danfei Xu, and Linxi Fan. Egoscale: Scaling dexterous manipulation with diverse egocentric human data, 2026. URL https://arxiv.org/abs/2602.16710.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks, 2020. URL https://arxiv.org/abs/1812.07035.

A Training Details

A.1 Model Architecture and Tensor Shapes

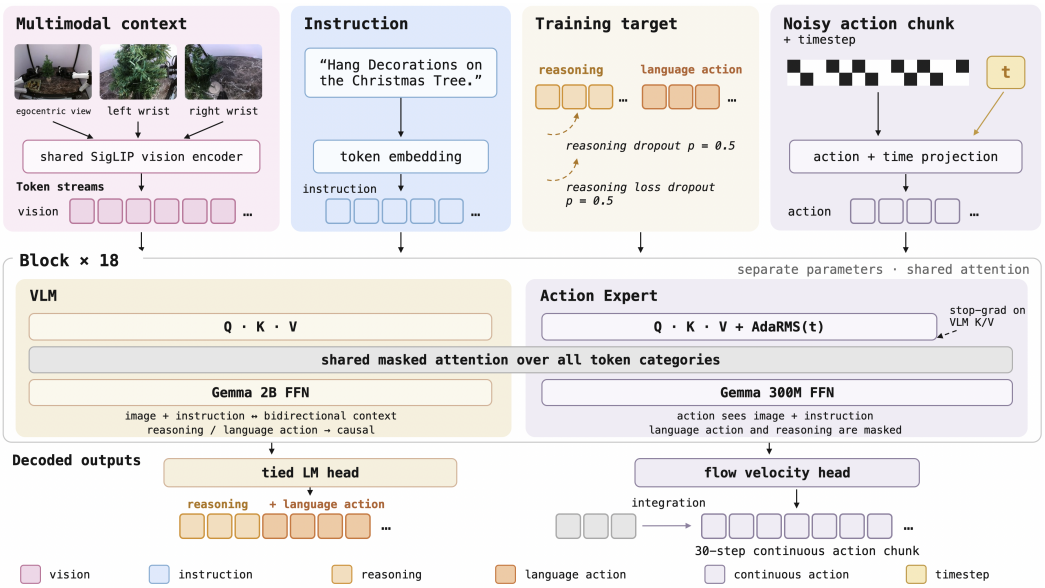


Figure 8: **Detailed EgoLAP architecture.** A shared SigLIP encoder maps each camera view to visual tokens. The 2B VLM branch predicts motion-level reasoning and the language action; the 300M action expert maps a noisy action chunk to its flow velocity. The two branches hold separate parameters but join the same masked-attention operation at every transformer layer. Action tokens attend to the observation, instruction, and discretized state, never to the textual targets, and gradients through the VLM keys and values are stopped.

Let B be the batch size and V the number of enabled camera slots. Before tokenization, a sample contains RGB observations $o_t \in \mathbb{R}^{B\times V\times 224\times 224\times 3}$, a task instruction, a normalized proprioceptive state $s_t \in \mathbb{R}^{B\times 20}$, and a continuous target $a_{t:t+30} \in \mathbb{R}^{B\times 30\times 14}$. The 20 state values comprise ten values per arm: end-effector position, a continuous 6D rotation, and gripper state. The 14 action values comprise seven Cartesian-control values per arm; single-arm samples occupy one seven-dimensional block and pad the other. For samples with fewer than three views, absent views are zero-filled and masked.

Each 224×224 image produces 256 SigLIP patch tokens. The normalized state is discretized into 256 uniform bins and serialized with the instruction, so it is part of the text prefix rather than a separate action-expert token. For EgoLAP, the padded text tensor has shape $B \times 320$. Thus, with three views, the maximum VLM prefix has shape $B \times (3 \cdot 256 + 320) \times 2048$. Padding and absent-image tokens are removed by the attention mask. The VLM has 18 transformer blocks, width 2048, an FFN width of 16,384, eight query heads, and one key–value head.

The action expert receives $a^{\tau}_{t:t+30} \in \mathbb{R}^{B\times 30\times 14}$ and projects it to $B \times 30 \times 1024$. A sinusoidal embedding of τ produces one $B \times 1024$ conditioning vector used by adaptive RMS normalization in all 18 expert blocks. The expert has FFN width 4096, eight query heads, and one key–value head. Its velocity head returns $v_\phi \in \mathbb{R}^{B\times 30\times 14}$, exactly matching the target action-chunk shape. The tied VLM head produces token logits of shape $B \times L \times 257{,}152$; only reasoning and action-target positions receive text loss. At inference, ten Euler steps integrate a noise tensor of shape $B\times 30\times 14$ to the final action chunk of the same shape. Text is not decoded for control.

A.2 Prompt and Language-Action Format

Prompt serialization. The observation images are followed by a single text sequence with the literal format

Task: [instruction], predict the robot's action in the egocentric frame; State: [b_1 ... b_d]; Answer:

The instruction is stripped of a trailing period, and underscores and line breaks are replaced by spaces. Each $b_i \in \{0, \ldots, 255\}$ is a uniformly discretized normalized state value. Trailing zero-padded state dimensions are omitted, leaving d=10 values for a single arm and d=20 for bimanual robot and human samples. The reference run reuses the exacting phrase `egocentric frame` and `robot's action` for all embodiments, including human samples.

For a sample without motion-level reasoning, the language action m_t immediately follows `Answer:`. When a rationale is included, the complete target is

Answer: [r_t] Concrete actions to take are: [m_t]

The boundary phrase is fixed and separates the optional rationale from the language action, allowing their losses to be masked independently while retaining one causal target sequence. At inference, the policy receives the same observation–instruction–state prefix, but the continuous action expert does not decode either textual field.

Language-action grammar. Each arm is represented by an ordered, comma-separated clause. Translation is emitted in the order longitudinal, vertical, then lateral; rotation follows as roll, pitch, then yaw; and the clause always ends with the final gripper state. Table 3 gives the exact vocabulary.

Table 3: Canonical component vocabulary for language actions. Here n is an integer number of centimeters and θ is an integer multiple of 5 degrees. A component is omitted when its rounded magnitude is zero.

Component	Positive direction	Negative direction
Translation x	`move forward` n `cm`	`move back` n `cm`
Translation z	`move up` n `cm`	`move down` n `cm`
Translation y	`move left` n `cm`	`move right` n `cm`
Roll	`tilt left` θ `degrees`	`tilt right` θ `degrees`
Pitch	`tilt back` θ `degrees`	`tilt forward` θ `degrees`
Yaw	`rotate counterclockwise` θ `degrees`	`rotate clockwise` θ `degrees`
Gripper state	`open gripper`	`close gripper`

Translational displacement is rounded to integer centimeters, and each Euler-angle component is rounded to the nearest 5 degrees. The open/close token describes the final aperture state rather than a displacement and is therefore retained even when all pose components are omitted. The embodiment-specific serializations are as follows.

Single-arm robot. One clause is emitted, with its arm prefix sampled uniformly during training:

[Left|Right] arm: [motion components], [open|close] gripper

For example, a DROID sample in the reference run is serialized as `Left arm: move forward 8 cm, move down 6 cm, move right 7 cm, tilt forward 10 degrees, rotate counterclockwise 10 degrees, close gripper.`

Bimanual robot. Both clauses are emitted in a fixed order and separated by a period:

Left arm: [left components]. Right arm: [right components]

Egocentric human. Human hand trajectories use the same bimanual template rather than a separate human-only vocabulary. `Left arm` and `Right arm` identify the tracked hands, while `open gripper` and `close gripper` encode the corresponding binary hand-aperture state. This shared serialization provides a common interface through which human and robot motion supervise the same VLM tokens.

A.3 Flow-Matching Objective

For a continuous action chunk $a_{t:t+K} \in \mathbb{R}^{K\times D}$, Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, and interpolation time $\tau = 0.001 + 0.999z$ with $z \sim \mathrm{Beta}(1.5, 1)$, define the interpolated state and target velocity:

$$a^{\tau}_{t:t+K} = (1-\tau)a_{t:t+K} + \tau\epsilon, \qquad u_t = \epsilon - a_{t:t+K}. \tag{5}$$

The action expert v_ϕ is trained to recover this velocity:

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{\tau,\epsilon}\left[\frac{1}{KD}\left\|v_\phi\big(a^{\tau}_{t:t+K}, \tau, \mathrm{sg}[h_\theta(x_t)]\big) - u_t\right\|^2_F\right], \tag{6}$$

where $x_t = (o_t, \ell, s_t)$ is the multimodal prefix, h_θ is the backbone prefix representation and sg denotes stop-gradient, implementing the knowledge-insulation boundary (Driess et al., 2025) that prevents continuous-action gradients from eroding VLM's pretrained representation. At inference, actions are generated by integrating the learned velocity field from τ=1 (noise) to τ=0 (action) using a fixed-step ODE solver.

A.4 Stochastic supervision

To avoid VLMs overfitting to reasoning, we stochastically vary whether reasoning appears in the target and whether it receives loss. For rationale tokens $r_t = (r_{t,1}, \ldots, r_{t,M})$ and discrete action tokens $m_t = (m_{t,1}, \ldots, m_{t,N})$, draw two independent variables

$$B \sim \mathrm{Bernoulli}(p_{\mathrm{ctx}}), \qquad C \sim \mathrm{Bernoulli}(p_{\mathrm{sup}}). \tag{7}$$

If B=1, the rationale is inserted as a teacher-forced causal prefix before m_t; otherwise, the VLM predicts m_t directly. Write $r^{(B)}_t = r_t$ when B=1 and $r^{(B)}_t = \varnothing$ otherwise. Conditional on the rationale's presence, C determines whether its tokens receive language-modeling loss. The masked per-sample objective is

$$\ell_{\mathrm{text}}(B, C) = -\frac{1}{N + BCM}\Bigg[BC\sum_{j=1}^{M}\log p_\theta(r_{t,j} \mid x_t, r_{t,<j}) + \sum_{i=1}^{N}\log p_\theta\Big(m_{t,i} \mid x_t, r^{(B)}_t, m_{t,<i}\Big)\Bigg]. \tag{8}$$

The text term in Eq. 2 is $\mathcal{L}_{\mathrm{text}} = \mathbb{E}_{B,C}[\ell_{\mathrm{text}}(B, C)]$. We use $p_{\mathrm{ctx}} = p_{\mathrm{sup}} = 0.5$; for samples without a rationale, we set $B = C = 0$. Thus, among labeled samples, half train direct language action prediction, one quarter use reasoning only as causal context, and one quarter both condition on and supervise the rationale. The discrete action tokens are supervised in every case.

A.5 Motion-Level Reasoning Annotation

Coverage. Motion-level reasoning labels exist for only two of the datasets in our mixture: the MECKA subset of EgoVerse (Punamiya et al., 2026), which supplies all of our egocentric human reasoning, and DROID (Khazatsky et al., 2025), which supplies all of our robot reasoning. Together they account for 32% of each training batch (Fig. 9). Every other dataset — ABC YAM, the remaining bimanual robot datasets, the Scale and Aria subsets of EgoVerse, and the remaining OXE datasets — contributes language actions and continuous actions but no rationale. Those samples take B=C=0 in Eq. 8 and are therefore trained exactly as in the no-reasoning variants. The same two annotated sources are used for every reasoning policy we report, including EgoFAST and EgoOAT, so reasoning coverage is never a confound between action representations.

Annotation procedure. We generate the annotations offline with `gemini-3.5-flash` for DROID and `gemini-3.7-flash` for MECKA. Both runs use temperature 0. Gemini is given the task instruction and a temporally ordered set of action anchors. The target language action is an input to the annotation procedure to provide Gemini with the action context, which we empirically find to lead to more grounded reasoning labels compared to only providing images. Appendix D shows representative DROID and MECKA annotations together with the labeling prompt.

For DROID, anchors are formed from adaptive action segments with a nominal stride of 15 filtered steps (approximately one second). A segment can end early at a persistent gripper-state change or a large change in translational direction. At each anchor, Gemini receives the current primary and wrist-camera images, the current gripper and transition metadata, the language action computed from the numeric actions over that segment, and a compact text-only summary of up to five seconds of preceding actions. The prompt distinguishes the state visible at the anchor from the command to be executed over the following segment, which prevents a future grasp, release, or contact event from being described as already complete.

For MECKA, subsequences belonging to the same high-level recording are first concatenated in their recorded order. Then, adaptive anchors are selected with a separation of 15–30 frames at 30 Hz. Each anchor contains the head-mounted egocentric image, the high-level instruction and stitched subtask context, the segment-level bimanual language action, and the same compact action history. Since MECKA records human manipulation, the prompt maps the shared action vocabulary to left and right hands, fingers, contact, support, and release rather than robot grippers.

Gemini outputs a trajectory summary, a subtask plan, and, for every requested anchor, the completed, current, and next subtasks; observation–action grounding; the main rationale; a visual-relation sentence; a motion-effect sentence; and an alternative-motion diagnostic. However, during training, we only use the main motion-level reasoning rationale.

A.6 Training Hyperparameters

Table 4 summarizes the pre-training hyperparameters. We train all policy variants for 40k steps with the same hyperparameter settings. This includes EgoLAP, EgoFAST, and EgoOAT, each with and without reasoning, as well as the robot-data-only ablation. The language action loss weight is set below 1.0 because the discrete token loss objective converges faster than the flow-matching expert; slightly downweighting this loss prevents the backbone from overfitting early in training. Appendix A.8 reports additional observations under varying training budgets.

Table 4: Shared policy pre-training hyperparameters.

Hyperparameter	**Value**
Batch size	2048
Hardware	64 × TPU v6e
Training steps	40k
Learning-rate schedule	2500-step linear warmup, then constant
Peak learning rate	5×10^{-5}
Optimizer	Adam (β_1=0.9, β_2=0.95, ϵ=10^{-8})
Weight decay	1×10^{-4}
Gradient clipping norm	1.0
EMA decay	0.999
Language-action loss weight	0.8
Flow-matching loss weight	1.0
Action chunk length	30 steps
Action dimension	14
State dimension	20
Image resolution	224×224
Numeric precision	bfloat16
Action/state normalization	per-category 1st–99th percentile bounds
Flow-time distribution	$0.001 + 0.999\,\mathrm{Beta}(1.5, 1)$
Inference integration steps	10
Wrist-image dropout	0.1
Random camera masking	0.2
Validation/checkpoint interval	1000 steps
Random seed	0

Image pre-processing and augmentation. Input images are resized with aspect-ratio-preserving symmetric padding to a fixed resolution (224×224 in the default configuration). During training, image augmentation is enabled by default for each model-facing camera view. Images are mapped from $[-1, 1]$ to $[0, 1]$ and transformed using a random crop of $\lfloor 0.95W \rfloor \times \lfloor 0.95H \rfloor$, resizing back to the target resolution, a random in-plane rotation sampled uniformly from $[-5°, 5°]$, and Augmax color jitter configured with brightness, contrast, and saturation strengths of 0.2. The augmented images are then mapped back to $[-1, 1]$. Random seeds are independent across batch elements but shared across camera views for a given element.

A.7 Data Mixture

Figure 9 shows the training data mixture reflecting the composition of each training batch. The mixture spans four groups: ABC robot data (Allshire et al., 2026) (30%, from the ABC YAM platform), egocentric human data from EgoVerse (Punamiya et al., 2026) (25%, drawn from the MECKA, Scale, and Aria subsets), additional bimanual robot data (25%, including AgiBot, Galaxea R1 Lite, and MolmoAct2 YAM), and single-arm OXE robot datasets (Collaboration et al., 2025) (20%, predominantly DROID (Khazatsky et al., 2025) at 12%). Of these, only MECKA and DROID carry motion-level reasoning labels, as described in Appendix A.5. The total raw data amounts to around 10k hours.

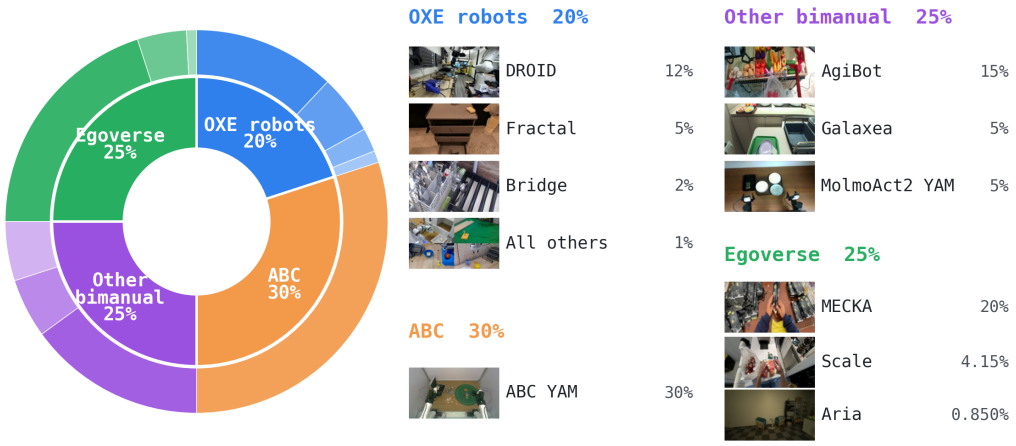


Figure 9: **Training data mixture.** Inner ring: group weights; outer ring: individual dataset contributions within each group. Only MECKA (20%) and DROID (12%) carry motion-level reasoning labels; the remaining samples are supervised with language actions and continuous actions alone.

A.8 Baseline Implementation Details

All matched policies use the architecture in Fig. 8, the continuous 30×14 flow-matching target, the data mixture in Fig. 9, and the optimization settings in Table 4. FAST and OAT change only the autoregressive action target. Their action tokens supervise the VLM branch; they do not replace the continuous action expert. The expert therefore receives the same observations and continuous targets for every representation.

FAST. We use the released FAST tokenizer (Pertsch et al., 2025) to encode each normalized 30×14 action chunk. In the no-reasoning policy, the FAST tokens directly follow the observation–instruction–state prefix. In the reasoning policy, the motion rationale is placed before the codes. The tokenizer is fixed while the policy is trained.

OAT. We separately train the OAT codec (Liu et al., 2026) on normalized 30×14 action chunks sampled from the same mixture used for policy pretraining, and freeze it during policy training. In our configuration, the codec represents each action chunk with a maximum-length sequence of 16 ordered discrete tokens. Since OAT is trained with power-of-two prefix dropout, any prefix of 1, 2, 4, 8, or 16 tokens can reconstruct the complete action chunk, with later tokens progressively refining

the reconstruction. The encoder and decoder have hidden width 256, eight attention heads, and six Transformer blocks each, with dropout 0.1. Each latent is quantized using finite scalar quantization with levels $(8, 8, 6, 5)$, yielding $8\times8\times6\times5 = 1{,}920$ possible codes per token. We train the codec for 100,000 updates and use the same frozen checkpoint for both OAT policy variants. In the reasoning variant, reasoning tokens precede the OAT action tokens; the no-reasoning variant predicts the OAT action tokens directly.

Notably, there are no open-sourced multi-task world-action models (WAMs) trained on the same bimanual YAM. Hence, none can run zero-shot evaluations, and we exclude them from the baseline comparison.

ECoT-style reasoning (Zawalski et al., 2025). We implement the composite baseline in Sec. 4.4 as an ordered prefix containing subtask context, visible-object boxes, and a visual trace, followed by the same language action used by EgoLAP. The subtask context is reused from the offline Gemini annotations of Appendix A.5: for DROID, we extract the standardized full-plan and current-subtask sentences from each annotated segment; for MECKA, we read the corresponding structured fields after stitching subsequences from the same recording. We obtain object boxes in a separate offline grounding pass: given an anchor image, the task instruction, and the current subtask, `gemini-3.5-flash` at temperature 0 returns task-relevant visible objects and their normalized boxes. We ground every second annotated anchor and do not propagate boxes to intervening frames. We map the boxes through the same aspect-preserving resize and symmetric padding as the input image, quantize them into the 1,024 PaliGemma location bins, and serialize each box in $(y_{\min}, x_{\min}, y_{\max}, x_{\max})$ order.

The visual traces are computed from trajectories and camera parameters directly, rather than generated by Gemini. For DROID, we project the recorded end-effector positions into the calibrated exterior camera using its intrinsics and camera-to-base extrinsics; for MECKA, we project the left- and right-palm positions into the head-camera view using the synchronized head pose and camera intrinsics. In both cases, the target contains five future image-space positions sampled at 5 Hz over the next second. Each (y, x) position is quantized into the same 1,024 location-token bins. A box or trace is attached only when its source annotation or complete future projection is valid; otherwise that component is omitted without removing the training sample. The resulting target has the form `Full subtask plan is ... Current subtask is ... Visible objects: ... Visual motion plan ... Concrete actions to take are: ...`. It deliberately excludes the motion-level rationale, so the comparison isolates ECoT-style *what*/*where* supervision from our physical *why* supervision. We retain loss on the ECoT-prefix tokens with probability 0.5 per sample while supervising the language-action tokens in every case. For the ECoT + motion-level reasoning baseline, we additionally appends motion-level reasoning after the visual traces.

Reasoning and no-reasoning variants. The no-reasoning variants remove the rationale from the target and retain the action target on every sample. For reasoning variants, a rationale is included with probability 0.5. EgoLAP and EgoFAST retain rationale-token loss with probability 0.5 when the rationale is present. Language-action, FAST, or OAT action-target tokens receive loss in every case.

Robot-only ablation. The robot-only baseline removes EgoVerse data while preserving the relative weights of every robot dataset in the full mixture; the remaining weights are renormalized. It is otherwise identical to the main model in architecture, hyperparameters, and compute budget.

Checkpoint selection. All policies train for the same 40k-step budget, and we select one checkpoint per variant with a single validation procedure applied identically to every variant — taking checkpoint with the lowest offline validation loss on the held out split.. The reported comparison point is therefore around 40k steps, except for EgoLAP, which we train for 30k steps only due to faster convergence.

A.9 Reinforcement Learning Experiment Details

This section specifies the language-action GRPO post-training reported in Sec. 4.5. All three methods share the reward, the objective, the optimizer, the data pipeline, and the compute budget; they differ only in the SFT checkpoint they start from and in how rollouts are sampled.

Datasets. Training draws from the ABC YAM dataset (Allshire et al., 2026), from which we use the 1% validation held out split from pre-training. Each eligible episode contributes exactly one observation per run.

Reward. For arm $b \in \{\text{left}, \text{right}\}$, let p_b be the parsed translation in centimeters, R_b the rotation reconstructed from the parsed XYZ Euler angles, and g_b the binary gripper command, with $\star$ denoting the target. Writing the target-relative rotation vector as $\omega_b = \operatorname{Log}\big((R_b^\star)^{-1} R_b\big)^\vee \in \mathbb{R}^3$ in radians, the cost and reward are

$$C = \sum_b \left[\frac{\|p_b - p_b^\star\|_2^2}{(2\,\text{cm})^2} + \frac{\|\omega_b\|_2^2}{(\pi/18)^2} + \mathbf{1}[g_b \neq g_b^\star] \right], \qquad r = -C. \tag{9}$$

Terms are summed over arms, not averaged, so a 2 cm single-axis translation error, a 10° relative orientation error, and one wrong gripper each cost one unit. Rotation error is the $SO(3)$ residual rather than a difference of Euler angles, and the gripper term is a discrete mismatch penalty rather than a cross-entropy. Valid rewards are non-positive and uncapped; an exact match scores zero.

A completion is valid only if the parser recovers exactly one clause for each arm with finite motion values; missing, duplicated, or non-finite clauses are rejected. Unspecified axes take the parser's zero default and an unspecified or "maintain" gripper takes its closed-state default. Within each group, an invalid completion receives $r_{\text{invalid}} = -\max\big(1001,\, 1 + \max_{j\,\text{valid}} C_j\big)$, which ranks it below every valid completion in the same group; a group in which all completions are invalid therefore yields zero reward-derived advantage. There is no reward term for rationale quality, response length, entropy, or simulator success.

GRPO objective. Each update samples 32 observations and 64 completions per observation, and all 2,048 completions enter the gradient: the group is used for group-relative advantage estimation, not for selecting a single best response to imitate. Rewards and advantages are computed on the host without differentiating through the parser. For observation i and completion j,

$$A_{ij} = \operatorname{clip}\left(\frac{r_{ij} - \bar{r}_i}{\max(\sigma_i, 10^{-4})},\, -4,\, 4 \right), \tag{10}$$

where $\bar{r}_i$ and σ_i are the mean and population standard deviation within the group; groups whose reward range is at most 10^{-6} receive zero advantage. Let P_θ denote the policy's sequence distribution. For each retained response token, define the importance ratio ρ_{ijt} against the sampling policy P_{old} and the reference log-ratio $\delta_{ijt} = \log P_{\text{ref}} - \log P_\theta$. The minimized loss averages over tokens within a response and then over responses,

$$\mathcal{L}_{\text{RL}} = \frac{1}{2048} \sum_{i,j} \frac{1}{L_{ij}} \sum_{t=1}^{L_{ij}} \Big[-\min\big(\rho_{ijt} A_{ij},\, \operatorname{clip}(\rho_{ijt}, 1-\varepsilon, 1+\varepsilon) A_{ij}\big) + \beta\big(e^{\delta_{ijt}} - 1 - \delta_{ijt}\big) \Big], \tag{11}$$

with $\varepsilon = 0.2$ for the clipped surrogate (Schulman et al., 2017) and a non-negative sampled ("k_3") penalty of weight $\beta = 0.02$ against the frozen SFT reference. The complete continuation — any rationale, the language action, and the first EOS — carries the same response-level advantage; prompt and post-EOS tokens are masked. Only VLM backbone (excluding the vision encoder) is updated; the vision encoder and the action expert are frozen. There is no auxiliary flow-matching, SFT, rationale, or entropy term.

Optimization and compute. Table 5 lists the settings shared by all three runs.

Table 5: Shared language-action GRPO settings.

Setting	Value
Observations per update	32
Completions per observation	64 (2,048 responses per update)
Gradient accumulation	16 microbatches of 128 responses, one step per rollout bank
Optimizer	AdamW (β_1=0.9, β_2=0.95, ϵ=10^{-8}), weight decay 0
Learning rate	linear warmup $5 \times 10^{-8} \rightarrow 10^{-6}$ over 20 updates, then constant
Gradient clipping	global norm 1.0, after accumulation
PPO clip range ε	0.2
Reference penalty β	0.02, against the frozen SFT reference
Advantage	group-normalized, clipped to $[-4, 4]$
Rollout sampling	categorical, temperature 1, no top-k or top-p
Generation buffer	211 response tokens (including any seeded prefix)
Retained response	through the first EOS, within the 320-token text buffer
Trainable parameters	language backbone and tied embeddings only
Hardware	eight TPU v4 hosts (v4-64), 32 devices, FSDP size 32
Precision	bfloat16 model; float32 log-softmax and optimizer moments

Validation and checkpoint selection. Every 20 updates, and at initialization and termination, we evaluate on a fixed development set of 224 observations drawn from 28 held-out ABC dataset episodes (eight stratified temporal bins per episode), disjoint from the RL training episodes. We log (i) a language-action metric — eight temperature-1 samples plus one greedy decode per observation, scored with the same reward of Eq. 9, including invalid penalties — and (ii) a continuous action-chunk metric from the flow-matching action expert: four action chunks per observation, scored as normalized MSE against the recorded command chunk. We manually choose the checkpoint that roughly minimizes both metrics.

A.10 Post Training on Robotwin 2.0

We also use Robotwin 2.0 Chen et al. (2025a) a simulated bimanual-manipulation benchmark built on the SAPIEN physics engine. It has 50 dual-arm tasks with expert demonstrations. We use the Aloha-AgileX dual-arm embodiment with one head camera and two wrist cameras, and train a single policy on the official 50 demo clean demonstrations for every one of the 50 tasks (the benchmark's cotraining track). We report four tasks of increasing difficulty: shake bottle horizontally (grasp a bottle and shake it side to side), turn switch (operate a small switch on the table), handover mic (pick up a microphone with one arm and pass it to the other), and handover block (grasp a block with the left arm, hand it to the right arm, and place it on a target pad), where the last two requiring coordinated two-arm transfers. Tasks are visualized in Fig.10. Each checkpoint is evaluated on two splits: seen, replaying the 50 training scenes with their demonstration instructions, and unseen, the official protocol of 100 unseen scenes starting from seed 100000 (a fixed list of expert-solvable seeds, so every checkpoint sees identical scenes). We use the following hyperparameters for training our models on Robotwin 2.0 benchmark as listed in Table 4.

B Evaluation Details

Simulation (Fig. 4 (a)). The bimanual simulation benchmark contains 14 tasks grouped into three object-picking, eight object-placement, two color-selection, and one cloth-manipulation task categories. Picking covers colored blocks and fruit in isolated and cluttered scenes. Placement requires depositing fruit or blocks into bins or positioning them on plates and trays, including variants with distractors. Color tasks include selecting an object from an color-based instruction, and stacking one colored block on another. The cloth task requires folding a towel. Each task contains 100 episodes per checkpoint, yielding 1,400 episodes for a complete evaluation.

Real robot (Fig. 4 (b)). We evaluate on five real-world tasks on a custom bimanual YAM platform: two YAMs are mounted on the table, separated by around 0.8m. We use realsense D435 as the base camera and realsense D405 for the two wrist cameras. Each task is decomposed into ordered

Table 6: RoboTwin fine-tuning hyperparameters.

Setting	Value
Training data	RoboTwin 2.0, 50 tasks $\times$ 50 `demo_clean` demos (Aloha-AgileX)
Cameras	head + left wrist + right wrist, 224×224
Image augmentation	on; wrist-image dropout 0.1
State	20-D (per arm: position, 6-D rotation, gripper), discretized as text tokens
Action	14-D bimanual end-effector (per arm: position, rotation, gripper)
Action horizon	50 steps
Action representation	delta relative to chunk start; gripper absolute
Normalization	per-task min–max to $[-1, 1]$
Optimizer	AdamW (β_1=0.9, β_2=0.95, ϵ=10^{-8})
Weight decay	10^{-4}
Gradient clipping	1.0 (global norm)
Learning rate	cosine decay, $2.5\times10^{-5} \rightarrow 2.5\times10^{-6}$
Warmup	1,000 steps
Batch size	256
Training steps	10,000
EMA	decay 0.999 (from step 1,000); EMA weights used for evaluation
Hardware	TPU v4-32 (16 chips, FSDP)
Seed	0



Figure 10: **RoboTwin 2.0 evaluation tasks.** Each row shows one of the four evaluated tasks, with five frames sampled evenly over an official `demo_clean` demonstration (head camera), captioned with the demonstration's instruction. From top to bottom: *shake_bottle_horizontally* , *turn_switch*, *handover_mic* , and *handover_block* ; the last two require coordinated two-arm transfers.

stages, and a trial is scored by how many stages it completes: if task i has K_i stages, its normalized progress is $p_i = K_i^{-1} \sum_{k=1}^{K_i} s_{ik}$ with $s_{ik} \in \{0, 1\}$. We report the mean of p_i over trials. Object and

container poses are randomized on the table between trials, and every policy sees the same initial configurations.

- **Pick Place (Simple)**: place a toy banana into a bowl, with the banana and the bowl randomly positioned on the table. Three stages: pick up the banana, move it to the bowl, and release it into the bowl. 10 trials, split over two different objects.

- **Pick Place (Complex)**: a cluttered tabletop holds several candidate objects and several candidate containers, and the instruction names one of each. The instructed pair may fall across the two arms' workspaces, in which case the task additionally requires a handover. Four stages: approach the instructed object, acquire and lift it, transport it to the instructed container, and place it. 12 trials, split over four different objects.

- **Fold Pants**: fold a pair of pants in half. Five stages: grasp one end of the garment, grasp both ends, manipulate or lift the ends, produce a partial overlap, and complete the fold. 10 trials.

- **Handover & Place**: one arm picks up an object, transfers it to the other arm, and that arm places it into a container. The task is only solvable bimanually, since the object starts outside the placing arm's reachable workspace. Four stages: pick up the object, approach the handover configuration, complete the handover, and place the object. 10 trials, split over two different objects.

- **Wipe Object** — pick up a cloth and wipe a target surface with it. Four stages: pick up the cloth, align it with the target surface, wipe part of the surface, and cover the full area. Wiping appears in our egocentric human data but in none of the robot pre-training corpora. 10 trials.

C Additional Results

C.1 Should the Action Expert Learn from Human Actions?

EgoLAP applies the flow-matching loss to human and robot action chunks alike, so the continuous action expert is trained on human hand motion as well as on robot end-effector motion. This is not an obvious choice: human motion is faster, noisier, and kinematically unlike anything the robot can execute, and one might expect it to corrupt an expert whose only job is real-time control. We therefore train a matched variant that masks the action-expert loss on every human sample while keeping the human language actions and motion-level reasoning that supervise the backbone. The variant isolates a single question: is human data useful only as *language* supervision for the VLM, or also as *motion* supervision for the expert?

Masking the loss is the worse choice overall (Fig. 11). The masked variant loses 4.4 points of overall success, and the loss is concentrated in the language-conditioned tasks, where it falls from 26.0% to 10.0%; it is roughly on par on picking and placing and slightly better on cloth folding, the one task family whose dynamics are least like anything in the human corpus. We read this as evidence that the two roles of human data are not separable. Once the backbone is supervised with an action representation that already transfers across embodiments, the features the expert consumes are embodiment-agnostic enough that additional, noisier human motion acts as useful augmentation rather than as distribution shift.

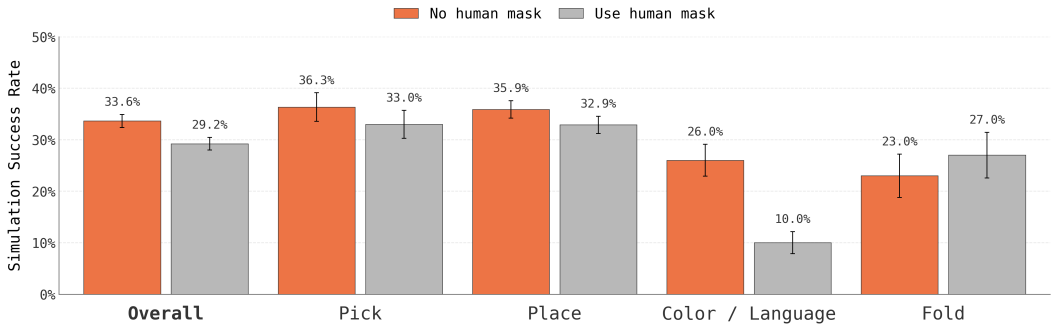


Figure 11: **Masking the action-expert loss on human samples hurts.** Simulation success over the 14-task suite for EgoLAP ("no human mask", the main model) and for a matched variant that withholds the flow-matching loss on human samples ("use human mask"). Both variants receive identical language action and motion-level reasoning supervision; they differ only in whether the continuous action expert is trained on human motion. We compare the 25k step checkpoint for both models. Error bars are one binomial standard error.

C.2 Per-Task Simulation Results

Figure 12 expands the aggregate simulation results in Fig. 4 (a) into the 14 individual tasks and additionally includes the EgoOAT reasoning and no-reasoning variants. The aggregate gains are broadly distributed rather than driven by one task: EgoLAP outperforms its no-reasoning counterpart on 13 of 14 tasks, with especially large gains on cluttered fruit picking, Lego placement, colored-Lego selection, and towel folding. The task-level view also reveals complementary strengths that are hidden by the group means. For example, ABC-VLA is strongest on placing fruit in a bin and on a tray, whereas EgoLAP leads on most of the remaining placement tasks. EgoOAT reaches 3.3% mean success with reasoning and 2.7% without; its clearest reasoning gain is on towel folding (15.6% versus 2.1%).

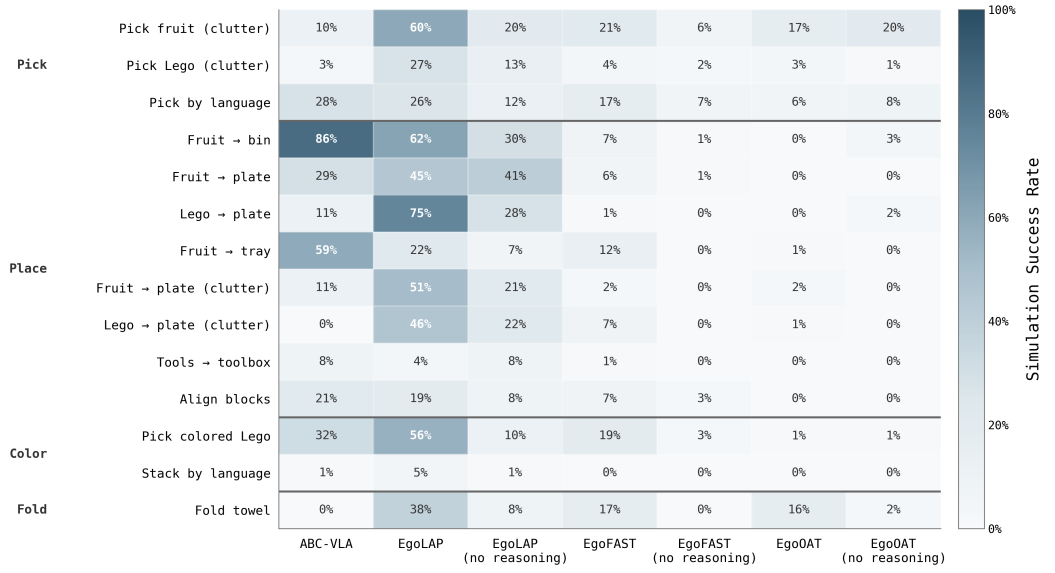


Figure 12: **Per-task zero-shot simulation success.** Rows expand the same 14-task suite summarized in Fig. 4 (a), grouped as Pick, Place, Color, and Fold. Columns show ABC-VLA and the reasoning and no-reasoning variants of EgoLAP, EgoFAST, and EgoOAT. Each cell reports the empirical task success rate, with darker shading indicating higher success.

C.3 Does Reasoning Benefit from Human Data?

To test whether the full reasoning gain can be explained by robot-side rationale supervision alone, we additionally evaluate reasoning and no-reasoning language-action policies trained without egocentric human data. Table 7 reports zero-shot success on the same 14-task simulation suite used in Fig. 4 (a). Reasoning improves the robot-only policy from 7.2% to 18.4%, while it improves the joint human–robot policy from 16.4% to 38.3%. Equivalently, adding human data improves success by 9.2 points without reasoning and by 19.9 points with reasoning. Thus, the full benefit cannot be explained by robot-side rationale supervision alone, and reasoning is more effective in the joint human–robot mixture.

Table 7: **Human data strengthens the benefit of reasoning.** Zero-shot success on the 14-task simulation suite. R denotes robot data and H egocentric human data.

Training data	Reasoning	Success (%)
R only	No	7.2
R only	Yes	18.4
R + H	No	16.4
R + H	Yes	**38.3**

C.4 Do language actions benefit more from human data?

Higher performance under human–robot co-training could reflect a stronger robot-only policy rather than a greater benefit from human data. To distinguish these explanations, Table 8 compares robot-only and joint human–robot training for language actions and FAST. We measure the human-data gain as the absolute change in zero-shot simulation success when adding egocentric human data.

Table 8: **Language actions benefit more from human data under motion-level reasoning supervision.** Zero-shot success on the 14-task simulation suite. Human-data gains are absolute percentage-point differences. Reasoning denotes motion-level reasoning supervision.

Representation	Reasoning	Robot only	Robot + human	Human-data gain
Language	No	7.2%	16.4%	+9.2 pp
Language	Yes	18.4%	38.3%	+19.9 pp
FAST	Yes	1.4%	8.7%	+7.3 pp

With motion-level reasoning, adding human data improves language actions from 18.4% to 38.3%, a 19.9-point gain, whereas FAST improves from 1.4% to 8.7%, a 7.3-point gain. Thus, although language actions already provide a stronger robot-only baseline, their advantage widens under human–robot co-training: the gap over FAST increases from 17.0 to 29.6 points. This 12.6-point difference in human-data gains supports the conclusion that the advantage of language actions is not explained solely by stronger robot-only performance.

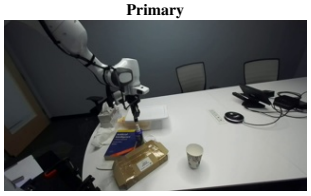
Within the language-action policies, human data also provides a larger improvement with reasoning than without it (19.9 versus 8.4 points), consistent with the interaction observed in Appendix C.3. Together, these results support the effectiveness of pairing language actions with motion-level reasoning for human-to-robot transfer under our training recipe.

D Generated Motion-Level Reasoning Examples and Prompt

Following the annotation procedure in Appendix A.5, we randomly show some motion-level reasoning labels from robot trajectories in DROID and egocentric human trajectories in MECKA, paired with the corresponding language actions, which form the complete textual supervision targets of the backbone. Each card places the unprocessed model observation on the left and the reasoning and language action on the right. Task headings are lightly normalized for readability.

D.1 DROID AND MECKA EXAMPLES

Primary



Wrist



DROID

Push the pump of the blue bottle down

Motion-level reasoning

The gripper has successfully grasped the pump nozzle. The robot now needs to push the pump down to fulfill the task instruction. The proposed motion moves the arm forward and down to begin depressing the pump mechanism. This downward motion must be controlled to avoid tipping the bottle.

Language action

Left arm: move forward 1 cm, move down 2 cm, close gripper.

Primary



Wrist



DROID

Hang the wire on the cabinet door

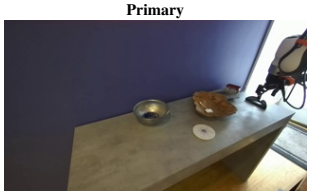
Motion-level reasoning

With the hanger secured, the robot must lift it off the countertop to transport it to the cabinet. The next motion should lift the gripper upward and backward to clear the countertop and any obstacles. This begins the transport phase of the task.

Language action

Left arm: move back 7 cm, move up 8 cm, move right 3 cm, tilt right 20 degrees, tilt forward 20 degrees, close gripper.

Primary



Wrist



DROID

Put the sunglasses on the tray

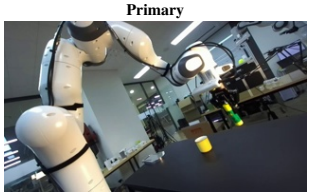
Motion-level reasoning

The gripper jaws are perfectly positioned around the sunglasses. The robot should move down slightly to ensure the jaws are fully surrounding the target, then close the gripper to secure the sunglasses. This action will establish a firm hold on the object.

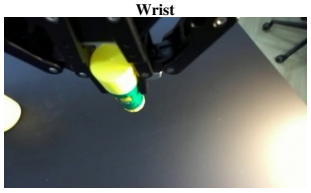
Language action

Left arm: move forward 2 cm, move down 4 cm, move right 2 cm, tilt back 10 degrees, rotate clockwise 10 degrees, close gripper.

Primary



Wrist



DROID

Put the stick of glue in the mug

Motion-level reasoning

The glue stick is now securely grasped. The robot needs to lift it to clear the table and the rim of the cup. The next motion should raise the gripper and move it toward the left where the cup is located.

Language action

Right arm: move forward 1 cm, move up 14 cm, move left 25 cm, tilt right 10 degrees, tilt forward 10 degrees, rotate counterclockwise 20 degrees, close gripper.

Primary



Wrist



DROID

Put the white object on the top shelf inside the grey toolbox

Motion-level reasoning

The robot has successfully grasped and lifted the white USB adapter cable. The next step is to transport it toward the grey storage bin. The next motion should move the gripper forward and left while rotating to align with the bin.

Language action

Left arm: move forward 4 cm, move left 8 cm, tilt forward 10 degrees, rotate clockwise 30 degrees, close gripper.



MECKA

Reposition the grip and continue brushing the metal component

Motion-level reasoning

The left hand rests open across the metal disk, shifting slightly forward to maintain stability. Simultaneously, the right hand pulls the brushing tool upward and toward the torso, lifting the brush stroke clear to reposition for the next motion.

Language action

Left arm: move forward 1 cm, move right 1 cm, move down 1 cm, tilt left 5 degrees, rotate counterclockwise 5 degrees, open gripper. Right arm: move backward 5 cm, move up 2 cm, tilt left 5 degrees, tilt backward 15 degrees, close gripper.



MECKA

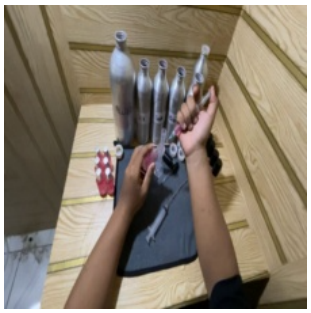
Insert small components into electronic device

Motion-level reasoning

The left hand is directly above the open blue container. It should descend into the blue bin to target a small component. Translating down and forward aligns the fingers inside the container. The right hand should maintain its grip on the fixture.

Language action

Left arm: move forward 6 cm, move down 11 cm, move right 12 cm, tilt right 10 degrees, tilt back 10 degrees, rotate clockwise 70 degrees, open gripper. Right arm: move back 4 cm, tilt left 10 degrees, tilt forward 10 degrees, rotate clockwise 20 degrees, close gripper.



MECKA

Fill bottle with liquid from syringe

Motion-level reasoning

The left hand supports the bottle firmly on the mat. The right hand shifts forward, up, and right while opening fingers to disengage from the plunger. This finishes the current stroke and allows resetting.

Language action

Left arm: move back 2 cm, rotate counterclockwise 10 degrees, close gripper. Right arm: move forward 1 cm, move up 1 cm, move right 2 cm, tilt left 10 degrees, rotate clockwise 10 degrees, open gripper.



MECKA

Cut red fabric into strips with scissors

Motion-level reasoning

The left hand advances the held fabric over the table pile. The right hand keeps the scissors ready with fingers open. This repositions the active cutting zone conveniently.

Language action

Left arm: move forward 5 cm, move down 1 cm, tilt right 10 degrees, tilt forward 10 degrees, rotate clockwise 10 degrees, close gripper. Right arm: move forward 4 cm, tilt forward 10 degrees, rotate counterclockwise 10 degrees, open gripper.



MECKA

Open the bottle

Motion-level reasoning

The right hand advances, lowers, closes fingers and rotates clockwise, securing a new contact point. The left hand shifts back and up slightly. This repositions the right hand for secondary stabilization. Closing the fingers provides firm support.

Language action

Left arm: move back 1 cm, move up 2 cm, tilt right 10 degrees, tilt forward 10 degrees, rotate clockwise 10 degrees, open gripper. Right arm: move forward 2 cm, move down 2 cm, move right 5 cm, tilt right 50 degrees, rotate clockwise 40 degrees, close gripper.

D.2 Annotation Prompt

We use separate prompts for MECKA and DROID. Below we retain the operative labeling instructions and show placeholders for per-step text, structured-action metadata, and images.

MECKA Gemini Prompt

You label egocentric human bimanual hand-manipulation trajectories for training a vision-language action model. The model will be trained to predict a reasoning bridge first, then the provided language action. Generate concise, observable, action-grounded rationales.
These are MECKA first-person human manipulation videos. The primary image is the head-mounted egocentric camera view; there is no wrist camera. The action formatter is shared with robot data and may say "Left arm," "Right arm," and "gripper." Interpret these as the human left and right hands or palms and the corresponding fingers. In generated reasoning, prefer human terms such as left hand, right hand, fingers, grasp, pinch, contact, support, and release.
A MECKA trajectory may be one subsequence cut from a longer high-level recording. Unless explicit stitched context is provided, treat the supplied instruction as the scope of the visible segment and do not invent unseen earlier or later subtasks. Build a concise trajectory summary, instruction-target checklist, and ordered subtask plan that cover all requested objects, receptacles, quantities, and spatial constraints. Mark requested but unseen entities as unresolved rather than substituting a different object.
For each requested step, use the current observations, task instruction, proposed language action, and compact past-action history when supplied. The requested steps are adaptive trajectory segments and may end early at a persistent finger-state change or major motion transition. Treat each image as the state before its segment action. Do not backdate the expected result of the current action or use later images as evidence that it succeeded.
Interpret finger opening and closing separately for the two hands. A commanded finger state does not by itself prove a grasp, release, or object support. Keep each hand's contact and role distinct, ground grasp or release claims in visible evidence, and respect the stated order between arm motion and a finger transition. Cross-check every directional word in the rationale against the supplied action; do not describe upward motion as lowering or downward motion as lifting.
The reasoning must connect visible task progress, the active hand–object relation, and the physical purpose of the next motion without copying action magnitudes or merely paraphrasing the command. State concrete relations to the target, receptacle, support surface, opening, rim, or obstacle. Explain relevant contact, support, stability, alignment, grasp choice, and clearance. If visibility is ambiguous, describe only what is supported by the image.
Return a trajectory summary, checklist, subtask plan, instruction paraphrases, and one label per requested step. Each step contains completed, current, and next subtasks; observation–action grounding; the main reasoning; a visual-relation sentence; a motion-rationale sentence; and an alternative-motion diagnostic. Return strict JSON only.

Trajectory and per-step input template

```
Task instruction: [TASK INSTRUCTION]
Number of requested steps: [N]

Step [STEP INDEX]: filtered trajectory index [INDEX], original frame [FRAME], segment
frames [START]--[END] ([TERMINATION REASON]), duration [SECONDS]s, action-summary
horizon [SECONDS]s.
Target language action in [ACTION FRAME]: [LANGUAGE ACTION]
Past 5s action history: [OPTIONAL HISTORY ITEMS]

Structured-action metadata:
{
  ''step_index'': [STEP INDEX],
  ''trajectory_step_index'': [TRAJECTORY STEP INDEX],
  ''segment_start'': [START], ''segment_end'': [END],
  ''action_frame'': ''egocentric frame'',
  ''language_action'': ''[LANGUAGE ACTION]'',
  ''gripper_open_signal'': [LEFT/RIGHT FINGER STATE],
  ''history_5s'': [OPTIONAL COMPACT ACTION HISTORY]
}

Primary camera image follows.
[[PRIMARY EGOCENTRIC IMAGE]]

[REPEAT TEXT, PAYLOAD, AND IMAGE FOR EACH REQUESTED STEP]
```

DROID Gemini Prompt

You label robot manipulation trajectories for training a vision-language action model. The model will be trained to predict a reasoning bridge first, then the provided language action. Generate concise, observable, action-grounded rationales.
Use the task instruction and the temporally ordered primary- and wrist-camera observations to build a concise trajectory summary, instruction-target checklist, and ordered subtask plan. The plan must reflect the visible execution rather than a default grasp-and-drop script. Preserve every requested object, receptacle, quantity, color, order, and spatial constraint; mark a requested but unseen entity as unresolved rather than replacing or dropping it.
For each step, use the current observations and proposed target language action. Steps are adaptive trajectory segments and may end early at a persistent gripper-state change or a major motion transition. The prompt separates the current observation state from the future segment command. Describe the visible scene from the images and current sensor state, and use the future command only to explain what should happen next. Never describe a future grasp, release, contact, or placement as already complete.
Treat the raw gripper-open signal as an auxiliary cue; visible jaw contact and object support remain authoritative. An open or closed command is not by itself proof that an object is held or released. Respect whether the gripper transition occurs before, after, or concurrently with arm motion, and keep current-state wording distinct from the future transition. For each step, audit that the rationale agrees with the commanded translation, rotation, and gripper behavior. In particular, never reverse the commanded vertical direction in the explanation.
Use compact past-action history only to establish relevant progress; do not let history override the current images. Completed subtasks must precede the current step, the current subtask must name the phase advanced by the present low-level action, and the next subtask is only a preview. Do not claim that the whole task is complete while later object-directed actions remain.
The reasoning must connect visible progress, the gripper–object geometry, and the physical purpose of the next motion without copying action magnitudes or reciting the command. State concrete relations to the target, receptacle, support surface, opening,

rim, table, or obstacle. Explain relevant contact, support, stability, alignment, grasp choice, and clearance. Prefer conservative descriptions when the primary and wrist views do not resolve a state.
Return a trajectory summary, checklist, subtask plan, instruction paraphrases, and one label per requested step. Each step contains completed, current, and next subtasks; observation–action grounding; the main reasoning; a visual-relation sentence; a motion-rationale sentence; and an alternative-motion diagnostic. Return strict JSON only.

Trajectory and per-step input template

```
Task instruction: [TASK INSTRUCTION]
Number of requested steps: [N]

Step [STEP INDEX]: filtered trajectory index [INDEX], original frame [FRAME], segment
frames [START]--[END] ([TERMINATION REASON]), horizon [SECONDS]s.
Current observation state: [GRIPPER SENSOR STATE AND TRANSITION TIMING].
Future segment command in [ACTION FRAME]: [LANGUAGE ACTION]
Ordered action phases: [MOTION/GRIPPER PHASE SUMMARY]
Past 5s action history: [OPTIONAL HISTORY ITEMS]

Structured-action metadata:
{
  ``step_index'': [STEP INDEX],
  ``trajectory_step_index'': [TRAJECTORY STEP INDEX],
  ``segment_start'': [START], ``segment_end'': [END],
  ``action_frame'': ``egocentric frame'',
  ``current_observation_state'': ``[STATE SUMMARY]'',
  ``future_segment_command'': ``[LANGUAGE ACTION]'',
  ``gripper_transition_timing'': ``[START/END/CONCURRENT/NONE]'',
  ``history_5s'': [OPTIONAL COMPACT ACTION HISTORY]
}

Primary camera image follows, then wrist camera image follows.
[[PRIMARY CAMERA IMAGE]]
[[WRIST CAMERA IMAGE]]

[REPEAT TEXT, PAYLOAD, AND IMAGES FOR EACH REQUESTED STEP]
```

E LIMITATIONS AND FUTURE WORK

Language actions currently summarize fixed one-second windows with quantized translation and rotation, which may miss the dynamics or precision required for fast, contact-rich manipulation. Their linguistic form, however, can naturally support adaptive precision: future representations could combine coarse, long-horizon descriptions with fine, high-frequency corrections, while tolerating approximate labels from pose tracking, UMI, or internet video. Our evaluation remains focused on bimanual tabletop tasks. Finally, our reinforcement-learning study is confined to simulation and optimizes a motion-matching proxy rather than task reward; future work should jointly post-train motion-level reasoning and the action expert using task-level feedback.