ACE-Ego-0: Unifying Egocentric Human and Robotic Data for VLA Pretraining

Hao Li[1,2*] **Ganlong Zhao**[1,2*,†] **Yufei Liu**[1,4*] **Haotian Hou**[1,2*] **Guoquan Ye**[1,3] **Tongyan Fang**[1,5]
Chunxiao Liu[1] **Siyuan Huang**[1†] **Jianbo Liu**[1] **Xiaogang Wang**[1,2] **Hongsheng Li**[2,1 ✉]

[1]ACE Robotics [2]CUHK MMLab [3]CUHK, Shenzhen [4]SJTU [5]THU

*Equal contribution †Project lead ✉Corresponding author

Vision-Language-Action (VLA) models benefit from large-scale and diverse embodied data, yet scaling robot trajectory collection is costly and labor-intensive. Recent advances show that large-scale egocentric human videos provide complementary real-world supervision in pretraining. However, joint training on human and robot data remains challenging due to divergences in action spaces, embodiment structures, temporal dynamics, and supervision quality. We introduce ACE-EGO-0, a unified VLA pretraining framework jointly leveraging heterogeneous data sources. To extract large-scale pretraining supervision from egocentric human videos, we build a scalable egocentric video-to-action pipeline that converts raw human videos into robot-format pseudo-action trajectories. To make these labels comparable with robot demonstrations, ACE-EGO-0 uses a unified action representation based on camera-space actions, morphology conditioning, and time-aligned action chunking. To robustly leverage noisy pseudo-action supervision from egocentric human videos, we formulate a reliability-aware training objective with a human auxiliary loss that concentrates supervision on reliable signals. We instantiate ACE-EGO-0 on 4.53K hours of robot and simulation data, together with 1.48K hours of pseudo-action-labeled egocentric human data. Experiments show that incorporating large-scale human supervision under reliability-aware weighting consistently improves both unified joint pretraining and supervised fine-tuning. ACE-EGO-0 achieves state-of-the-art performance on RoboCasa GR1 TableTop and RoboTwin 2.0, while demonstrating strong transfer to real-world bimanual manipulation.

Date: Jun 2026
Website: https://acerobotics-vla.github.io/ACE-Ego/
Code: https://github.com/ACERobotics-VLA/ACE-Ego

Keywords: Vision-Language-Action Models, Robot Manipulation, Learning from Human Video

1 Introduction

Developing general-purpose robotic systems capable of operating across diverse real-world environments remains a central objective of embodied AI. Vision-Language-Action (VLA) models [1, 2, 3, 4] offer a promising path toward this goal by jointly modeling perception, language, and action. A common premise is that broad and diverse embodied experience is critical for acquiring generalizable manipulation skills. Similar to the scaling trends observed in language and vision foundation models, the performance of VLA policies is strongly correlated with the scale and diversity of the training data available during pretraining. However, collecting robot demonstrations at scale remains costly and labor-intensive, limiting both dataset size and behavioral diversity. Large-scale egocentric human videos provide a compelling complementary source of embodied supervision, offering substantially broader coverage of real-world interactions at much lower collection cost. Integrating these heterogeneous data sources into a unified training framework remains challenging due to discrepancies in spatial representations, embodiment structures, temporal horizons, and supervision fidelity.

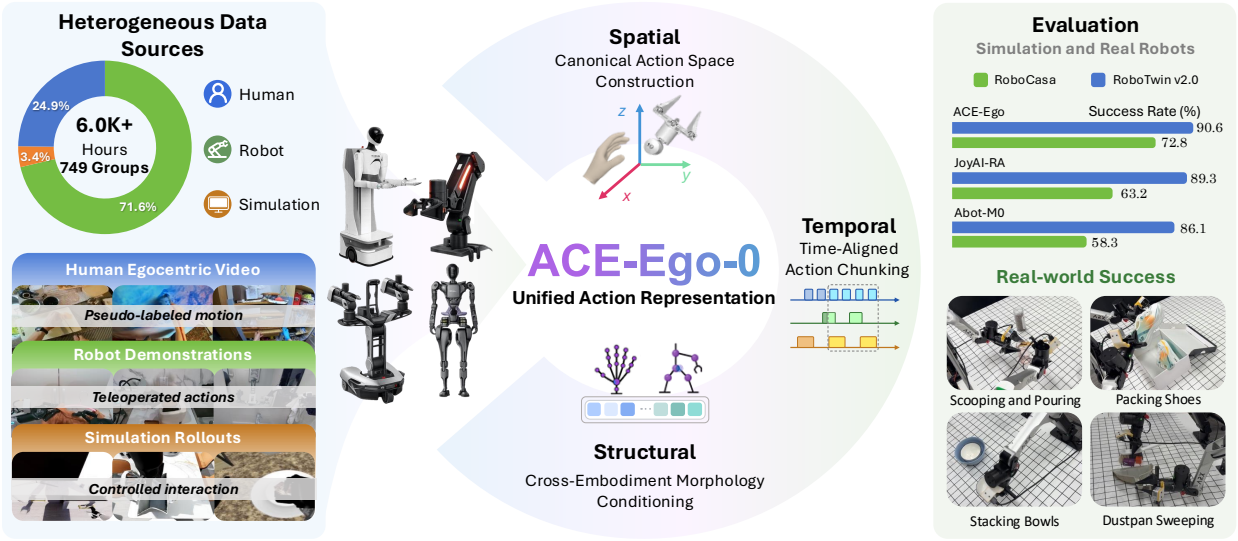


Figure 1 Overview of ACE-EGO-0. We pretrain a unified VLA policy on a 6.0K+ hour mixed embodied dataset comprising large-scale egocentric human videos, multi-embodiment robot demonstrations, and simulation rollouts. ACE-EGO-0 unifies heterogeneous human and multi-embodiment robot data into a shared representation space through spatial, structural, and temporal alignment. We achieve state-of-the-art performance on RoboCasa and RoboTwin 2.0, while demonstrating strong real-world bimanual transfer.

Existing cross-embodiment VLA methods [5, 6, 7, 8, 9] address representation heterogeneity through shared action spaces, embodiment-specific tokenizers, soft-prompted action experts, or latent action representations, enabling heterogeneous robot demonstrations to be trained within a unified policy framework. However, these approaches remain bottlenecked by the scalability of robot data collection, as they rely primarily on teleoperated demonstrations. Large-scale egocentric human videos have recently emerged as an appealing complementary source: they are far cheaper to collect and cover a much broader range of manipulation skills in everyday scenes. Several recent works [8, 10, 11, 12] leverage egocentric human videos, reconstructing hand trajectories and contact targets as action proxies for pseudo-action supervision. However, treating pseudo-actions as equivalent to sensor-logged robotics actions during training injects the label noise directly into the model. In addition, current 3D human hand reconstruction methods typically express hand poses in local space [13, 14] using MANO [15], whereas robot demonstrations are generally recorded in global world space. This misalignment prevents policy models from effectively using both human and robot data for unified policy training. Neither representation heterogeneity nor supervision-quality mismatch is fully resolved in current mixed-source VLA pretraining frameworks.

We present ACE-EGO-0, a VLA pretraining framework with unified action representation for heterogeneous embodied data, bridging spatial, structural, and temporal discrepancies. Specifically, we introduce canonical action space construction that represents both robot end-effector trajectories and reconstructed human hand pseudo-action trajectories in a common observation-centric coordinate frame, eliminating the need for the policy to learn embodiment-specific coordinate transformations beyond a standard camera extrinsic. To accommodate diverse embodiments, we incorporate cross-embodiment morphology conditioning via embedding robot kinematic descriptions and learned surrogate embeddings for human-video sources. Furthermore, we propose time-aligned action chunking, which indexes future actions according to physical timestamps rather than frame indices, ensuring temporal consistency across datasets collected at different control frequencies. As supervision quality varies substantially across data sources, representation alignment alone is insufficient to achieve effective mixed-source pretraining. We introduce a reliability-aware training objective that explicitly accounts for supervision fidelity. Sensor-logged robot trajectories supervise the primary flow-matching objective, while pseudo-actions are down-weighted and serve as auxiliary supervision primarily on noiseless position channels and modulated by dataset-level and step-level quality estimates.

We propose a scalable five-stage egocentric data processing pipeline and apply it over six diverse egocentric video datasets to obtain 1.48K hours of pseudo-action-labeled human video. Combining it with 4.53K+ hours of sensor-logged multi-embodiment robot demonstrations and simulation rollouts yields a 6.0K+ hour heterogeneous pretraining

dataset for our proposed ACE-EGO-0. We evaluate ACE-EGO-0 on RoboCasa, RoboTwin 2.0, and a real bimanual ARX platform. ACE-EGO-0 reaches 72.8% average success on RoboCasa GR1 TableTop benchmark, achieves 91.12% and 90.62% average success rates on RoboTwin 2.0 Easy/Hard splits, and demonstrates strong real-world bimanual performance on long-horizon, contact-rich tasks. Ablation studies confirm that morphology conditioning, time-aligned action chunking, and reliability-aware human supervision each contribute to the final performance, and that scaling pseudo-action-labeled human video on top of robot data yields further gains.

Our contributions are summarized as follows.

- We introduce ACE-EGO-0, a unified VLA pretraining framework addressing representation heterogeneity via a unified action representation and supervision-quality mismatch via a reliability-aware training objective.
- We develop a scalable five-stage pipeline that converts large-scale egocentric human videos into robot-compatible pseudo-action trajectories, producing 1.48K hours of pseudo-action-labeled human data and enabling joint pretraining with 4.53K hours of multi-embodiment robot and simulation data.
- We demonstrate that large-scale human supervision consistently improves both unified VLA pretraining and downstream supervised fine-tuning, achieving state-of-the-art performance on RoboCasa and RoboTwin 2.0 while exhibiting strong transfer to real-world bimanual manipulation.

2 Related Work

2.1 Scalable Vision-Language-Action Model Pretraining

Recent progress in robot learning has moved from task-specific imitation policies toward generalist vision-language-action (VLA) models trained on large and diverse robot datasets. RT-1 [1] showed that transformer policies can absorb large-scale real-robot demonstrations and generalize across language-conditioned manipulation tasks, and RT-2 [2] connected web-scale vision-language pretraining with robot action prediction. The Open X-Embodiment and RT-X effort [5] then aggregated robot trajectories across institutions, embodiments, and task families, establishing cross-embodiment training as a viable route to broader generalization. A growing family of open and large-scale VLA systems—including Octo [6], OpenVLA [16], π_0 [3], $\pi_{0.5}$ [4], RDT [17], CogACT [18], and GR00T [8]—has since scaled model capacity, data diversity, and action-generation flexibility. However, the very data scaling that fuels these foundation models also introduces a fundamental bottleneck: as a single policy ingests increasingly diverse sources, treating them as a homogeneous corpus becomes exceptionally challenging, because robot datasets differ simultaneously in coordinate frames, kinematic structure, and control frequency.

Prior works have attempted to mitigate this *representation mismatch* along individual axes. Shared end-effector action formats and discrete action tokenizers facilitate cross-dataset training [5, 16]; embodiment-aware tokenizers, adapters, or projectors handle kinematic heterogeneity prior to a shared backbone [7, 8]; and universal or latent action spaces seek to minimize embodiment-specific action discrepancies [19, 20, 21]. Spatially grounded policies, such as SpatialVLA [22], 3D-VLA [23], and TraceVLA [24], incorporate 3D geometric structures or image-space trajectories to align perception and action. Yet, these mechanisms rarely address all three dimensions of heterogeneity jointly: a shared action vector does not guarantee aligned coordinate frames; fixed-length action chunks span disparate physical durations under varying control frequencies; and kinematic structures are often implicitly absorbed via simple dataset IDs or learned codes. In contrast, ACE-EGO-0 systematically aligns heterogeneous robot sources across all three axes prior to the shared VLA training objective—employing a unified camera-space action representation, cross-embodiment morphology tokens, and time-aligned action chunking.

2.2 Learning from Egocentric Human Video

Beyond robot-collected data, egocentric human video offers a highly scalable and cost-effective source of manipulation experience, capturing rich object interactions, diverse environments, and long-tail behaviors that are difficult to acquire via robot teleoperation. Large-scale egocentric datasets, such as Ego4D [25], EPIC-KITCHENS [26], EgoExo4D [27], EgoDex [28], and EgoScale [29], have significantly amplified this potential. Earlier paradigms leveraged such videos primarily for representation or visual reward learning [30, 31, 32, 33, 34, 35, 36, 37]; while these methods extract strong visual priors, they still rely heavily on downstream robot demonstrations to map perception to motor control. More recent endeavors extract direct action-level supervision from human videos, either by learning latent or inverse-dynamics actions from action-free footage [20], or by reconstructing explicit hand, wrist, or body trajectories and mapping them

to robot-compatible commands via retargeting, inverse kinematics, visual domain translation, or morphology-agnostic formulations [12, 38, 10, 39, 11, 40, 41, 42, 43]. DIAL [44] takes a different route, incorporating egocentric human video into VLA pretraining through a latent world model that decouples high-level intent prediction from low-level action generation.

Although these advances unlock human video as a scalable supervision source, they expose a critical *supervision-quality mismatch* that is orthogonal to representation heterogeneity. Unlike high-fidelity sensor-logged robot trajectories, human action labels extracted via vision pipelines are inherently noisy pseudo-actions, prone to tracking jitter, occlusions, and estimation bias. Existing frameworks typically either bypass direct action-level training or naively feed these noisy pseudo-actions into the same behavior-cloning or diffusion objectives used for clean robot data. This equivalent treatment forces the policy to directly mimic the artifacts and failures of the reconstruction pipeline. To resolve this, ACE-EGO-0 routes human-video samples through a reliability-aware auxiliary objective. By restricting supervision to highly reliable position channels and dynamically weighting the loss based on both dataset-level and step-level quality estimates, we ensure that high-fidelity robot data anchor the primary action expert, while human videos provide safe, robust, and complementary auxiliary supervision.

3 Method

To pretrain a generalizable VLA policy on heterogeneous embodied data, we must overcome two fundamental challenges: *representation heterogeneity* and *supervision-quality mismatch*. ACE-EGO-0 introduces a two-fold framework as illustrated in Fig. 2. First, we establish a **Unified Action Representation** (Sec. 3.1) that aligns multi-embodiment data along spatial, structural, and temporal spaces. Second, to prevent estimation noise from human pseudo-actions from corrupting the shared policy in the unified action representation, we propose a **Reliability-Aware Training Objective** (Sec. 3.2) that leverages noisy human pseudo-actions as auxiliary supervision.

3.1 Unified Action Representation

Jointly training on diverse robot trajectories and human videos requires a shared action interface that removes dataset-specific coordinate and temporal conventions. We achieve this by projecting all data sources from three perspectives: *spatial* alignment via camera-space coordinates (Sec. 3.1.1), *structural* alignment via kinematic morphology conditioning (Sec. 3.1.2), and *temporal* alignment via time-aligned action chunking (Sec. 3.1.3). Together, these mechanisms map heterogeneous trajectories into a single, embodiment-agnostic action space.

3.1.1 Canonical Action Space

ACE-EGO-0 first aligns sources spatially by representing actions from both robot and human data in the head-camera coordinate frame before they enter the model. Predicting in the camera frame keeps actions and observations in a unified coordinate system, eliminating the need for the policy to learn complex, platform-specific world-to-camera transformations. Under this formulation, actions and observations are fed in a platform-agnostic framework, and the pretrained policy transfers to a new embodiment by simply swapping a single camera extrinsic at inference time.

Robot action convention. For each robot source, the bimanual end-effector poses are projected into the head-camera frame. Poses on top of a robot base or in a world frame s are transformed using the calibrated camera extrinsic:

$$p_{\text{cam}} = R_{\text{cam}\leftarrow s}\, p_s + t_{\text{cam}\leftarrow s}, \qquad R_{\text{cam},ee} = R_{\text{cam}\leftarrow s}\, R_{s,ee}, \tag{1}$$

where p_s and $R_{s,ee}$ denote the end-effector position and orientation in the source frame, respectively. Orientations are parameterized using a continuous 6D representation [45]. Combined with gripper commands and arm activity flags, this yields a unified bimanual action vector (see Appendix A.1 for the details). Expressing actions in this unified format ensures that the action expert consumes both human and robot trajectories through a shared states interface.

Human end-effector equivalents. Since human hands do not have a physical end-effector, we define a hand-centric coordinate frame as proxy end-effector, allowing human motions to be represented in a robot-compatible form while remaining directly connected to hand mesh reconstruction. We designate the wrist joint as the end-effector origin, as it

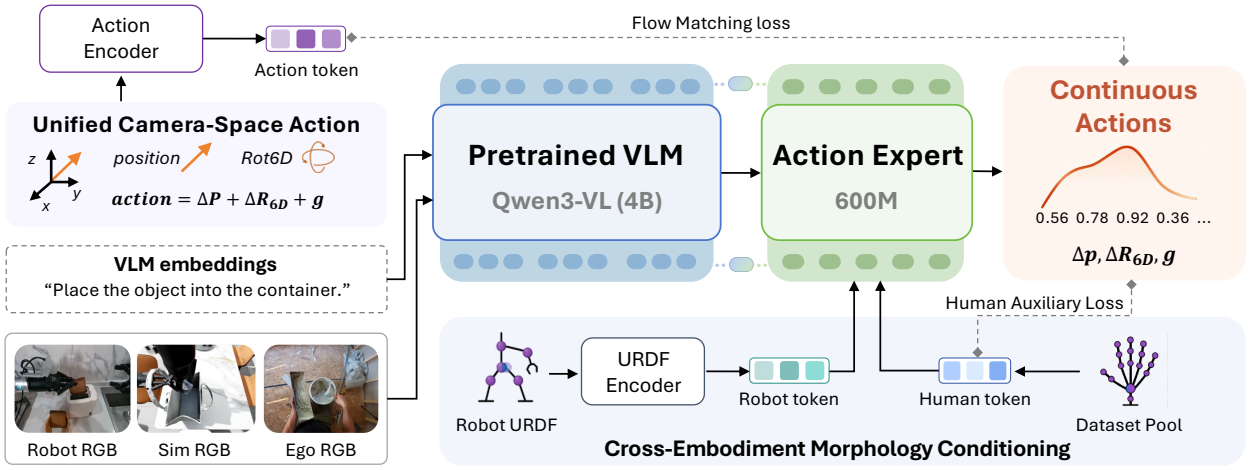


Figure 2 Architecture of ACE-EGO-0. The vision-language backbone processes multi-view images and language instructions into shared representations. The action expert receives these representations together with morphology tokens that encode each source's embodiment (robot URDF or human surrogate) to predict time-aligned camera-space action chunks via flow matching. Robot samples supervise the primary action loss; human samples contribute through an auxiliary loss with per-channel reliability weighting that concentrates supervision on the position channels.

is reconstructed most consistently in HaMeR's [13] frame-wise predictions. To mitigate yaw drift under occlusion, we construct a stable hand-centric orientation frame $R \in \mathrm{SO}(3)$ using the palm plane and wrist-to-finger vectors, which is then converted to the same continuous 6D representation used for robots. For gripper openness, we employ the normalized thumb-to-palm distance as a proxy for hand closure, linearly scaled to match the robot's physical gripper stroke. This parameterization normalizes human trajectories using the base value and maps them into the shared bimanual action space, which is used for seamless joint training across human and robot data. The exact geometric derivations of the hand frame are detailed in Appendix A.1.

3.1.2 Cross-Embodiment Morphology Conditioning

Although a canonical action space resolves spatial discrepancies, differences in kinematic chains, joint limits, and physical dimensions still persist across embodiments. To unify the cross-embodiment discrepancy, we embed humans and each robot type into a shared morphology space. A major contribution of ACE-EGO-0 is addressing this structural mismatch by conditioning the action expert on a morphology token. For robots, this token is dynamically computed from its URDF graph; for humans, it is shared across different people and updated via back-propagation. Crucially, we keep this morphology token isolated from the vision-language backbone and inject it only during action decoding, thereby keeping our VLM backbone embodiment-agnostic. Robot and human morphologies are projected into this shared token space via parallel pathways:

$$h_{\text{morph}} = \begin{cases} P_{\text{morph}}(E_{\text{urdf}}(\mathcal{G}_r)), & \text{robot source } r, \\ P_{\text{surr}}(e_d), & \text{human source } d, \end{cases} \tag{2}$$

where E_{urdf} encodes the URDF graph $\mathcal{G}_r$ at both global and local manipulation scales (Appendix A.2), and e_d is a learned surrogate embedding capturing the visual and dataset-specific priors of human source d (Appendix A.4). Both pathways condition the action expert through a unified interface.

3.1.3 Time-Aligned Action Chunking

For temporal alignment, robot datasets often have different control frequencies. If we predict a fixed number of future steps, the policy must plan for different physical durations across datasets. To prevent this temporal mismatch, ACE-EGO-0 defines action chunks by physical duration rather than step count. For a dataset d with control frequency f_d, we set the step horizon H_d based on a target physical duration $T^\star$:

$$H_d = \mathrm{round}(f_d T^\star). \tag{3}$$

This formulation ensures that all datasets supervise the same future physical window $T^\star$. However, training on variable-horizon chunks within the same batch can cause large padding overhead and training instability. We address these issues with a structured batch sampling strategy. Specifically, trajectories are pre-chunked according to the target physical window to maintain temporal consistency and minimize padding overhead. For a sample starting at index t in an episode of length L_e, we define the normalized episode phase ϕ as:

$$\phi = \mathrm{clip}\left(\frac{t + \frac{1}{2}H_d}{L_e}, 0, 1\right), \tag{4}$$

Since H_d is determined by the target physical duration and dataset control frequency, ϕ is comparable across datasets with different frame rates. We discretize ϕ into a phase bucket b_ϕ and H_d into a horizon bucket b_H. We then form mini-batches using a composite key:

$$k = (c_{\text{task}}, b_\phi, b_H), \tag{5}$$

where c_{task} is a task cluster from episode metadata. This bucketing strategy balances training stability and computational efficiency. Grouping by task ensures semantic coherence within each batch, while grouping by horizon minimizes the padding required for samples with different chunk lengths, thereby significantly reducing padding overhead and stabilizing the gradient updates.

3.2 Reliability-Aware Training Objective

Even with aligned action spaces, naive joint training on mixed-source data risks propagating estimation noise from human pseudo-actions directly into the action expert, which degrades the learning of the robust control policy from high-fidelity robot data. To resolve this supervision-quality mismatch, we propose a reliability-aware training objective. We formally define the spatiotemporal reliability for each action dimension (e.g., control channel) $j \in \{1, \ldots, D\}$ at step t as:

$$W_{t,j} = \rho_j \cdot w_{t,j}, \tag{6}$$

where $\rho_j \in [0, 1]$ is a static, channel-level prior reflecting the intrinsic tracking stability of different action dimensions. In practice, these priors ρ_j are empirically assigned based on the measurement noise of the human pose estimator (e.g., positioning channels are highly reliable and assigned $\rho = 1.0$, whereas wrist rotations and gripper states are prone to occlusion noise and assigned lower weights). The term $w_{t,j} \in [0, 1]$ represents a dynamic, step-level smoothness factor that down-weights local tracking failures or implausible kinematic jumps.

With this reliability-aware weighting strategy, high-fidelity robot data anchors the primary objective across all channels, while noisy human pseudo-actions contribute to training through a robust auxiliary loss scaled by $W_{t,j}$. The dynamic term $w_{t,j}$ further factorizes into a dataset-level prior and a local step-level smoothness weight, with exact formulations detailed in Appendix A.5.

Robot Primary Loss The primary robot loss follows the standard conditional flow-matching formulation, optimized over the valid action dimensions selected by the action mask M. Given a clean, sensor-logged robot action target $\mathbf{a}$ and Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$, the flow interpolant is defined as $\mathbf{a}_s = s\mathbf{a} + (1-s)\boldsymbol{\epsilon}$ for $s \sim \mathcal{U}(0, 1)$. Then the robot loss is formulated as:

$$\mathcal{L}_{\text{action}} = \mathbb{E}_{s,\boldsymbol{\epsilon}} \sum_{t,j} M_{t,j} \left\| \hat{v}_\theta(\mathbf{a}_s, s)_{t,j} - (\mathbf{a} - \boldsymbol{\epsilon})_{t,j} \right\|^2, \tag{7}$$

where $\hat{v}_\theta$ is the predicted velocity field and $M_{t,j} \in \{0, 1\}$ is the action mask. During training, we use the delta action chunk formulation following [3], expressed in the head-camera frame.

Human Auxiliary Loss To incorporate human demonstrations without corrupting the policy's primary control capabilities, we introduce human auxiliary loss. Let $\tilde{\mathbf{a}}$ denote the temporally smoothed human target, and $\mathbf{a}_s = s\tilde{\mathbf{a}} + (1-s)\boldsymbol{\epsilon}$ be the corresponding flow interpolant. We apply the spatiotemporal reliability weight $W_{t,j}$ within a robust Huber regression loss:

$$\mathcal{L}_{\text{haux}} = \mathbb{E}_{s,\boldsymbol{\epsilon}} \frac{1}{Z} \sum_{t,j} M_{t,j} W_{t,j} \ \mathrm{Huber}_\beta(\hat{v}_\theta(\mathbf{a}_s, s)_{t,j} - (\tilde{\mathbf{a}} - \boldsymbol{\epsilon})_{t,j}), \tag{8}$$

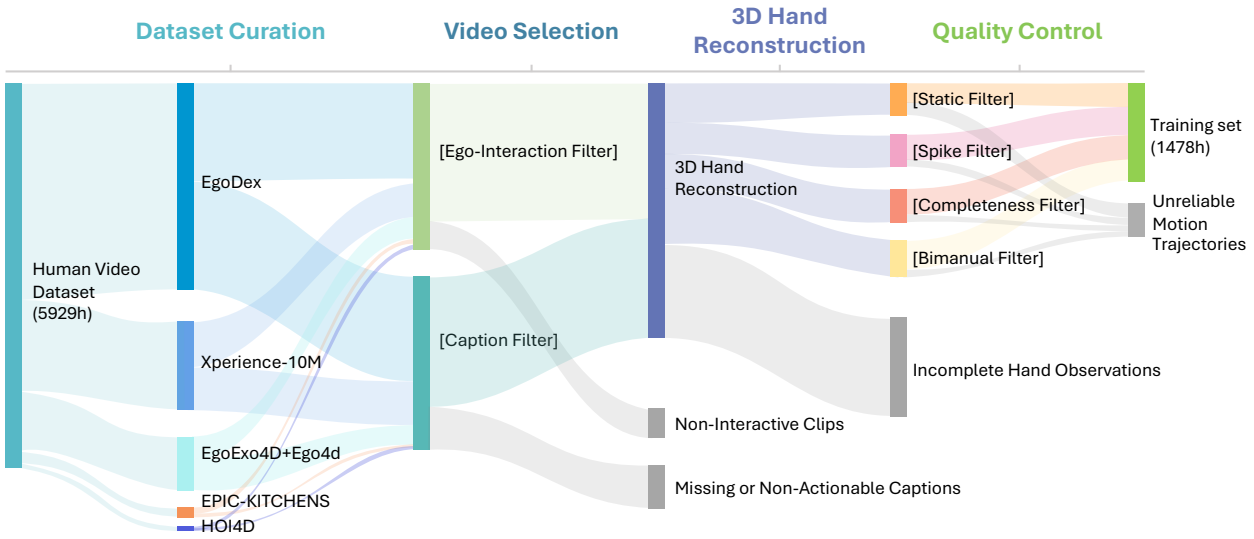


Figure 3 Overview of the ACE-EGO-0 data processing pipeline for constructing training-ready embodied manipulation data from large-scale egocentric human video. Raw videos pass through video selection, motion reconstruction, and multi-stage quality control, yielding 1,478 hours of pseudo-action-labeled embodied manipulation data that complement the robot and simulation portions of the training pool.

where $Z = \sum_{t,j} M_{t,j} W_{t,j}$ is the normalization factor. This formulation concentrates human supervision on highly reliable position channels while safely discounting noisy rotation and gripper signals (see Appendix A.5 for details on the smoothness statistics and thresholds).

The joint training objective is a weighted combination of the two losses:

$$\mathcal{L} = \mathcal{L}_{\text{action}} + \lambda_{\text{haux}} \mathcal{L}_{\text{haux}}, \tag{9}$$

where λ_{haux} balances the contribution of the human auxiliary loss (hyperparameters and sensitivity analyses are provided in Appendix A.5).

4 Heterogeneous Pretraining Data

The ACE-EGO-0 pretraining pool covers the full spectrum of embodied experience, including sensor-logged robot demonstrations across multiple platforms, simulation rollouts, and pseudo-action-labeled egocentric human videos, totaling more than 6.0K hours as shown in Table 1. These sources exhibit significant representation heterogeneity as identified in Section 1: they differ in spatial coordinate frames, kinematic structures, and control frequencies, and further vary in action-label quality. Section 4.1 catalogs our overall mixed-source datasets, while Section 4.2 describes the pipeline that converts raw egocentric videos into pseudo-action labels compatible with the unified interface defined in Section 3.1. Figure 3 provides a visual overview of this conversion process.

4.1 Heterogeneous Data Sources

Robot demonstrations and simulation. The robot portion consists of AgiBot Alpha/Beta demonstrations, Galaxea R1Lite data, AgiBot DigitalWorld simulation rollouts, RoboCasa Tabletop simulation data (24 tasks, 1,000 episodes each, GR1 humanoid robot), and 1,800+ hours of self-collected Galbot demonstrations. These platforms span humanoid (AgiBot G1), single-arm wheeled (Galaxea R1Lite), and mobile bimanual (Galbot) embodiments, with control frequencies ranging from 10 to 30 Hz and end-effector poses logged in different reference frames depending on the platforms. This heterogeneity exposes representation mismatch and motivates the unified interface introduced in Section 3.1. All sources provide sensor-grounded end-effector action labels, which serve as high-fidelity supervision for the primary action expert in Section 3.2.

Human egocentric videos. The human-video portion draws from six sources: Ego4D [25], EgoExo4D [27], EPIC-KITCHENS-100 [46], HOI4D [47], EgoDex [28], and Xperience-10M [48]. Together, they span diverse kitchens, homes, and workshops, capturing long-tail manipulation behaviors that are difficult to cover via robot teleoperation alone. Since their action labels are inferred from vision-based pipelines rather than physical sensors, we treat them as pseudo-action-labeled supervision and route them through the reliability-aware human objective in Section 3.2.

Table 1 ACE-EGO-0 pretraining data pool. Hours are computed from dataset metadata as frames$/(\text{fps} \times 3600)$; Galbot hours are reported from our self-collected collections.

Source	Episodes	Frames	Hours	Supervision
Ego4D	948,683	23,396,157	216.6	Pseudo-action
EgoExo4D	41,414	1,110,275	10.3	Pseudo-action
EPIC-KITCHENS-100	74,788	3,486,432	32.3	Pseudo-action
HOI4D	2,966	774,275	7.2	Pseudo-action
EgoDex	327,317	83,894,075	776.8	Pseudo-action
Xperience-10M	99,027	31,370,900	435.7	Pseudo-action
Human video subtotal	1,494,195	144,032,114	1,478.9	Pseudo-action
AgiBot Alpha/Beta	116,013	209,284,239	1,937.8	Robot action
Galaxea R1Lite	20,662	26,358,560	488.1	Robot action
AgiBot DigitalWorld	47,910	24,333,788	225.3	Robot action
RoboCasa Tabletop	24,000	6,020,058	83.6	Robot action
Galbot self-collected	∼60,000	∼194M	1,800+	Robot action
Robot subtotal	∼268,585	∼460M	4,534.8+	Robot action
Total	∼1,762,780	∼604M	6,013.7+	Mixed

4.2 Egocentric Video-to-Action Conversion

Generating pseudo-labeled actions compatible with robotic data from large-scale video datasets requires bridging two major challenges: the *structural discrepancy*, since 2D video carries no metric 3D hand trajectories, and the *behavioral discrepancy*, since not every clip contains a clean manipulation primitive worth supervising on. We address both with a five-stage pipeline (Figure 4) that includes clip-level filtering, geometric recovery, action formatting, and fidelity-based quality control. Running this pipeline over six egocentric video datasets, we produce 1,478 hours of pseudo-action-labeled clips that share the same camera-space action format as robot data and enter the unified action space of Section 3.1. All quantitative thresholds are collected in Table 2. We describe each stage in detail below.

Stage 1: Dataset curation. We begin with publicly available human video collections and select sources that satisfy three criteria: an egocentric viewpoint, diverse real-world interaction scenes, and high-quality action-centric captions. This process yields the six datasets listed in Table 1, which forms our human video pool. We then standardize all sources into a unified storage format with consistent metadata fields, including clip identifiers, frame indices, camera intrinsics (when available), narrations, and licensing information. For sources that provide only video-level annotations, we split videos into clips. We discard clips that are shorter than 4 seconds or longer than 30 seconds, as they are unlikely to contain complete manipulation primitives at the downstream temporal granularity.

Stage 2: Video selection. The previous stage produces a large pool of egocentric videos that vary substantially in interaction quality and manipulation relevance. Before applying computationally intensive geometric reconstruction, we adopt an *ego-interaction filter* to remove clips that are unlikely to provide useful action supervision. The filter targets videos with limited human-object interaction and employs several lightweight cues to identify such cases. Among them, strong face detections serve as an effective signal of non-egocentric or observer-centric viewpoints, which rarely contain usable manipulation trajectories. We therefore discard clips whose maximum face-detection confidence exceeds a predefined threshold. A subsequent *image captioning-based filter* retains only clips whose narrations contain at least one manipulation verb and one manipulable object noun, further enriching the dataset with object-centric interaction behaviors.

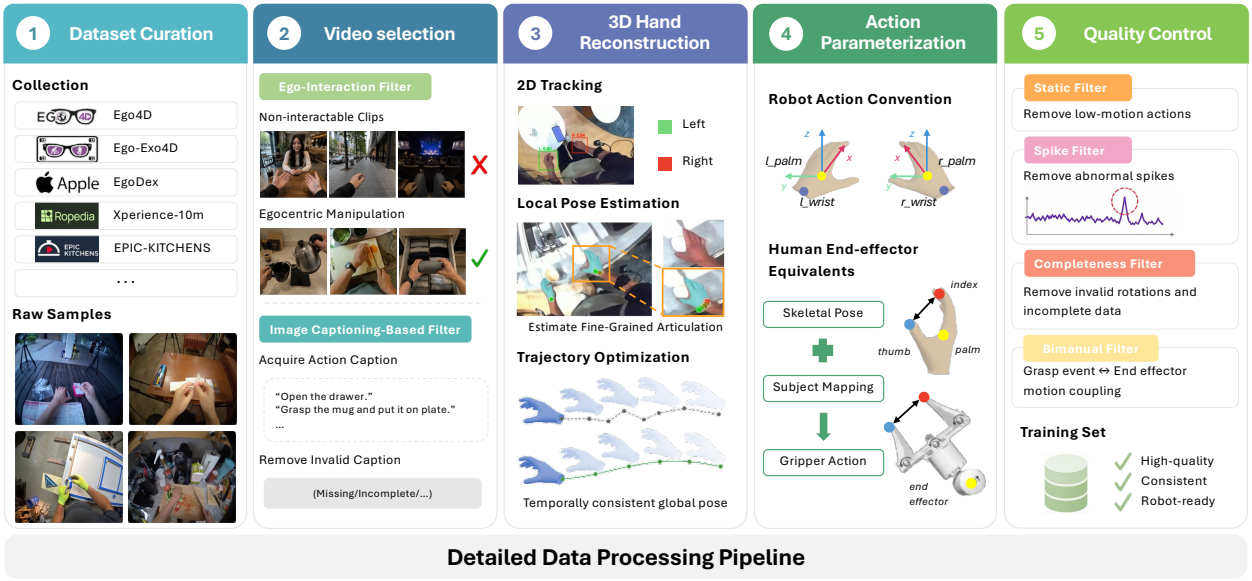


Figure 4 Pipeline for converting raw egocentric videos into camera-space pseudo-actions. The pipeline consists of five stages: (1) dataset curation; (2) video selection using ego-interaction and image captioning-based filters; (3) 3D hand reconstruction, including 2D tracking, local pose estimation, and trajectory optimization; (4) action parameterization using robot action conventions and human end-effector equivalents; and (5) quality control through multiple filtering.

Stage 3: 3D hand reconstruction. Hand reconstruction is performed in three sub-stages: 2D tracking, local pose estimation, and global trajectory optimization. We first apply a SAM3-based [49] tracker to obtain temporally consistent hand bounding boxes and segmentation masks throughout each clip, and discard detections with keypoint confidence below τ_{kp} or track length below $\ell_{\min}$ frames. We then feed the retained hand crops into HaMeR [13], which reconstructs MANO shape and pose parameters $\{\beta, \theta_t, \mathbf{t}_t^{\text{local}}\}_{t=1}^{T}$ for each frame in hand-related clips. Since per-frame reconstruction suffers from depth ambiguity, occlusions, and temporal jitter, we further perform a two-stage global trajectory optimization inspired by [50], where the first stage (N_{root} iterations) estimates globally consistent root translation and orientation, and the second stage (N_{smooth} L-BFGS iterations) jointly minimizes reprojection error and a temporal smoothness regularizer:

$$\mathcal{L}_{\text{smooth}} = \mathcal{L}_{\text{reproj}} + \lambda_{\text{tv}} \sum_{t} \left\| \mathbf{t}_{t+1}^{\text{global}} - 2\mathbf{t}_{t}^{\text{global}} + \mathbf{t}_{t-1}^{\text{global}} \right\|_2^2, \tag{10}$$

where $\mathcal{L}_{\text{reproj}}$ denotes the 2D keypoint reprojection loss, $\mathbf{t}_t^{\text{global}}$ is the optimized global hand root translation at frame t, and λ_{tv} controls the strength of temporal smoothness regularization. Both optimization stages leverage per-frame camera poses $(\mathbf{R}_t^{\text{cam}}, \mathbf{t}_t^{\text{cam}})$ estimated by VIPE [51], enabling conversion of local reconstructions into temporally coherent 3D trajectories in a shared world coordinate frame. The optimized global trajectory is used only for temporal consistency; the final pseudo-action labels are transformed back into the corresponding head-camera frame before training.

Stage 4: Action parameterization. The parameterization itself (wrist origin, palm-plane orientation, thumb-to-palm gripper proxy) is defined in Section 3.1.1. Here we explain two implementation details. *Storage layout.* On disk, each per-hand action is stored as a 16-dimensional bimanual vector: 3 position +3 XYZ Euler +1 gripper +1 activity flag, per hand; 8D $\times$ 2 hands $=$ 16D total. At training time the Euler angles are converted to the continuous 6D rotation representation [45], producing the 22-dimensional action vector, defined in Section 3.1.1. *Gripper normalization.* Thumb-to-palm distances d_t are linearly normalized to the robot gripper stroke range: $[d_{\min}^{\text{grip}}, d_{\max}^{\text{grip}}] = [0.04, 0.10]\,\text{m}$. Trajectories whose 10th–90th percentile range satisfies $d_{90} - d_{10} < \tau_{\text{grip}}$ are treated as degenerate (e.g., closed-fist motion with no grasp transition) and assigned a constant neutral gripper state.

Stage 5: Quality control. This stage removes corrupted or behaviorally implausible human episodes before they are collected into the mixed-source pretraining datasets. Here we apply four post-processing filters. *Completeness*

Table 2 Egocentric video pipeline hyperparameters used in Stages 1–5. Values are shared across the six human-video sources unless noted otherwise.

Stage	Hyperparameter	Value
Stage 1: Curation	Min clip duration	4 s
	Max clip duration	30 s
Stage 2: Selection	Face-detection threshold	0.5
	Caption verb/noun requirement	both present
Stage 3: Reconstruction	Keypoint confidence τ_{kp}	0.4
	Min track length $\ell_{\min}$	15 frames
	Root-fitting iterations N_{root}	30
	Smooth-fitting iterations N_{smooth}	200
	Smoothness weight λ_{tv}	1.0
Stage 4: Parameterization	Gripper stroke range $[d^{\text{grip}}_{\min}, d^{\text{grip}}_{\max}]$	$[0.04, 0.10]$ m
	Gripper-degeneracy threshold τ_{grip}	1.5 cm
	On-disk action dim / training action dim	16-D / 22-D
Stage 5: Filtering	Quaternion tolerance τ_{quat}	10^{-3}
	Static motion energy τ_{static}	source-specific
	Spike σ-multiplier κ_{spike}	3
	Spike frame fraction ρ_{spike}	5%

filter. We require each episode to be free of NaN/Inf values, contain contiguous frame indices, and satisfy quaternion normalization constraints: $|\|q\| - 1| \leq \tau_{\text{quat}}$. *Static filter.* We discard episodes when neither hand exhibits per-second motion energy above τ_{static}, indicating little or no meaningful interaction. *Spike filter.* We reject trajectories if inter-frame positional changes exceed $\kappa_{\text{spike}}\sigma$ of the per-episode velocity distribution on more than ρ_{spike} of frames, which typically indicates tracking failures or reconstruction artifacts. *Bimanual filter.* We remove episodes with implausible dual-arm behaviors based on anomalous inter-hand distance statistics or weak temporal correlation between the two hands. We record the corresponding thresholds in the released data manifests since they depend on source-level hand-detection density.

5 Experiments

5.1 Experimental Setup

We evaluate ACE-EGO-0 on two simulation benchmarks and one real-robot platform: RoboCasa GR1 TableTop [8], a humanoid tabletop benchmark with 24 pick-and-place and articulated-object tasks; RoboTwin 2.0 [52], a bimanual benchmark with 50 tasks and strong domain randomization; and an ARX bimanual platform with six real-world manipulation tasks. For simulation evaluation, we compare against GR00T-N1.6 [8], Qwen3PI, FLARE [53], ABot-M0 [54], JoyAI-RA [55], and DIAL [44] on RoboCasa, as well as $\pi_{0.5}$ [4], Motus [56], LingBot-VLA [57], ABot-M0 [54], JoyAI-RA [55], and Hy-VLA [58] on RoboTwin 2.0 (full per-task comparisons including π_0 are in Appendix C.5). For the physical real-robot evaluation, we compare against fine-tuned $\pi_{0.5}$ and GR00T-N1.7 [8], adopting the N1.7 version to leverage its latest optimizations for physical deployment. All models are trained in a multi-task setting and evaluated by task success rate. Model architecture, training protocol, and evaluation details are provided in Appendix B.

Camera-space inference. To execute these camera-space action chunks on a physical robot, we apply the inverse of the camera extrinsic used during data standardization:

$$\hat{p}_s = R^{\top}_{\text{cam}\leftarrow s}(\hat{p}_{\text{cam}} - t_{\text{cam}\leftarrow s}), \qquad \hat{R}_{s,ee} = R^{\top}_{\text{cam}\leftarrow s}\hat{R}_{\text{cam},ee}, \tag{11}$$

where s denotes the robot's execution frame (e.g., base or torso). The 6D rotation output is first reconstructed into a full rotation matrix via Gram–Schmidt orthogonalization before applying the inverse transform. Because ACE-EGO-0 predicts actions in the head-camera coordinate frame, deployment only requires a standard extrinsic transform

Table 3 Evaluation results on the RoboCasa GR1 TableTop benchmark (selected tasks). Success rates (%) over 50 rollouts per task. Full 24-task results are in Appendix C.4.

Task	GR00T-N1.6	Qwen3PI	FLARE	ABot-M0	JoyAI-RA	DIAL	ACE-EGO-0
CuttingboardToCardboardbox	46.5	46.0	54.0	58.0	46.0	–	**84.0**
PlacematToTieredshelf	28.5	28.0	26.0	26.0	14.0	–	**44.0**
PlateToPlate	78.7	48.0	76.0	64.0	88.0	–	**98.0**
PlateToBowl	57.0	52.0	50.0	54.0	48.0	–	**68.0**
TrayToPlate	71.0	64.0	64.0	68.0	88.0	–	**90.0**
… (24 tasks total; see Appendix for full results)							
Average (24 tasks)	47.6	43.9	55.0	58.3	63.2	70.2	**72.8**

Table 4 Overall evaluation results on the RoboTwin 2.0 benchmark. Success rates (%) averaged over 50 tasks, 100 trials per task. Easy denotes the clean setting and Hard denotes the randomized setting. Full per-task results are in Appendix C.5.

Method	$\pi_{0.5}$		Motus		LingBot-VLA		ABot-M0		JoyAI-RA		Hy-VLA		ACE-EGO-0	
	Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard
Success Rate	82.74	76.76	88.66	87.02	88.56	86.68	86.06	85.08	90.48	89.28	90.9	90.1	**91.12**	**90.62**

to convert predicted camera-frame end-effector poses into the robot coordinate frame. A new embodiment can be integrated by registering its URDF to obtain a morphology token and executing the resulting poses with its own low-level controller. Thus, camera-space prediction removes the need for the policy to learn source-specific coordinate transforms, while embodiment differences are handled by morphology conditioning.

5.2 Simulation Results

5.2.1 RoboCasa GR1 TableTop

RoboCasa GR1 TableTop evaluates humanoid tabletop manipulation on the GR1 platform across 24 tasks: 18 pick-and-place rearrangement tasks and 6 articulated-object interaction tasks. We train one model jointly on all 24 tasks and report mean success rate over 50 rollouts per task.

As shown in Table 3, ACE-EGO-0 achieves **72.8%** average success rate, surpassing all baselines including DIAL [44] (70.2%), JoyAI-RA [55] (63.2%), ABot-M0 [54] (58.3%), and FLARE [53] (55.0%). The gains are consistent across both articulated-object interaction and pick-and-place rearrangement task categories, suggesting that the camera-space action interface and reliability-aware training generalize broadly rather than benefiting a narrow subset of tasks.

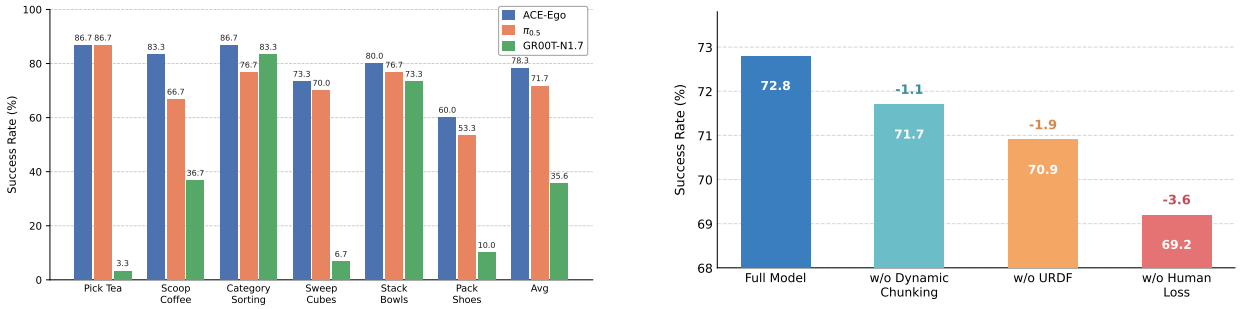
5.2.2 RoboTwin 2.0

RoboTwin 2.0 is a bimanual tabletop manipulation benchmark covering 50 tasks with strong domain randomization. We train on 2,500 clean demonstrations (50 per task) plus 25,000 randomized demonstrations (500 per task), and evaluate under both Easy/Clean and Hard/Randomized settings. Overall results are shown in Table 4, with full per-task results in Appendix C.5, Table 10.

ACE-EGO-0 achieves **91.12%** average success rate on the Easy/Clean setting and **90.62%** on the Hard/Randomized setting, surpassing JoyAI-RA by 0.64% and 1.34%, respectively. The improvement is distributed across diverse manipulation primitives—grasping, placement, tool use, and bimanual coordination—indicating that the unified pretraining recipe transfers effectively to multi-task bimanual control under strong domain randomization.

5.3 Real-Robot Evaluation

We evaluate ACE-EGO-0 on an ARX bimanual platform equipped with a head-mounted RGB-D camera, controlled via camera-space delta end-effector commands. The policy outputs actions directly in the head-camera coordinate frame and is deployed by simply applying a single camera extrinsic at inference time.



(a) Real-robot results on the ARX bimanual platform vs. $\pi_{0.5}$. Trials: 30 per task.

(b) Component ablation on RoboCasa GR1 TableTop. Each bar shows the effect of removing one component from the full model.

Figure 5 Real-robot evaluation (a) and ablation study (b) for ACE-EGO-0.

We evaluate on six manipulation tasks of increasing complexity—spanning single-arm pick-and-place, long-horizon multi-step manipulation, contact-rich bimanual coordination, and language-grounded semantic reasoning (see Appendix C.1 for full descriptions). We compare against two strong baseline methods: $\pi_{0.5}$ fine-tuned on the same downstream task data, and GR00T-N1.7. A trial is considered successful only if the robot completes the entire task sequence without human intervention; per-task success criteria are detailed in Appendix C.2, and the quantitative results are summarized in Figure 5(a).

ACE-EGO-0 achieves a **78.3%** average success rate across the six tasks, outperforming $\pi_{0.5}$ (71.7%) by 6.6%, and demonstrating a decisive margin over GR00T-N1.7, which struggles on several long-horizon sequences and obtains a 35.6% average success rate. Specifically, ACE-EGO-0 leads across five out of the six tasks. On Scoop Coffee, a contact-rich bimanual task requiring tight spatiotemporal coordination between both arms, ACE-EGO-0 achieves 86.7%, outperforming $\pi_{0.5}$ (70.0%) by 16.7% and GR00T-N1.7 (36.7%) by 50.0%.

In the multi-class object placement task, Category Sorting, ACE-EGO-0 maintains a steady performance of 90.0%, compared to 80.0% for $\pi_{0.5}$ and 83.3% for GR00T-N1.7. While GR00T-N1.7 exhibits reasonable capability on relatively structured setups such as Stack Bowls (73.3%), its execution consistency drops sharply on tasks that require extended horizontal trajectories or explicit bimanual coordination, such as Sweep Cubes (6.7%).

In sharp contrast, ACE-EGO-0 demonstrates its clear advantage in spatiotemporal alignment on Scoop Coffee, a contact-rich bimanual task requiring tight synchronization between both arms, sustaining an 86.7% success rate while GR00T-N1.7 falls to 36.7%. On Pack Shoes, which features the longest operational sequence including a delicate lid-closing phase, all evaluated models experience a visible degradation in performance. This joint performance drop suggests that managing compounding trajectory drift over long-horizon manipulation chains remains a common shared challenge for existing pretrained VLA architectures.

5.4 Ablation Studies

Component ablation. We ablate three components of ACE-EGO-0 on RoboCasa GR1 TableTop, removing one at a time from the full model and reporting the average success rates of the checkpoints trained for 190K steps (Figure 5(b)). Removing either one of the three components reduces performance. Removing morphology tokens makes the success rate drop from 72.8% to 70.9% (-1.9%): even though all sources share the same camera-space action format, different robot platforms have different kinematic structures, and the morphology tokens provide the action expert kinematics-related information. Removing time-aligned action chunking drops the success rate to 71.7% (-1.1%): a fixed number of actions now covers different physical durations across datasets collected at different frame rates and introduces temporal inconsistency between the mixed-source data. Removing the reliability-aware human auxiliary loss leads to the largest success rate drop to 69.2% (-3.6%): without label-quality weighting, noisy pseudo-actions from human videos receive equal supervision weight as sensor-logged robot actions, which confuses the action expert training.

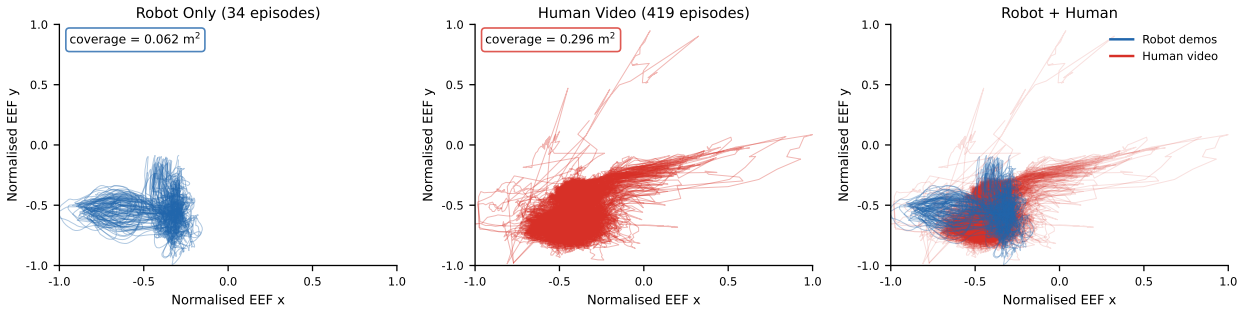


Figure 6 Right end-effector trajectories for the Sweep Cubes fine-tuning data, projected onto the horizontal plane. Axes are remapped to $[-1, 1]$ for readability; each panel uses the same scale. **Left:** 34 robot demonstrations are concentrated in a small region (0.062 m^2 convex-hull area). **Middle:** 419 human video episodes cover a substantially broader area (0.296 m^2, 4.8× larger). **Right:** both sources overlaid, showing the robot cluster embedded within the wider human distribution.

Data source ablation. We also evaluate three pretraining configurations on RoboCasa GR1 TableTop to assess the contribution of each data source (see Table 5).

Table 5 Pretraining data ablation on RoboCasa GR1 TableTop (success rate).

Pretraining Configuration	Success Rate (%)
Robot + Human (full ACE-EGO-0)	**72.8**
Robot Only (no human video)	68.3
From Qwen (no embodied pretrain)	65.4

The success rate increases with each additional data source. The Qwen-initialized model without embodied pretraining reaches 65.4%. Adding robot data raises the success rate to 68.3% (+2.9%), showing that embodied pretraining provides action-level knowledge that pure language-vision pretraining does not. Adding human videos further raises the rate to 72.8% (+4.5%, the largest single gain), showing that human videos contribute diverse behavioral coverage beyond the robot demonstrations alone.

5.5 Human Data for Augmented Fine-Tuning

We investigate how human egocentric videos improve task-specific adaptation when robot demonstration data alone are insufficient. Starting from the pretrained ACE-EGO-0 checkpoint, we fine-tune on the Sweep Cubes task using only 34 robot demonstrations (2 sessions, ∼45.8K frames). With robot data alone, the policy achieves only a 10% success rate (1/10 trials).

Figure 6 shows the reason: The 34 robot demonstrations occupy only a narrow region of the action space, covering only 0.062 m^2 of the end-effector workspace. The 419 episodes of task-matched human video spread across 0.296 m^2, 4.8× broader coverage, filling in motion patterns that the limited robot data does not include. Augmenting the fine-tuning mixture with this human video (∼117.5K frames) increases the success rate to 40% (4/10 trials), a 4× improvement, confirming that human video provides complementary action coverage and substantially recovers performance in data-scarce fine-tuning regimes.

6 Conclusion

We presented ACE-EGO-0, a VLA pretraining framework that jointly resolves representational and label-quality heterogeneity when learning from large-scale human and multi-embodiment robot data. ACE-EGO-0 resolves representation heterogeneity through a unified action representation, aligning heterogeneous sources along spatial, structural, and temporal spaces via camera-space actions, cross-embodiment morphology tokens, and time-aligned action chunking. It further addresses the supervision-quality mismatch through a reliability-aware training objective that provides

noise-resilient supervision for large-scale pseudo-action labels. Instantiated on a 6.0K+ hour pool spanning multiple robot platforms, simulation environments, and 1.48K hours of large-scale egocentric human video, ACE-EGO-0 achieves 72.8% on RoboCasa GR1 TableTop and 91.12%/90.62% on RoboTwin 2.0 Easy/Hard splits, outperforming all compared methods; human-augmented fine-tuning further demonstrates a 4× improvement in data-scarce regimes. On a real bimanual ARX platform, ACE-EGO-0 reaches a 78.3% average success rate across six physical manipulation tasks, consistently outperforming fine-tuned $\pi_{0.5}$ (71.7%) and demonstrating a decisive margin over GR00T-N1.7 (35.6%), with prominent capabilities in multi-step sequential execution and coordinated bimanual control.

7 Limitations

While ACE-EGO-0 demonstrates strong performance across simulation and real-world benchmarks, several directions remain open. Our current evaluation focuses on tabletop manipulation; extending to mobile manipulation, whole-body humanoid control, or deformable-object tasks would test the generality of the camera-space action interface under more diverse spatial conventions and longer task horizons. The pretraining pool, though large, does not yet include dexterous hand data or force/torque sensing; incorporating richer modalities could further improve contact-rich manipulation. Finally, scaling the human-video portion and improving the fidelity of pseudo-action pipelines—particularly for rotation and fine-grained finger motion—would allow the reliability-aware objective to supervise additional action dimensions beyond position, potentially unlocking stronger transfer from human demonstrations to robot control.

Acknowledgments

If a paper is accepted, the final camera-ready version will (and probably should) include acknowledgments. All acknowledgments go at the end of the paper, including thanks to reviewers who gave useful comments, to colleagues who contributed to the ideas, and to funding agencies and corporate sponsors that provided financial support.

References

[1] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *Robotics: Science and Systems XIX*, 2023.

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

[3] K. Black, N. Brown, D. Driess, A. Esmail, M. R. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, L. Smith, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025. doi:10.15607/RSS.2025.XXI.010.

[4] K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. R. Equi, C. Finn, N. Fusai, M. Y. Galliker, D. Ghosh, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, D. LeBlanc, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, A. Z. Ren, L. X. Shi, L. Smith, J. T. Springenberg, K. Stachowicz, J. Tanner, Q. Vuong, H. Walke, A. Walling, H. Wang, L. Yu, and U. Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In J. Lim, S. Song, and H.-W. Park, editors, *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pages 17–40. PMLR, 27–30 Sep 2025. URL https://proceedings.mlr.press/v305/black25a.html.

[5] A. ONeill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[6] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, T. Kreiman, Y. Tan, L. Y. Chen, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[7] L. Wang, X. Chen, J. Zhao, and K. He. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi:10.52202/079017-3952.

[8] NVIDIA, J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang, J. Jang, Z. Jiang, J. Kautz, K. Kundalia, L. Lao, Z. Li, Z. Lin, K. Lin, G. Liu, E. Llontop, L. Magne, A. Mandlekar, A. Narayan, S. Nasiriany, S. Reed, Y. L. Tan, G. Wang, Z. Wang, J. Wang, Q. Wang, J. Xiang, Y. Xie, Y. Xu, Z. Xu, S. Ye, Z. Yu, A. Zhang, H. Zhang, Y. Zhao, R. Zheng, and Y. Zhu. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025. doi:10.48550/arXiv.2503.14734. URL https://arxiv.org/abs/2503.14734.

[9] J. Zheng, J. Li, Z. Wang, D. Liu, X. Kang, Y. Feng, Y. Zheng, J. Zou, Y. Chen, J. Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. *arXiv preprint arXiv:2510.10274*, 2025.

[10] S. Kareer, D. Patel, R. Punamiya, P. Mathur, S. Cheng, C. Wang, J. Hoffman, and D. Xu. Egomimic: Scaling imitation learning via egocentric video. In *2025 IEEE International Conference on Robotics and Automation*, pages 13226–13233. IEEE, 2025.

[11] R. Yang, Q. Yu, Y. Wu, R. Yan, B. Li, A.-C. Cheng, X. Zou, Y. Fang, X. Cheng, R.-Z. Qiu, H. Yin, S. Liu, S. Han, Y. Lu, and X. Wang. EgoVLA: Learning vision-language-action models from egocentric human videos. *arXiv preprint arXiv:2507.12440*, 2025. doi:10.48550/arXiv.2507.12440.

[12] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, and C. Finn. Humanplus: Humanoid shadowing and imitation from humans. In P. Agrawal, O. Kroemer, and W. Burgard, editors, *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 2828–2844. PMLR, 2025.

[13] G. Pavlakos, D. Shan, I. Radosavovic, A. Kanazawa, D. Fouhey, and J. Malik. Reconstructing hands in 3d with transformers. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9826–9836. IEEE, June 2024. doi:10.1109/cvpr52733.2024.00938.

[14] R. A. Potamias, J. Zhang, J. Deng, and S. Zafeiriou. Wilor: End-to-end 3d hand localization and reconstruction in-the-wild. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12242–12254, 2025.

[15] J. Romero, D. Tzionas, and M. J. Black. Embodied hands: modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 36(6):1–17, November 2017. doi:10.1145/3130800.3130883.

[16] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong, et al. Openvla: An open-source vision-language-action model. In *Conference on Robot Learning*, pages 2679–2713. PMLR, 2025.

[17] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. In *International Conference on Learning Representations*, volume 2025, pages 29982–30009, 2025.

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

[19] J. Zheng, J. Li, D. Liu, Y. Zheng, Z. Wang, Z. Ou, Y. Liu, J. Liu, Y.-Q. Zhang, and X. Zhan. Universal actions for enhanced embodied foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22508–22519, 2025.

[20] S. Ye, J. Jang, B. Jeon, S. J. Joo, J. Yang, B. Peng, A. Mandlekar, R. Tan, Y.-W. Chao, B. Y. Lin, L. Liden, K. Lee, J. Gao, L. Zettlemoyer, D. Fox, and M. Seo. Latent action pretraining from videos. In *International Conference on Learning Representations*, 2025.

[21] S. Liu, B. Li, K. Ma, L. Wu, H. Tan, X. Ouyang, H. Su, and J. Zhu. Rdt2: Exploring the scaling limit of umi data towards zero-shot cross-embodiment generalization. *arXiv preprint arXiv:2602.03310*, 2026.

[22] D. Qu, H. Song, Q. Chen, Y. Yao, X. Ye, J. Gu, Z. Wang, Y. Ding, B. Zhao, D. Wang, and X. Li. SpatialVLA: Exploring spatial representations for visual-language-action models. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025. doi:10.15607/RSS.2025.XXI.011.

[23] H. Zhen, X. Qiu, P. Chen, J. Yang, X. Yan, Y. Du, Y. Hong, and C. Gan. 3D-VLA: A 3D vision-language-action generative world model. In R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 61229–61245. PMLR, 21–27 Jul 2024.

[24] R. Zheng, Y. Liang, S. Huang, J. Gao, H. Daumé III, A. Kolobov, F. Huang, and J. Yang. TraceVLA: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. In *International Conference on Learning Representations*, 2025.

[25] K. Grauman et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18973–18990. IEEE, June 2022. doi:10.1109/cvpr52688.2022.01842.

[26] D. Damen, H. Doughty, G. M. Farinella, A. Furnari, E. Kazakos, J. Ma, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. Rescaling egocentric vision: Collection, pipeline and challenges for EPIC-KITCHENS-100. *International Journal of Computer Vision*, 130(1):33–55, 2022. doi:10.1007/s11263-021-01531-2.

[27] K. Grauman, A. Westbury, L. Torresani, K. Kitani, J. Malik, T. Afouras, K. Ashutosh, V. Baiyya, S. Bansal, B. Boote, E. Byrne, Z. Chavis, J. Chen, F. Cheng, F.-J. Chu, et al. Ego-Exo4D: Understanding skilled human activity from first- and third-person perspectives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19383–19400, 2024.

[28] R. Hoque, P. Huang, D. J. Yoon, M. Sivapurapu, and J. Zhang. EgoDex: Learning dexterous manipulation from large-scale egocentric video. *arXiv preprint arXiv:2505.11709*, 2025. doi:10.48550/arXiv.2505.11709.

[29] R. Zheng, D. Niu, Y. Xie, J. Wang, M. Xu, Y. Jiang, F. Castañeda, F. Hu, Y. L. Tan, L. Fu, T. Darrell, F. Huang, Y. Zhu, D. Xu, and L. Fan. EgoScale: Scaling dexterous manipulation with diverse egocentric human data. *arXiv preprint arXiv:2602.16710*, 2026. doi:10.48550/arXiv.2602.16710.

[30] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3M: A universal visual representation for robot manipulation. In K. Liu, D. Kulic, and J. Ichnowski, editors, *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 892–909. PMLR, 14–18 Dec 2023. URL https://proceedings.mlr.press/v205/nair23a.html.

[31] Y. J. Ma, S. Sodhani, D. Jayaraman, O. Bastani, V. Kumar, and A. Zhang. VIP: Towards universal visual reward and representation via value-implicit pre-training. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=YJ7o2wetJ2.

[32] Y. J. Ma, V. Kumar, A. Zhang, O. Bastani, and D. Jayaraman. LIV: Language-image representations and rewards for robotic control. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *Proceedings of the 40th*

International Conference on Machine Learning, volume 202 of *Proceedings of Machine Learning Research*, pages 23301–23320. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/ma23b.html.

[33] T. Xiao, I. Radosavovic, T. Darrell, and J. Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022. URL https://arxiv.org/abs/2203.06173.

[34] A. Majumdar, K. Yadav, S. Arnaud, Y. J. Ma, C. Chen, S. Silwal, A. Jain, V.-P. Berges, T. Wu, J. Vakil, P. Abbeel, J. Malik, D. Batra, Y. Lin, O. Maksymets, A. Rajeswaran, and F. Meier. Where are we in the search for an artificial visual cortex for embodied intelligence? In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/022ca1bed6b574b962c48a2856eb207b-Abstract-Conference.html.

[35] S. Karamcheti, S. Nair, A. S. Chen, T. Kollar, C. Finn, D. Sadigh, and P. Liang. Language-driven representation learning for robotics. In *Robotics: Science and Systems*, 2023. URL https://arxiv.org/abs/2302.12766.

[36] K. Q. Lin, J. Wang, M. Soldan, M. Wray, R. Yan, E. Z. Xu, D. Gao, R.-C. Tu, W. Zhao, W. Kong, C. Cai, H. Wang, D. Damen, B. Ghanem, W. Liu, and M. Z. Shou. Egocentric video-language pretraining. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/31fb284a0aaaad837d2930a610cd5e50-Abstract-Conference.html.

[37] Y. Zhao, I. Misra, P. Krähenbühl, and R. Girdhar. Learning video representations from large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6586–6597, June 2023.

[38] J. Li, Y. Zhu, Y. Xie, Z. Jiang, M. Seo, G. Pavlakos, and Y. Zhu. OKAMI: Teaching humanoid robots manipulation skills through single video imitation. In P. Agrawal, O. Kroemer, and W. Burgard, editors, *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 299–317. PMLR, 2025.

[39] M. Lepert, J. Fang, and J. Bohg. Phantom: Training robots without robots using only human videos. In J. Lim, S. Song, and H.-W. Park, editors, *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pages 4545–4565. PMLR, 2025.

[40] L. Y. Zhu, P. Kuppili, R. Punamiya, P. Aphiwetsa, D. Patel, S. Kareer, S. Ha, and D. Xu. EMMA: scaling mobile manipulation via egocentric human data. *IEEE Robotics Autom. Lett.*, 11(3):3087–3094, 2026. doi:10.1109/LRA.2026.3653320. URL https://doi.org/10.1109/LRA.2026.3653320.

[41] G. Li, Y. Lyu, Z. Liu, C. Hou, Y. Xu, J. Zhang, and S. Zhang. H2R: A human-to-robot data augmentation for robot pre-training from videos. *arXiv preprint arXiv:2505.11920*, 2025. doi:10.48550/arXiv.2505.11920.

[42] V. Liu, A. Adeniji, D. Zhan, S. Haldar, R. Bhirangi, P. Abbeel, and L. Pinto. Egozero: Robot learning from smart glasses. *arXiv preprint arXiv:2505.20290*, 2025. doi:10.48550/arXiv.2505.20290.

[43] J. Shi, Z. Zhao, T. Wang, I. Pedroza, A. Luo, J. Wang, J. Ma, and D. Jayaraman. Zeromimic: Distilling robotic manipulation skills from web videos. In *2025 IEEE International Conference on Robotics and Automation*, pages 16939–16947. IEEE, 2025. doi:10.1109/ICRA55743.2025.11128283.

[44] Y. Chen, Y. Ge, H. Zhou, M. Ding, Y. Ge, and X. Liu. Dial: Decoupling intent and action via latent world modeling for end-to-end vla. *arXiv preprint arXiv:2603.29844*, 2026.

[45] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5745–5753, 2019.

[46] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. The epic-kitchens dataset: Collection, challenges and baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):4125–4141, November 2021. doi:10.1109/tpami.2020.2991965.

[47] Y. Liu, Y. Liu, C. Jiang, K. Lyu, W. Wan, H. Shen, B. Liang, Z. Fu, H. He, and H. Dong. Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[48] Ropedia. Xperience-10m: A large-scale egocentric multimodal dataset with structured 3d/4d annotations, 2026. Dataset.

[49] N. Carion, L. Gustafson, Y.-T. Hu, S. Debnath, R. Hu, D. Suris, C. Ryali, K. V. Alwala, H. Khedr, A. Huang, et al. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

[50] Z. Yu, S. Zafeiriou, and T. Birdal. Dyn-hamr: Recovering 4d interacting hand motion from a dynamic camera. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 27716–27726, 2025.

[51] J. Huang, Q. Zhou, H. Rabeti, A. Korovko, H. Ling, X. Ren, T. Shen, J. Gao, D. Slepichev, C.-H. Lin, et al. Vipe: Video pose engine for 3d geometric perception. *arXiv preprint arXiv:2508.10934*, 2025.

[52] T. Chen, Z. Chen, B. Chen, Z. Cai, Y. Liu, Z. Li, Q. Liang, X. Lin, Y. Ge, Z. Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

[53] R. Zheng, J. Wang, S. Reed, J. Bjorck, Y. Fang, F. Hu, J. Jang, K. Kundalia, Z. Lin, L. Magne, A. Narayan, Y. L. Tan, G. Wang, Q. Wang, J. Xiang, Y. Xu, S. Ye, J. Kautz, F. Huang, Y. Zhu, and L. Fan. Flare: Robot learning with implicit world modeling. In J. Lim, S. Song, and H.-W. Park, editors, *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pages 3952–3971. PMLR, 2025. URL https://proceedings.mlr.press/v305/zheng25a.html.

[54] Y. Yang, S. Zeng, T. Lin, X. Chang, D. Qi, J. Xiao, H. Liu, R. Chen, Y. Chen, D. Huo, F. Xiong, X. Wei, Z. Ma, and M. Xu. Abot-m0: Vla foundation model for robotic manipulation with action manifold learning. *arXiv preprint arXiv:2602.11236*, 2026.

[55] T. Zhang, Z. Yuan, D. Chi, P. Liu, D. Li, K. Hu, L. Zhang, J. Nie, Z. Wei, Z. Chen, Y. Tang, J. Li, Z. Xiang, M. Li, T. Luo, H. Wan, A. Li, L. Zhai, Z. Zhan, X. Bai, J. Cai, P. Cao, K. Chen, S. Chen, Y. Dai, S. Di, Y. Gong, C. Gui, Y. Guo, P. Hao, Q. He, H. Huang, K. Huang, Z. Huang, S. Jin, Y. Jin, A. Li, D. Li, J. Li, R. Li, Y. Li, Y. Li, J. Liang, F. Liu, J. Long, M. Luo, X. Pan, H. Shen, X. Tian, D. Wang, S. Wang, J. Xiong, H. Xu, W. Xu, Z. Yu, H. Zhang, J. Zhang, L. Zhao, C. Zhou, N. Duan, Y. Zhuang, and L. Lin. Joyai-ra 0.1: A foundation model for robotic autonomy, 2026.

[56] H. Bi, H. Tan, S. Xie, Z. Wang, S. Huang, H. Liu, R. Zhao, Y. Feng, C. Xiang, Y. Rong, H. Zhao, H. Liu, Z. Su, L. Ma, H. Su, and J. Zhu. Motus: A unified latent action world model, 2025. URL https://arxiv.org/abs/2512.13030.

[57] W. Wu, F. Lu, Y. Wang, S. Yang, S. Liu, F. Wang, Q. Zhu, H. Sun, Y. Wang, S. Ma, et al. A pragmatic vla foundation model. *arXiv preprint arXiv:2601.18692*, 2026.

[58] Tencent Robotics and Tencent Hy Team. Hy-Embodied-0.5-VLA: From vision-language-action models to a real-world robot learning stack. *arXiv preprint arXiv:2606.14409*, 2026.

A Additional Method Details

Figure 7 Camera-space action visualization across real robot demonstrations, simulation rollouts, and human egocentric video. All sources express end-effector or hand motion relative to the head-camera frame, making heterogeneous action labels comparable under the same observation-aligned coordinate convention.

A.1 Camera-Space Action Standardization and Layout

This subsection details the spatial alignment pipeline for both robot and human sources introduced in Sec. 3.1.1.

Coordinate Transformation. For robot sources, action labels are sensor-grounded end-effector poses. If an end-effector pose is reported in a source frame s (e.g., robot base or world frame), we convert it to the head-camera frame using the calibrated camera extrinsic as formulated in Eq. 1. For bimanual platforms, this transformation is applied independently to the left and right end-effectors. If a dataset already stores aligned camera-frame actions, the conversion is the identity.

Continuous 6D Orientation. To avoid the discontinuities of quaternions or Euler angles during training, orientations are normalized to a continuous 6D representation [45]. Quaternions are first converted to rotation matrices $R_{\text{cam},ee} \in \text{SO}(3)$. We then extract and concatenate the first two columns of the rotation matrix:

$$\text{rot6d}(R_{\text{cam},ee}) = \left[R_{\text{cam},ee}^{(:,1)};\ R_{\text{cam},ee}^{(:,2)}\right] \in \mathbb{R}^6. \tag{12}$$

Human Hand-Centric Frame Derivation. To parameterize human trajectories into the identical coordinate space, we construct a stable hand-centric coordinate frame $R_{\text{cam},hand} = [\mathbf{x}, \mathbf{y}, \mathbf{z}] \in \text{SO}(3)$ from the reconstructed hand mesh. We designate the wrist joint $\mathbf{p}_{\text{wrist}}$ as the origin. Let $\mathbf{p}_{\text{palm}}$ denote the palm centroid, computed as the mean of the index, middle, and ring fingertip positions. The orthogonal axes of the hand frame are constructed as:

$$\mathbf{x} = \frac{\mathbf{p}_{\text{palm}} - \mathbf{p}_{\text{wrist}}}{\|\mathbf{p}_{\text{palm}} - \mathbf{p}_{\text{wrist}}\|_2}, \quad \mathbf{z} = \hat{n}(\mathbf{p}_{\text{wrist}}, \mathbf{p}_{\text{thumb}}, \mathbf{p}_{\text{middle}}), \quad \mathbf{y} = \mathbf{z} \times \mathbf{x}, \tag{13}$$

where $\hat{n}(\mathbf{a}, \mathbf{b}, \mathbf{c})$ denotes the unit normal of the plane defined by points $\mathbf{a}, \mathbf{b}, \mathbf{c}$, with the sign chosen to point away from the palm. The resulting rotation matrix $R_{\text{cam},hand}$ is then converted to the continuous 6D representation. For gripper openness, the thumb-to-palm distance $d_t = \|\mathbf{p}_{\text{thumb},t} - \mathbf{p}_{\text{palm},t}\|_2$ is linearly normalized to the gripper stroke range of our robot platforms.

Unified 22-Dimensional Action Layout. After standardization, both robot and human trajectories are mapped into a unified 22-dimensional bimanual action vector $\mathbf{a} \in \mathbb{R}^{22}$. The vector is structured as a concatenation of symmetric 11-dimensional single-arm action blocks:

$$\mathbf{a} = [\mathbf{a}_{\text{left}};\ \mathbf{a}_{\text{right}}] \in \mathbb{R}^{22}, \tag{14}$$

where each arm's action block $\mathbf{a}_{\text{arm}} \in \mathbb{R}^{11}$ is defined as:

$$\mathbf{a}_{\text{arm}} = \Big[\underbrace{p_x, p_y, p_z}_{\text{Position (3D)}}, \underbrace{r_1, \ldots, r_6}_{\text{Continuous Orientation (6D)}}, \underbrace{g}_{\text{Gripper (1D)}}, \underbrace{\alpha}_{\text{Activity Flag (1D)}}\Big]. \tag{15}$$

The binary activity flag $\alpha \in \{0, 1\}$ indicates whether the corresponding arm is active in the dataset, allowing the policy to seamlessly handle both single-arm and bimanual embodiments.

Projection-Based Data Validation. We utilize camera projection as a validation signal to filter out tracking failures. Given a camera-frame end-effector position (X, Y, Z) and camera intrinsics (f_x, f_y, c_x, c_y), the projection onto the image plane is:

$$u = f_x \frac{X}{Z} + c_x, \qquad v = f_y \frac{Y}{Z} + c_y. \tag{16}$$

Frames with non-positive depth ($Z \leq 0$) or projections falling outside the image boundaries are flagged and masked out using the action validity mask M.

A.2 Robot Kinematic Graph Construction

For each robot embodiment with a URDF, we build the compact kinematic graph $\mathcal{G}_r$ used by the morphology encoder. This graph feeds only the morphology conditioning and never enters the shared vision-language trunk. Training samples carry a robot identifier rather than a URDF path. A registry maps each identifier to a canonical robot name, a URDF file, a base link, and left/right end-effector links. We cache the graph tensors by canonical name, so URDF parsing and graph construction happen once and are reused across training.

We use a joint-centric graph: each node is a URDF joint, and edges follow the parent–child relations of the kinematic tree. This puts the quantities most relevant to control—joint type, motion axis, origin transform, limits, and actuation state—directly on the nodes. The registered end-effector links define the left and right manipulation chains $\mathcal{C}_L^r$ and $\mathcal{C}_R^r$, which we use to mark action-relevant joints and to measure each joint's distance to the terminal joints of the two chains.

Each joint carries a fixed-dimensional descriptor with four groups of information: local kinematic attributes, range and actuation properties, graph-topological position, and relation to the left/right end-effector chains. In our implementation this is a 29-dimensional node feature vector. Stacking the descriptors for all joints gives a node feature matrix $X_r \in \mathbb{R}^{N_r \times 29}$ for robot r, where N_r is the number of URDF joints. The cached payload holds X_r, the normalized adjacency matrix, the left/right chain masks $\mathcal{C}_L^r, \mathcal{C}_R^r$, and joint metadata.

A.3 Morphology Encoder

This subsection details the URDF encoder E_{urdf} introduced in Sec. 3.1.2. It maps the cached graph $\mathcal{G}_r$ (Appendix A.2) to the body summary z_{body}^r and manipulation-chain summary z_{chain}^r that make up $E_{\text{urdf}}(\mathcal{G}_r)$ in Eq. 2.

The encoder has two stages: it first contextualizes each joint by message passing over the kinematic tree, then pools the joint states into the two summaries.

Message Passing. Starting from the joint descriptors X_r, the encoder runs L residual layers:

$$H^{(0)} = \phi_{\text{in}}(X_r), \qquad H^{(\ell+1)} = H^{(\ell)} + \phi_\ell\Big(\Big[H^{(\ell)}; \bar{A}_r H^{(\ell)}\Big]\Big), \quad \ell = 0, \ldots, L-1, \tag{17}$$

where $\bar{A}_r = D_r^{-1}(A_r + I)$ is the adjacency matrix with self-loops, row-normalized by its degree matrix D_r; ϕ_{in} and ϕ_ℓ are MLPs. At each layer, $\bar{A}_r H^{(\ell)}$ averages every joint with its neighbors, while the residual path preserves the joint's own state.

Pooling and Concatenation. Writing $\mathrm{mp}(\mathcal{S}) = \frac{1}{|\mathcal{S}|}\sum_{j\in\mathcal{S}} H_j^{(L)}$ for the mean of the final states over a joint set $\mathcal{S}$, the two summaries are:

$$z^r_{\mathrm{body}} = \rho_{\mathrm{body}}(\mathrm{mp}(\mathcal{J}_r)), \qquad z^r_{\mathrm{chain}} = \rho_{\mathrm{chain}}([\mathrm{mp}(\mathcal{C}^r_L);\, \mathrm{mp}(\mathcal{C}^r_R)]), \tag{18}$$

where $\mathcal{J}_r$ is the full joint set and $\mathcal{C}^r_L, \mathcal{C}^r_R$ are the left and right end-effector chains from Appendix A.2, and $\rho_{\mathrm{body}}, \rho_{\mathrm{chain}}$ are MLPs. The body summary captures the global embodiment, while the chain summary focuses on the kinematic paths most involved in manipulation. The final URDF representation is the concatenation of these two summaries:

$$E_{\mathrm{urdf}}(\mathcal{G}_r) = \left[z^r_{\mathrm{body}};\, z^r_{\mathrm{chain}}\right], \tag{19}$$

which is then projected by P_{morph} into the shared morphology token space as shown in Eq. 2.

A.4 Human Surrogate Morphology Embeddings

Human egocentric video has no robot URDF, so the encoder E_{urdf} of Appendix A.3 does not apply. Human sources still differ from one another in embodiment and capture conditions, and the action expert should be conditioned on these differences just as it is for robots. We therefore represent each human-video source by a learned surrogate embedding $e_d \in \mathbb{R}^D$ and project it with P_{surr} into the same morphology token as the URDF-conditioned robots (Eq. 2).

The surrogate absorbs stable source-level factors that the shared camera-space action representation does not explain: camera placement and field of view, the visual domain of each corpus, annotation quality, and source-specific action statistics. These factors stay roughly constant within a source but differ across sources, so a per-source embedding fits them better than a per-sample input. We can allocate the embedding per dataset (one e_d per human-video source) or share it across all human-video data, and we use the per-dataset variant by default. After the morphology-token interface, the action expert treats robot and human-video samples the same way: robots get the condition from a structured URDF graph, and human-video sources get it from learned surrogate embeddings. The surrogate embeddings e_d are randomly initialized and optimized end-to-end during pretraining alongside all other model parameters.

A.5 Reliability-Aware Human Auxiliary Loss Details

This subsection expands the mathematical formulation of the spatiotemporal reliability weight $W_{t,j}$ and the temporal smoothing summarized in Sec. 3.2. All quantitative thresholds are collected in Table 6.

Hierarchical Reliability Decomposition. As introduced in Eq. 6, the spatiotemporal reliability $W_{t,j}$ of channel j at step t is decomposed into a static channel-level prior ρ_j and a dynamic step-level weight $w_{t,j}$. We further decompose the step-level weight $w_{t,j}$ into a dataset-level prior w_{data} and a local step-level smoothness factor w_{step}, yielding the final hierarchical formulation:

$$W_{t,j} = \rho_j \cdot w_{\mathrm{data}}(d, h(j)) \cdot w_{\mathrm{step}}(t, h(j)), \tag{20}$$

where $h(j)$ maps action channel j to its corresponding hand (left or right). The dataset prior w_{data} sets a global quality ceiling for each source, while the step weight w_{step} modulates it locally in response to tracking anomalies.

Normalization. The human auxiliary loss $\mathcal{L}_{\mathrm{haux}}$ is normalized per sample by the total effective supervision weight:

$$Z = \sum_{t,j} M_{t,j}\, W_{t,j}. \tag{21}$$

This formulation ensures that the auxiliary loss is scale-invariant to the number of valid entries and concentrates supervision on highly reliable channels. When a minibatch contains no human sample, $\mathcal{L}_{\mathrm{haux}}$ is set to zero.

Step-Level Smoothness Weight. The time-step factor w_{step} down-weights segments whose motion is locally implausible, which typically indicates reconstruction error rather than genuine fast motion. For hand h we compute, from the clean position chunk, the first- and second-order differences:

$$\Delta p^h_t = \left\|p^h_t - p^h_{t-1}\right\|_2, \qquad \Delta^2 p^h_t = \left\|p^h_{t+1} - 2p^h_t + p^h_{t-1}\right\|_2, \tag{22}$$

which measure inter-frame speed and jerk, respectively. For each human-video dataset d and hand h, we precompute robust thresholds $\tau_{\text{jump}}(d,h)$ and $\tau_{\text{jerk}}(d,h)$ as the 95th percentiles of Δp_t^h and $\Delta^2 p_t^h$ over clean position chunks from that dataset. At training time, we compute:

$$q_{t,h} = \max\left(\frac{\Delta p_t^h}{\tau_{\text{jump}}(d,h)}, \frac{\Delta^2 p_t^h}{\tau_{\text{jerk}}(d,h)}\right). \tag{23}$$

The step weight is then formulated as:

$$w_{\text{step}}(t,h) = \begin{cases} 1, & q_{t,h} \le 1, \\ \max\{w_{\min}, \exp[-\alpha(q_{t,h}-1)]\}, & q_{t,h} > 1. \end{cases} \tag{24}$$

Thus, nominally smooth steps retain full weight, while unusually large jumps or jerks relative to the dataset-hand statistics are softly attenuated.

Dataset-Level Prior. Each human-video source carries a different reconstruction quality, so we attach a per-source, per-hand prior $w_{\text{data}}(d,h) \in (0,1]$. For dataset d and hand h this prior is estimated from the clean position trajectories of that source: we aggregate the fraction of frames surviving the sanity filters together with the median normalized jerk of the retained trajectories, and map sources with higher survival and lower jerk to priors closer to 1. The prior is computed once per source and held fixed during training.

Temporal Smoothing. Before constructing the auxiliary target velocity $(\tilde{\mathbf{a}} - \boldsymbol{\epsilon})$, we apply a temporal smoothing filter of window W_{smooth} to the clean human action targets. This suppresses high-frequency pose jitter introduced by per-frame hand mesh regression without altering the supervised dimensions or the $W_{t,j}$ weights, which are computed from the pre-smoothing chunk.

Table 6 Reliability-aware human supervision hyperparameters used in the human auxiliary loss (Section 3.2). Values are shared across the six human-video sources unless noted otherwise.

Component	Hyperparameter	Value
Auxiliary loss	Loss weight λ_{haux}	0.1
	Huber transition β	1.0
Human channel prior ρ_j	Position channels $\mathcal{P}$ (ρ_j)	1.0
	Rotation / gripper channels (ρ_{low})	0.001
	Position channel set $\mathcal{P}$	wrist xyz, both hands (6 dims)
Step weight	Jump threshold $\tau_{\text{jump}}(d,h)$	per-dataset/hand 95th percentile
	Jerk threshold $\tau_{\text{jerk}}(d,h)$	per-dataset/hand 95th percentile
	Attenuation sharpness α	1.5
	Minimum step weight $w_{\min}$	0.2
Dataset prior	Prior range w_{data}	[0.25, 1.0]
	Estimation	q95 jump/jerk ratio to robot reference
Smoothing	Smoothing window W_{smooth}	3 frames

B Training Details

B.1 Architecture, Training, and Evaluation Protocol

Model architecture. ACE-EGO-0 uses Qwen3-VL-4B-Instruct as the vision-language backbone and a flow-matching Diffusion Transformer (∼600M parameters) as the action expert. Images from head and wrist cameras are processed at 256×256 resolution, and actions are decoded in 4 flow-matching steps at inference. Full layer and dimension configurations are listed in Table 7.

Training protocol. Pretraining runs on 128×A800 (80GB) GPUs with AdamW and a cosine schedule; task-specific fine-tuning uses 16×A800 GPUs with the same optimizer settings. Full optimizer hyperparameters, learning rates, and schedule are listed in Table 8.

Evaluation protocol. RoboCasa evaluates 50 rollouts per task across 24 tasks. RoboTwin 2.0 evaluates 100 trials per task across 50 tasks under both Easy and Hard settings. Real-robot experiments use 30 trials per task. A trial is considered successful only if the robot completes the entire task sequence without human intervention; per-task real-robot success criteria are detailed in Appendix C.2.

Table 7 Model architecture configuration for ACE-EGO-0.

Component	Configuration
VLM backbone	Qwen3-VL-4B-Instruct (∼4B params)
Vision encoder	24 layers, patch size 16×16
Language model	36 layers, hidden size 2560
Input resolution	256×256 (head + wrist)
Action expert	Flow-matching DiT
Layers / hidden	36 / 1024
Attention heads	16 (head dim 64)
Parameters	∼600M
Inference decoding steps	4

B.2 Hyperparameters

Table 8 summarizes the key training hyperparameters for ACE-EGO-0 pretraining and fine-tuning.

Table 8 Training hyperparameters for ACE-EGO-0 pretraining and fine-tuning.

Hyperparameter	Value
VLM backbone	Qwen3-VL-4B-Instruct
Action expert	Flow-matching DiT (36 layers, 1024 hidden, 16 heads)
Action expert parameters	∼600M
Image resolution	256×256
Action horizon	dataset-specific $H_d = \mathrm{round}(f_d T^\star)$, with $T^\star = 2\,\mathrm{s}$ 40 steps for 20 Hz RoboCasa SFT sources
Flow matching inference steps	4
Optimizer	AdamW (β_1=0.9, β_2=0.95, ϵ=1e-8)
VLM learning rate	2e-5
Action expert learning rate	1e-4
LR schedule	Cosine with min LR 5e-7
Warmup steps	5000
Weight decay	1e-8
Gradient clipping	1.0
Batch size (per device)	8
Pretraining GPUs	128×A800 (80GB)
Pretraining steps	200K
Fine-tuning GPUs	16×A800 (80GB)
λ_{haux}	0.1
Human quality mode	dataset + step
Repeated diffusion steps	4

B.3 Dataset Mixtures and Sampling

Full dataset statistics are reported in Table 1 (main text). The pool is assembled from named dataset groups rather than from one monolithic corpus, which lets us control the sampling weight and preprocessing path of each source independently. The Ego4D entry combines cooking and non-cooking splits; hours are computed from LeRobot metadata as frames/(fps$\times$3600).

Sampling is performed at the dataset-group level. Each mixture entry has a sampling weight and a source type. Human-video sources are marked with the `human_video` source type and are routed through the camera-space pseudo-action path and reliability-aware human loss. Robot sources use their corresponding robot type and are supervised with the main robot action objective. This separation lets large but noisy human-video corpora contribute broad visual and behavioral coverage without overwhelming higher-fidelity robot demonstrations.

C Additional Experiments

C.1 Real-Robot Task Descriptions

The six real-robot tasks, ordered by increasing complexity, are:

- **Pick Tea**: grasp a shopping basket and place it at the workspace center, then pick up a tea box and drop it into the basket.
- **Scoop Coffee**: the right arm grasps a coffee scoop while the left arm holds a coffee canister; the right arm scoops coffee from the canister and pours it into a designated cup.
- **Category Sorting**: multiple objects (toiletries and beverages) are scattered on the workspace; the robot sorts each object into the corresponding bin based on semantic category.
- **Sweep Cubes**: the left arm holds a dustpan in a fixed pose while the right arm uses a broom to sweep cubes on the workspace into the dustpan.
- **Stack Bowls**: sequentially pick up three bowls from the workspace and stack them vertically.
- **Pack Shoes**: move a shoe box to the workspace center, sequentially place two shoes inside, and close the lid.

C.2 Real-Robot Success Criteria

Each real-robot trial is evaluated by a human judge. A trial is marked successful only if the robot completes the full task sequence without human intervention. The per-task success definitions are:

- **Pick Tea**: The shopping basket is placed at the workspace center and the tea box is dropped inside the basket.
- **Scoop Coffee**: Coffee is scooped from the canister and a visible amount is deposited into the designated cup.
- **Category Sorting**: All scattered objects are placed into their correct category bins (toiletries vs. beverages).
- **Sweep Cubes**: All cubes on the workspace are swept into the dustpan held by the left arm.
- **Stack Bowls**: All three bowls are picked up and stacked vertically without toppling.
- **Pack Shoes**: Both shoes are placed inside the shoe box and the lid is closed.

C.3 Qualitative Results

Figure 8 shows qualitative rollout sequences of ACE-EGO-0 on the real ARX bimanual platform. Each row visualizes key frames from a successful episode, illustrating the policy's ability to execute long-horizon multi-step manipulation, bimanual coordination, and contact-rich tool use in real-world settings.

C.4 Full RoboCasa GR1 TableTop Results

Table 9 reports per-task success rates on all 24 RoboCasa GR1 TableTop tasks.

C.5 Full RoboTwin 2.0 Results

Table 10 reports per-task success rates on all 50 RoboTwin 2.0 tasks under both Easy/Clean and Hard/Randomized settings.

Figure 8 Qualitative rollout sequences of ACE-EGO-0 on the real ARX bimanual platform. Each row shows key frames from a representative task, demonstrating the policy's capability across single-arm placement, bimanual coordination, and contact-rich manipulation.

Table 9 Full evaluation results on the RoboCasa GR1 TableTop benchmark. Success rates (%) over 50 rollouts per task.

Task	GR00T-N1.6	Qwen3PI	FLARE	ABot-M0	JoyAI-RA	ACE-EGO-0
CupToDrawerClose	8.5	42.0	46.0	**48.0**	**48.0**	36.0
PotatoToMicrowaveClose	41.5	42.0	30.0	50.0	**70.0**	58.0
MilkToMicrowaveClose	14.0	50.0	58.0	46.0	**84.0**	56.0
BottleToCabinetClose	51.5	26.0	66.0	**86.0**	84.0	78.0
WineToCabinetClose	16.5	32.0	38.0	**66.0**	54.0	56.0
CanToDrawerClose	13.0	62.0	64.0	74.0	**90.0**	70.0
CuttingboardToBasket	58.0	40.0	44.0	70.0	**88.0**	84.0
CuttingboardToCardboardbox	46.5	46.0	54.0	58.0	46.0	**84.0**
CuttingboardToPan	68.5	60.0	80.0	76.0	**92.0**	**92.0**
CuttingboardToPot	65.0	40.0	64.0	66.0	80.0	**84.0**
CuttingboardToTieredbasket	46.5	44.0	46.0	38.0	36.0	**72.0**
PlacematToBasket	58.5	44.0	48.0	52.0	76.0	**86.0**
PlacematToBowl	57.5	52.0	58.0	66.0	52.0	**72.0**
PlacematToPlate	63.0	50.0	74.0	60.0	38.0	**80.0**
PlacematToTieredshelf	28.5	28.0	26.0	26.0	14.0	**44.0**
PlateToBowl	57.0	52.0	50.0	54.0	48.0	**68.0**
PlateToCardboardbox	43.5	40.0	56.0	48.0	38.0	**70.0**
PlateToPan	51.0	36.0	**70.0**	66.0	46.0	**70.0**
PlateToPlate	78.7	48.0	76.0	64.0	88.0	**98.0**
TrayToCardboardbox	51.5	34.0	52.0	54.0	**82.0**	78.0
TrayToPlate	71.0	64.0	64.0	68.0	88.0	**90.0**
TrayToPot	64.5	44.0	70.0	64.0	88.0	**98.0**
TrayToTieredbasket	57.0	50.0	60.0	60.0	62.0	**74.0**
TrayToTieredshelf	31.5	28.0	28.0	38.0	24.0	**50.0**
Average	47.6	43.9	55.0	58.3	63.2	**72.8**

Table 10 Full evaluation results on the RoboTwin 2.0 benchmark. Success rates are reported in percentage. Easy denotes the clean setting and Hard denotes the randomized setting. 100 trials per task.

Simulation Task	π_0		$\pi_{0.5}$		Motus		LingBot-VLA		ABot-M0		JoyAI-RA		ACE-EGO-0	
	Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard
Adjust Bottle	99	95	100	99	89	93	100	100	–	–	100	100	100	100
Beat Block Hammer	79	84	96	93	95	88	92	89	–	–	95	91	98	92
Blocks Ranking RGB	80	63	92	85	99	97	92	91	90	79	94	93	98	97
Blocks Ranking Size	14	5	49	26	75	63	76	70	–	–	81	75	89	91
Click Alarmclock	77	68	98	89	100	100	97	43	–	–	64	56	52	38
Click Bell	71	48	99	66	100	100	43	36	–	–	81	70	66	71
Dump Bin Bigbin	88	83	92	97	95	91	97	97	–	–	97	99	100	97
Grab Roller	98	94	100	100	100	100	100	100	–	–	100	100	100	100
Handover Block	47	31	66	57	86	73	83	95	72	69	99	93	96	85
Handover Mic	97	97	98	97	78	63	94	99	–	–	100	99	91	94
Hanging Mug	14	11	18	17	38	38	34	53	–	–	31	28	29	31
Lift Pot	80	72	96	85	96	99	100	100	–	–	100	99	100	100
Move Can Pot	68	48	51	55	34	74	89	87	–	–	97	87	100	98
Move Pillbottle Pad	67	46	84	61	93	96	92	90	94	86	98	99	100	100
Move Playingcard Away	74	65	96	84	100	96	98	100	–	–	99	95	100	98
Move Stapler Pad	41	24	56	42	83	85	74	48	57	61	93	96	90	89
Open Laptop	71	81	90	96	95	91	98	96	–	–	96	100	100	98
Open Microwave	4	32	34	77	95	91	91	92	88	84	97	99	91	85
Pick Diverse Bottles	69	31	81	71	90	91	88	85	71	65	85	90	84	86
Pick Dual Bottles	59	37	93	63	96	90	99	90	70	61	95	93	89	88
Place A2B Left	43	47	87	82	88	79	89	85	–	–	99	96	95	96
Place A2B Right	39	34	87	84	91	87	80	80	–	–	97	92	90	94
Place Bread Basket	62	46	77	64	91	94	95	93	89	86	88	91	92	93
Place Bread Skillet	66	49	85	66	86	83	90	92	–	–	92	89	94	89
Place Burger Fries	81	76	94	87	98	98	98	94	–	–	99	93	98	100
Place Can Basket	55	46	62	62	81	76	75	72	72	63	71	73	78	82
Place Cans Plasticbox	63	45	94	84	98	94	100	98	–	–	100	98	100	98
Place Container Plate	97	92	99	95	98	99	99	100	–	–	96	99	98	100
Place Dual Shoes	59	51	75	75	93	87	87	86	80	80	90	97	95	96
Place Empty Cup	91	85	100	99	99	98	100	100	–	–	100	100	100	100
Place Fan	66	71	87	85	91	87	92	87	97	95	91	92	94	93
Place Mouse Pad	20	20	60	39	66	68	86	79	–	–	89	82	96	95
Place Object Basket	67	70	80	76	81	87	90	88	91	88	90	88	93	89
Place Object Scale	57	52	86	80	88	85	90	88	–	–	90	87	95	92
Place Object Stand	82	68	91	85	98	97	93	88	90	91	95	93	95	94
Place Phone Stand	49	53	81	81	87	86	90	87	–	–	95	95	91	98
Place Shoe	76	76	92	93	99	97	99	99	–	–	99	100	100	100
Press Stapler	44	37	87	83	93	98	86	93	–	–	87	81	98	98
Put Bottles Dustbin	65	56	84	79	81	79	92	93	80	89	95	97	94	93
Put Object Cabinet	73	60	80	79	88	71	85	88	–	–	87	86	82	79
Rotate QRcode	74	70	89	87	89	73	86	82	–	–	83	82	94	95
Scan Object	55	42	72	65	67	66	92	96	85	86	98	96	95	97
Shake Bottle Horizontally	98	92	99	99	100	98	99	98	–	–	100	100	100	100
Shake Bottle	94	91	99	97	100	97	100	99	–	–	100	100	100	100
Stack Blocks Three	72	52	91	76	91	95	96	95	84	77	60	62	87	82
Stack Blocks Two	93	79	97	100	100	98	100	99	96	98	95	93	100	100
Stack Bowls Three	77	75	77	71	79	87	71	77	80	86	80	81	80	85
Stack Bowls Two	94	95	95	96	98	98	90	97	–	–	95	93	96	98
Stamp Seal	46	33	79	55	93	92	74	77	72	75	90	90	94	100
Turn Switch	41	42	62	54	84	78	67	63	55	66	71	76	59	57
Average (%)	65.92	58.40	82.74	76.76	88.66	87.02	88.56	86.68	86.06	85.08	90.48	89.28	**91.12**	**90.62**