AtomEgo: Exploring Ego–Robot Integration for Embodied Foundation Model Pretraining

Di Wu[1,4] Dongchen Zheng[1,3] Junhe Sheng[1] Zhongxing Wei[1] Songxin Zhang[2] Zejian Xie[2] Xiaoquan Sun[5] Junyang Zheng[1] Zhuoyang Song[2] Jiaxing Zhang[2] Jiayu Chen[1,3]

[1]INFIFORCE [2]Lionrock Artificial Intelligence Laboratory
[3]HKU [4]Tongji University [5]Huazhong University of Science and Technology

Abstract

Embodied foundation models are constrained by the limited scale and diversity of robot demonstrations, motivating the use of large-scale egocentric human interaction data. However, how to effectively incorporate such data into embodied-model pre-training remains unclear because of substantial embodiment and action-space gaps between humans and robots. We present **AtomEgo**, a systematic study of ego–robot co-training supported by a curated corpus of approximately 2,659 hours and a scalable data processing and quality-control pipeline. Across vision–language–action and world–action model architectures, we investigate three representative paradigms: joint co-training with domain-specific action heads, progressive ego-to-robot transfer through embodiment alignment, and joint video–action modeling. We evaluate these paradigms through closed-loop real-robot experiments and language-conditioned cross-embodiment representation analysis. Our results show that egocentric data can improve policy generalization, but effective transfer requires structured alignment rather than merely mixing data. These findings provide practical guidance for leveraging scalable human experience in embodied foundation-model pre-training.

Code: `https://github.com/Agentic-Intelligence-Lab/Atom-0`
Correspondence: Jiayu Chen (jiayuc@hku.hk)
Sep, 2026

1 Introduction

Recent vision–language–action (VLA) models combine pretrained vision–language representations with action generation, enabling instruction-conditioned manipulation and transfer across tasks and embodiments [9, 13]. World–action models (WAMs) [6, 36]further learn predictive physical dynamics from visual observations, offering a complementary route toward general-purpose embodied intelligence. These developments suggest that embodied foundation models may benefit from the same scaling principles that have driven progress in language and multimodal modeling. However, scaling these models requires large and diverse datasets, while robot demonstrations are costly to collect, constrained by specific hardware, and limited in task, environment, object, and interaction diversity [18, 34]. Consequently, data has become a central bottleneck to scaling embodied foundation models.

Egocentric human interaction data provide a compelling source of scalable supervision. First-person videos capture rich object interactions in diverse, naturally occurring environments and can be collected with substantially less hardware and operational overhead than robot trajectories [17]. Yet their scale does not translate automatically into useful policy supervision. Most egocentric videos lack executable action labels, while inferred hand motions differ from robot commands in morphology, kinematics, coordinate frames, and visual appearance. Existing studies have demonstrated several promising mechanisms for bridging this gap, including action retargeting, embodiment alignment, representation sharing, and latent-action or world-model objectives [10, 25, 26, 37]. Nevertheless, these methods have largely been studied in isolation, and many operate in task-specific adaptation rather than large-scale pre-training. This naturally raises the following question:

How can egocentric data be incorporated simply and effectively into ego–robot co-training during pre-training to improve model performance?

To answer this question, we present **AtomEgo**, a systematic study of ego-robot co-training for embodied foundation models. Rather than committing to a single mechanism for exploiting human data, we study both VLA and WAM architectures and investigate three complementary pre-training paradigms: joint co-training with domain-specific action heads, progressive ego-to-robot transfer through embodiment alignment, and joint video–action modeling for cross-embodiment transfer. Together, these paradigms test whether simply isolating domain-specific supervision is sufficient to bridge the human–robot gap, or whether effective transfer requires a stronger mechanism through either explicit embodiment alignment or the unified visual-prediction objective of a WAM. Studying them within a common framework allows us to determine both the value of egocentric data and how the form of alignment shapes that value.

To support this study, we curate an ego-robot corpus at the scale of 3,033 hours and develop a scalable data processing and curation pipeline. We then conduct systematic pre-training experiments with matched data and computational settings, integrating each ego-data paradigm into a unified embodied foundation model. Finally, we build an evaluation pipeline that measures policy performance and generalization across tasks and conditions, enabling objective comparison among the three paradigms and yielding practical guidance for future ego-robot pre-training.

Our main contributions are summarized as follows:

- We construct a curated ego-robot pre-training corpus comprising approximately 3,033 hours of heterogeneous interaction data, together with a scalable processing and quality-control pipeline.
- We formulate and implement three representative paradigms for incorporating egocentric data into embodied foundation-model pre-training, spanning domain isolation, explicit embodiment alignment, and joint video–action modeling.
- We conduct controlled pre-training and comprehensive evaluation under a unified data and experimental framework, enabling a systematic comparison of alternative ego-robot co-training strategies.
- We derive empirical takeaways on how egocentric data should be aligned and utilized, providing practical guidance for scaling future embodied foundation models beyond robot-only supervision.

2 Related Work

2.1 Embodied Foundation Models

Vision–language–action (VLA) models extend pretrained vision–language models with action generation, enabling instruction-conditioned policies trained on large, heterogeneous robot corpora [4, 5, 8, 13, 24]. Recent work advances cross-embodiment transfer, long-horizon execution, temporal memory, and tactile grounding [7, 9, 14, 16, 21, 22, 29]. In parallel, world–action models (WAMs) jointly predict actions and future visual states [6], using video dynamics to complement VLA semantic priors [1, 3, 23, 33, 36]. However, embodied foundation models remain limited by scarce, costly robot demonstrations [18, 34, 38], and naively mixing incompatible embodiments and action spaces can cause negative transfer [25, 35]. We therefore study how to align large-scale egocentric experience with robot data for effective embodied foundation-model pre-training.

2.2 Leveraging Egocentric Data for Embodied Pre-Training

Egocentric data are video recordings collected from a human wearer's viewpoint that capture first-person human interactions at scale [17, 30], but their human-centric motions are not directly executable by robots [10, 20]. Prior work bridges human observations and motions to robot representations and control, demonstrating the value of egocentric data across VLA and WAM formulations [10, 12, 15, 16, 31, 32, 37]. However, how to jointly use ego and robot data during VLA/WAM pre-training stage remains a challenge. For example, Ego2Robot [26] converts human videos into robot-format visual–action trajectories, whereas JoyAI-RA 0.5 [25] aligns heterogeneous data through implicit latent actions and explicit canonical actions. However, existing studies primarily validate individual design choices, rather than systematically comparing different ego–robot pre-training strategies under controlled settings [15, 25, 26]. Our work systematically examines multiple co-training paradigms within a unified framework.

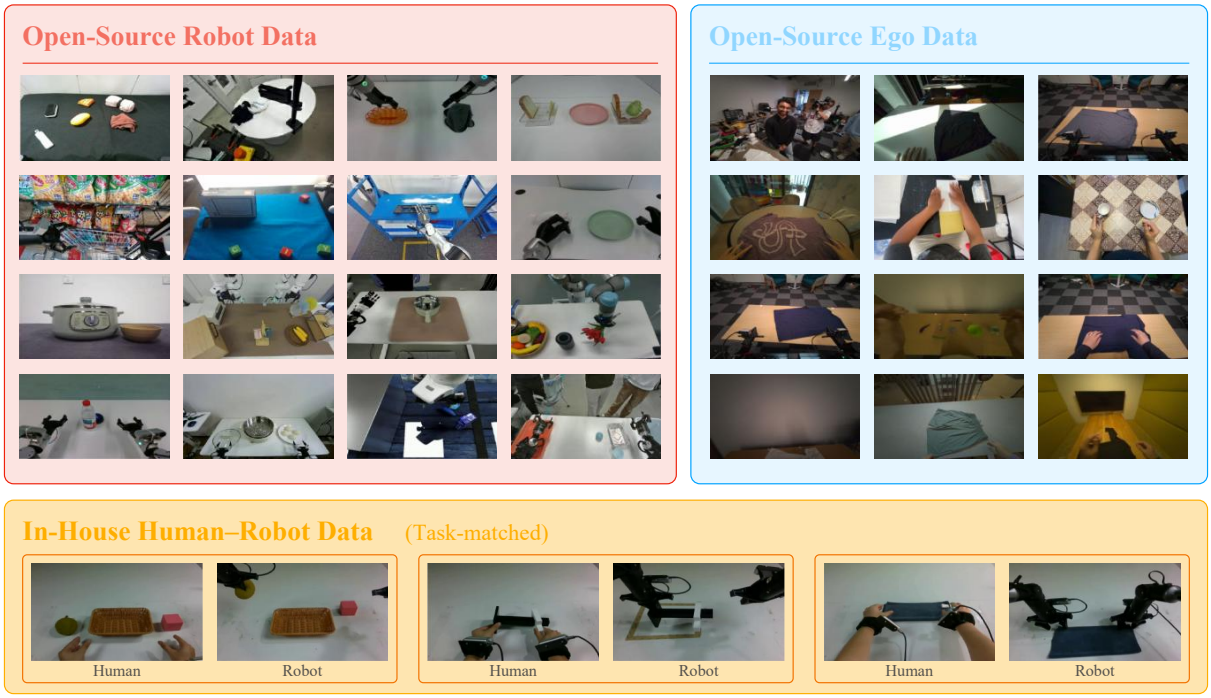

Figure 1: Qualitative overview of the three data pools used in AtomEgo pre-training: open-source robot data, open-source egocentric data, and in-house task-matched human–robot alignment data.

3 Data Preprocessing and Curation

This section describes our data recipe, data processing, and the quality-control procedure. Starting from approximately 3,033 hours of robot and egocentric data, our quality filtering pipeline retains about 2,659 hours of trajectories.

3.1 Data Recipe

Our final expanded co-training recipe contains 334,054 training episodes, corresponding to approximately 2,659 hours. We organize the corpus into three groups according to their roles in ego-robot co-training; Table 1 reports the contribution of each source.

Robot Data. We combine open-source robot demonstrations from AgiBot World Beta [2], DROID [11], RoboCOIN [28], and RoboMIND [27]. We also collect in-house demonstrations using the Piper bimanual robot. Together, these datasets span single-arm, bimanual, mobile, and humanoid robots, providing broad cross-embodiment action supervision over approximately 1,560 hours.

Ego Data. We use EgoVerse [20] as the source of approximately 1,079.5 hours of large-scale egocentric interaction data. Its first-person human and robot trajectories provide diverse manipulation experience beyond the environments and tasks covered by the robot datasets.

Alignment Data. Beyond the in-house robot demonstrations, we collect egocentric human demonstrations using wearable devices, and record robot executions of the same task set. We refer to these human and robot trajectories as alignment data: although they differ in embodiment and action space, they share task semantics and interaction objectives, providing explicit supervision for bridging the human–robot embodiment gap. This collection contains 1,296 episodes, corresponding to approximately 20 hours.

3.2 Data quality control

Inspired by Qwen-Manip [35], we apply three complementary state–action checks to identify corrupted or inconsistent episodes before training. Applying this quality-control procedure to the original 3,033 hours of data yields the final 2,659-hour training recipe.

Table 1: Composition of the expanded co-training recipe. Proportions are calculated by training episode count.

Group	Source	Episodes	Hours	Proportion
Robot data	Piper (in-house)	5,829	25.5	1.745%
	AgiBot World Beta [2]	21,837	314.0	6.537%
	DROID [11]	64,124	343.0	19.196%
	RoboCOIN [28]	93,352	655.7	27.945%
	RoboMIND [27]	83,152	221.7	24.892%
Ego data	EgoVerse [20]	64,464	1,079.5	19.297%
Alignment data	In-house aligned ego–robot data	1,296	∼20.0	0.388%
Total		334,054	∼2,659.5	100%

Table 2: Semantic layout of the unified 80-dimensional state–action space. Indices are zero-based and end-exclusive.

Range	Width	Physical meaning	Action semantics
0:7	7	Left arm joints	Relative
7:13	6	Left end-effector position and Euler rotation	Absolute
13:16	3	Reserved	Masked
16	1	Left gripper	Absolute
17:29	12	Left hand joints	Absolute
29:36	7	Right arm joints	Relative
36:42	6	Right end-effector position and Euler rotation	Absolute
42:45	3	Reserved	Masked
45	1	Right gripper	Absolute
46:58	12	Right hand joints	Absolute
58:70	12	Left and right leg joints	Absolute
70:74	4	Head and waist joints	Absolute
74:75	1	Other body joint	Absolute
75:80	5	Reserved	Masked

Sudden-change detection (S1). We smooth each state and action dimension and detect abrupt changes using the residual, acceleration, and jerk relative to robust split-level thresholds. A frame is flagged only when a large residual coincides with abnormal acceleration or jerk, reducing false positives from normal rapid motion.

State–action trend alignment (S2). For physically comparable state–action dimensions, we measure temporal lag through cross-correlation and evaluate whether their motion directions agree. This check identifies mismatched trajectories, timing offsets, dropped actions, and cases in which the observed state incorrectly leads the command.

Extreme-value filtering (S3). We estimate per-dimension 1st and 99th percentiles over each dataset and flag values outside an expanded quantile interval, as well as any NaN or infinite values. This removes severe long-tail outliers that could distort normalization and destabilize training.

3.3 Data processing

Format unification. We convert each filtered source dataset into a validated episode-level RLDS representation. Distinct embodiment, state-action space, and camera configurations are retained as separate builders, producing 45 builders in total. At loading time, all builders are mapped to a common model interface, with unavailable camera views replaced by masked placeholders.

Unified state–action space. The shared 80-dimensional space is organized by physical semantics rather than by simply padding each native vector. The principal slots are listed in Table 2. Each dataset defines a mapping from its native state and action fields to these slots. Dimensions not used by a sample are set to zero and excluded from training through the action mask. Single-arm embodiments are consistently mapped to the right arm and gripper slots.

For egocentric demonstrations, we use a 12-dimensional bimanual action representation in which each hand is encoded as an absolute end-effector pose comprising 3D position and Euler angles.

For arm slots configured as relative joint actions, the target at horizon step h is computed against the current observation state,

$$\Delta a_{t,h,j} = a_{t+h,j} - s_{t,j}. \tag{1}$$

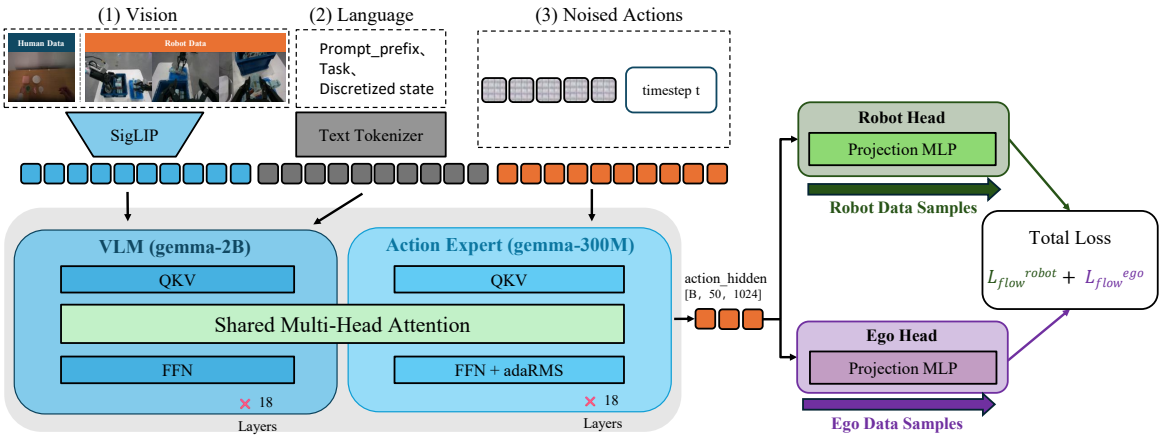

Figure 2: Overview of joint ego–robot co-training with domain-specific action heads (Atom-DH). See Section 4.2 for details.

Other valid slots, including gripper commands, dexterous-hand joints, body joints, and end-effector poses, retain their absolute semantics. For every observation at time t, we construct a 50-step target action chunk.

Dataset-specific normalization. As embodiments differ in physical ranges and control units, we compute statistics independently for each dataset after applying the same dimension mapping, 50-step chunk construction, and relative-action conversion. For every valid state and action dimension, we estimate the 1st and 99th percentiles and apply

$$\tilde{x} = 2\frac{x - q_{0.01}}{q_{0.99} - q_{0.01} + 10^{-6}} - 1, \tag{2}$$

During training, these dataset-specific statistics place heterogeneous state and action values on a common numerical scale, enabling stable joint optimization across embodiments.

4 Exploring Ego-Robot Co-Training Paradigms

Rather than committing to a single recipe for leveraging egocentric data, we investigate three complementary co-training paradigms. First, **direct utilization** (Section 4.2) incorporates ego data into joint training while separate action heads isolate domain-specific control outputs. Second, **explicit alignment** (Section 4.3) uses a progressive curriculum based on aligned demonstrations to transfer egocentric knowledge to robot control. Third, **world modeling** (Section 4.4) exploits future-state prediction, an objective naturally suited to the visual interaction structure of egocentric data. Evaluating these paradigms under a shared framework enables a systematic comparison of alternative mechanisms for ego-data pre-training.

4.1 Main Architecture

AtomEgo investigates ego–robot co-training across both vision–language–action (VLA; Sections 4.2–4.3) and world–action model (WAM; Section 4.4) architectures.

Model instantiation. The two VLA paradigms adopt the $\pi_{0.5}$ policy [8] architecture and are initialized from the base PaliGemma VLM weights rather than a pretrained $\pi_{0.5}$ policy checkpoint. The WAM paradigm instead retains the Fast-WAM architecture, as detailed in Section 4.4. Although their backbones and learning objectives differ, all paradigms use the same processed ego–robot corpus and standardized control interface.

Model inputs and outputs. Let $\mathcal{O}_t$ denote the current visual observation, organized into one base-view and two wrist-view slots; let ℓ denote the task instruction, μ the action-representation metadata, and s_t the unified 80-dimensional robot state. At the policy interface, the two model families are expressed as

$$\hat{A}_{\mathrm{VLA}} = f_{\mathrm{VLA}}(\mathcal{O}_t, \ell, \mu, s_t), \qquad \hat{A}_{\mathrm{WAM}} = f_{\mathrm{WAM}}(\mathcal{O}_t, \ell), \tag{3}$$

Both outputs are normalized action chunks in $\mathbb{R}^{B\times 50\times 80}$, representing 50 future control steps in the action space.

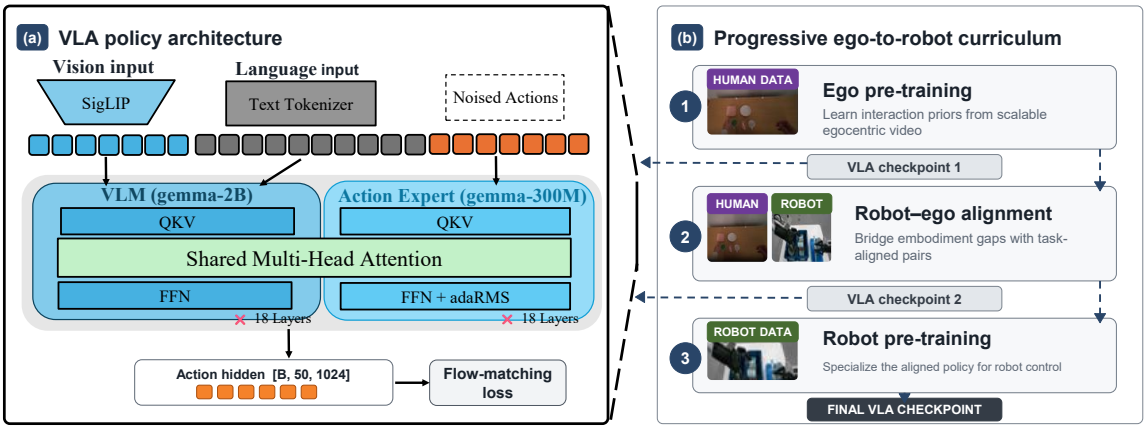

Figure 3: Overview of progressive ego-to-robot transfer through curriculum-based embodiment alignment (Atom-CL). See Section 4.3 for details.

Table 3: Three-stage pre-training curriculum for progressive ego-to-robot transfer.

Stage	Objective and training data	Initialization	Updates	Trainable components
1	Egocentric prior learning on Ego data	Base PaliGemma VLM	100,000	Full model
2	Embodiment alignment on task-aligned human and robot data	Stage 1 checkpoint	50,000	Vision encoder and action expert
3	Robot pre-training on robot datasets	Stage 2 checkpoint	97,728	Full model

4.2 Joint Co-Training with Domain-Specific Action Heads

Egocentric and robot trajectories provide complementary interaction experience, but their control targets follow distinct embodiment- and domain-specific distributions. A fully shared policy must therefore fit heterogeneous action mappings through the same output projection, which can introduce conflicting supervision at the control interface. We address this issue with a minimal architectural modification: the perception, language, and action representations are shared across domains, while the final action projection is separated into ego-specific and robot-specific heads.

Let $h_i \in \mathbb{R}^{50\times D}$ denote the action representation produced by the shared backbone for sample i, and let $d_i \in \{\text{ego}, \text{robot}\}$ denote its domain. Two independently parameterized projections decode the shared representation,

$$\hat{v}_i = \begin{cases} W_{\text{ego}} h_i, & d_i = \text{ego}, \\ W_{\text{robot}} h_i, & d_i = \text{robot}. \end{cases} \tag{4}$$

Each sample supervises only its corresponding head, while gradients from both domains continue to update the shared backbone. The ego head is used only to provide auxiliary supervision during co-training; downstream robot control is decoded exclusively through the robot head.

We additionally evaluate an optimal-transport (OT) regularizer on the aligned human–robot subset. It derives temporal correspondences from end-effector trajectories and encourages matched action latents to remain close, while a lightweight embodiment prompt distinguishes human and robot inputs. Implementation details and experimental results for this variant are provided in Appendix C.

Overall, the dual-head model provides a minimal baseline for exploiting egocentric data through shared representation learning while retaining domain-specific action decoding. It also serves as a controlled reference for the more structured transfer paradigms introduced below.

4.3 Progressive Ego-to-Robot Transfer via Embodiment Alignment

Directly mixing egocentric and robot supervision requires the model to bridge a substantial embodiment gap within a single training stage. Prior work has shown that introducing an intermediate alignment stage can improve the utilization of egocentric data [37]; we further investigate this strategy at pre-training scale. We formulate ego-to-robot transfer as a curriculum with three successive data distributions: The model first learns general interaction priors from large-scale egocentric trajectories, then adapts on a smaller set of task-aligned human and robot demonstrations,

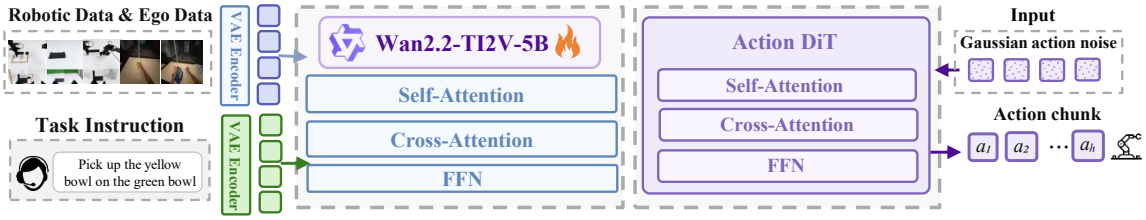

Figure 4: Overview of joint world–action modeling for ego–robot co-training (Atom-WAM). See Section 4.4 for details.

and finally trains on robot pre-training data. The intermediate alignment stage provides a gradual transition from human interaction patterns to robot behavior, enabling the three-stage curriculum to fully leverage the benefits of egocentric data.

Aligned human–robot supervision. The alignment set contains human and robot demonstrations that share task semantics and interaction objectives, while differing in morphology, appearance, and native control space. Within this stage, we convert both domains to a common end-effector representation. A tracked human hand pose is treated as a canonical end effector, while robot joint trajectories are converted to Cartesian end-effector poses using embodiment-specific forward kinematics when Cartesian poses are not directly available.

Both are expressed in their fixed egocentric camera frame and then transformed into motion relative to the current end-effector frame. Let ${}^{C}T_{E_t}$ denote the end-effector pose at time t in camera frame C. For prediction horizon τ_h, the aligned target is the relative transform from the current pose to the future pose,

$$\Delta T_{t,h} = \left({}^{C}T_{E_t}\right)^{-1} {}^{C}T_{E_{t+\tau_h}}. \tag{5}$$

This construction aligns ego and robot trajectories at the representation level, allowing the task-matched dataset to provide an explicit transition from egocentric pre-training to robot learning. This common representation is used specifically for the alignment stage; the subsequent robot-only stage retains each robot dataset's original joint-space control targets. Consequently, only the small alignment set requires additional conversion, avoiding costly reconstruction of the full pre-training corpus.

Progressive training. The pre-training curriculum consists of three sequential stages. Egocentric pre-training first acquires broad interaction priors, alignment fine-tuning then bridges human and robot motion under the shared end-effector representation, and robot pre-training finally adapts the aligned model to heterogeneous executable control. Each stage initializes from a checkpoint of the preceding stage. Table 3 summarizes this schedule.

By dedicating the second stage to the aligned subset, the curriculum uses these data as a concentrated bridge between egocentric priors and robot control, rather than diluting their supervision within the full pre-training mixture.

4.4 Joint World-Action Modeling for Cross-Embodiment Transfer

The preceding paradigms address the embodiment gap at the action interface, either by separating domain-specific outputs or by explicitly aligning human and robot motions. Our third paradigm instead exploits a signal that is naturally shared across embodiments: the visual evolution of an interaction. Although human hands and robot manipulators have different control spaces, both induce temporally structured changes in objects and scenes. We therefore instantiate this route by fine-tuning Fast-WAM [36] on a mixture of egocentric and robot trajectories, using future-video modeling as a shared channel for cross-embodiment transfer.

Fast-WAM initialization. Unlike the two VLA-based routes above, this paradigm initializes from Fast-WAM architecture. Fast-WAM couples a video Diffusion Transformer (DiT) [19] with an Action DiT under a shared-attention Mixture-of-Transformers formulation. The Video DiT models future visual latents, while the Action DiT predicts a continuous action chunk from the current observation and language instruction. During training, clean observation tokens provide a common visual anchor for both branches. A structured attention mask allows both branches to access this anchor while preventing action tokens from attending to noisy future-video latent tokens, thereby avoiding future-information leakage.

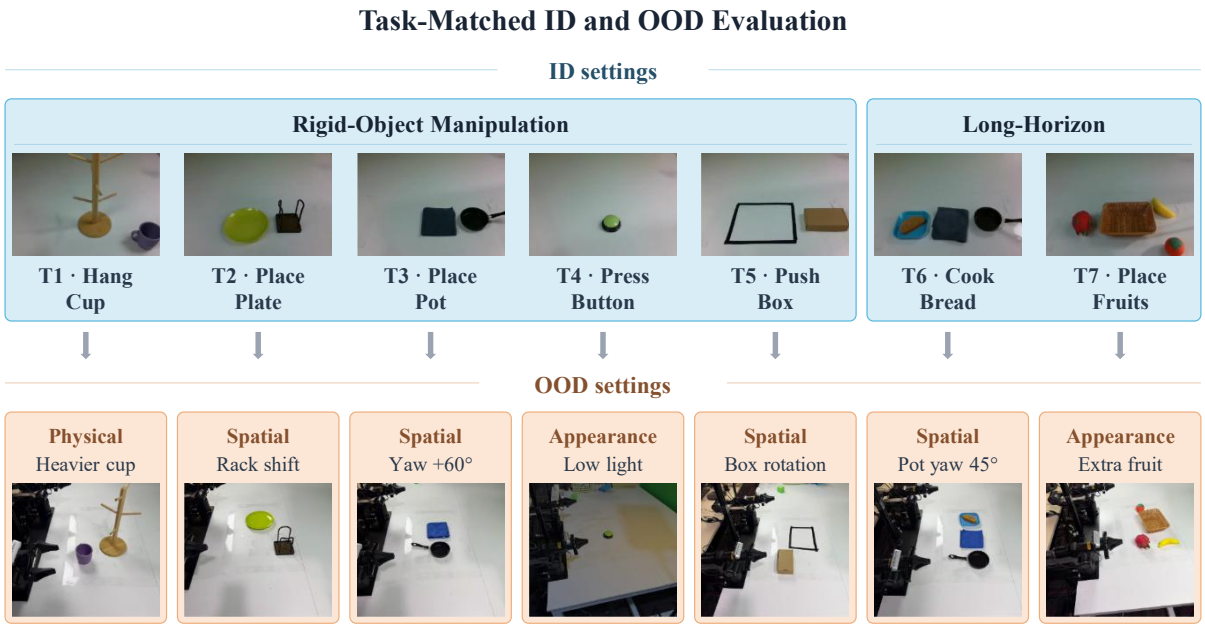

Figure 5: Task-matched ID and OOD evaluation settings for the seven real-robot tasks. Each OOD setting preserves the task objective while changing one primary spatial, appearance, or physical factor.

Joint ego–robot fine-tuning. We train the model on the curated robot and egocentric data described in Section 3. We construct the fine-tuning distribution as

$$\mathcal{D}_{\text{mix}} = \rho\mathcal{D}_{\text{ego}} + (1-\rho)\mathcal{D}_{\text{robot}}, \tag{6}$$

where ρ is the effective ego-data sampling probability. Future frames from both domains supervise the video flow-matching objective. Following Fast-WAM, we optimize

$$\mathcal{L}_{\text{joint}} = \mathbb{E}_{\mathcal{D}_{\text{mix}}}\left[\mathcal{L}_{\text{action}} + \lambda_{\text{video}}\mathcal{L}_{\text{video}}\right], \tag{7}$$

where λ_{video} balances action learning and video co-training. In this way, both domains update the world representation through future-video prediction, while action supervision is restricted to the control dimensions available for each sample. The shared visual-dynamics objective therefore provides the primary transfer channel between human and robot embodiments without requiring their native action spaces to be identical.

At deployment, we follow the Fast-WAM inference procedure and omit future-video tokens, denoising, and decoding. The Video DiT processes only the current visual context to produce the latent world representation, which then generates the robot action chunk.

4.5 Training pipeline

All three paradigms follow a common pre-training–post-training–evaluation pipeline. Our study focuses on pre-training, where the methods differ in how they incorporate egocentric and robot data. Each pre-training resulting checkpoint is subsequently post-trained on the same in-house multi-task robot dataset using an identical recipe, isolating the effect of the pre-training strategy. Pre-training uses 3×8 NVIDIA B200 GPUs and takes approximately four days. Detailed training hyperparameters are provided in Appendix B.

5 Experiments

Having introduced three paradigms for incorporating egocentric data during pre-training, we now evaluate their effectiveness and derive practical insights. Real-robot task performance provides the most direct evaluation; in Section 5.1, we evaluate a suite of manipulation tasks under both in-distribution and out-of-distribution conditions. Beyond real-robot evaluation, Section 5.2 examines the latent representations of egocentric and robot data, providing an offline measure of cross-embodiment representation learning.

Table 4: Real-robot success rates for the VLA and WAM checkpoints on seven manipulation tasks. Each ID and OOD result aggregates seven tasks with ten trials per task; Combined pools all 140 trials.

Family	Checkpoint	ID success	OOD success	Combined success
VLA	Baseline	42.86% (30/70)	22.86% (16/70)	32.86% (46/140)
	Atom-DH	44.29% (31/70)	**42.86% (30/70)**	43.57% (61/140)
	Atom-CL	**60.00% (42/70)**	**42.86% (30/70)**	**51.43% (72/140)**
WAM	Baseline	28.57% (20/70)	28.57% (20/70)	28.57% (40/140)
	Atom-WAM	20.00% (14/70)	21.43% (15/70)	20.71% (29/140)

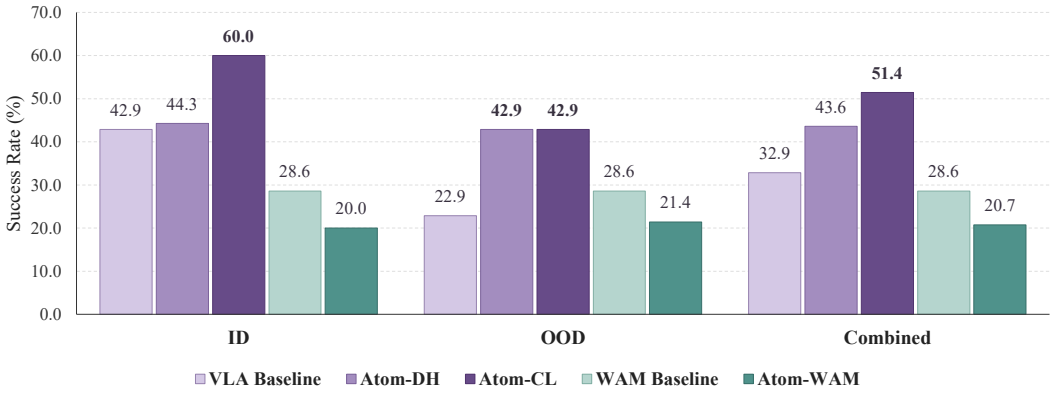

Figure 6: Real-robot success rates of the VLA and WAM checkpoints under ID, OOD, and combined evaluation settings.

5.1 Real-Robot Evaluation

Experimental setup. We evaluate seven manipulation tasks: *Cook Bread*,*Hang Cup*, *Place Fruits*, *Place Plate*, *Place Pot*, *Press Button*, and *Push Box*. For each checkpoint, we conduct ten trials per task under the nominal ID condition and ten trials under the corresponding OOD condition, yielding 70 ID and 70 OOD trials. Each OOD variant preserves the task objective while changing one primary spatial, appearance, or physical factor. A rollout is counted as successful only when it reaches the complete predefined task end state. Detailed task configurations are provided in Appendix A.1. The matched nominal and shifted task settings are summarized in Figure 5.

Baselines. To evaluate the contribution of egocentric data, each baseline is pre-trained only on robot demonstrations using the standard training objective, without ego data or any ego-specific integration mechanism, and is then post-trained on the same in-house multi-task robot dataset. In contrast, the Atom variants incorporate both robot and ego data during pre-training through their respective co-training strategies, while following the identical post-training procedure. Because the VLA and WAM routes use different model architectures, we construct a separate robot-only baseline for each family. The VLA baseline is compared with domain-specific dual-head co-training (**Atom-DH**) and progressive embodiment alignment (**Atom-CL**), while the WAM baseline is compared with joint world–action co-training (**Atom-WAM**). This setup isolates the effect of ego-data integration within each architecture family.

Results. Table 4 reports the best measured checkpoint from each route, with the results visualized in Figure 6. Detailed per-task ID and OOD success rates are provided in Appendix A.3. Within the VLA family, Atom-CL achieves the highest ID and combined success rates of 60.00% and 51.43%, improving over the VLA baseline by 17.14 and 18.57 percentage points, respectively. Atom-DH reaches 43.57% combined success, and both methods improve OOD success from 22.86% to 42.86%. In contrast, Atom-WAM surprisingly underperforms its WAM baseline across both conditions: ID success decreases from 28.57% to 20.00%, OOD success decreases from 28.57% to 21.43%, and combined success decreases from 28.57% to 20.71%. This counterintuitive result suggests that visual prediction alone may capture domain-specific dynamics without sufficiently grounding them in executable robot actions. Together, the results indicate that egocentric data are not inherently beneficial and that their value depends strongly on the integration strategy, with progressive embodiment alignment providing the most consistent gains under this evaluation.

Table 5: Language-conditioned cross-embodiment representation results on seven tasks. Higher raw matched-task similarity indicates stronger consistency, while lower normalized different-task similarity indicates better task separation.

Checkpoint	Raw matched mean ↑	Normalized different-task mean ↓
Atom-DH	**0.999625**	0.592719
Atom-CL	0.994481	**0.387240**
Atom-WAM	0.997357	0.480727

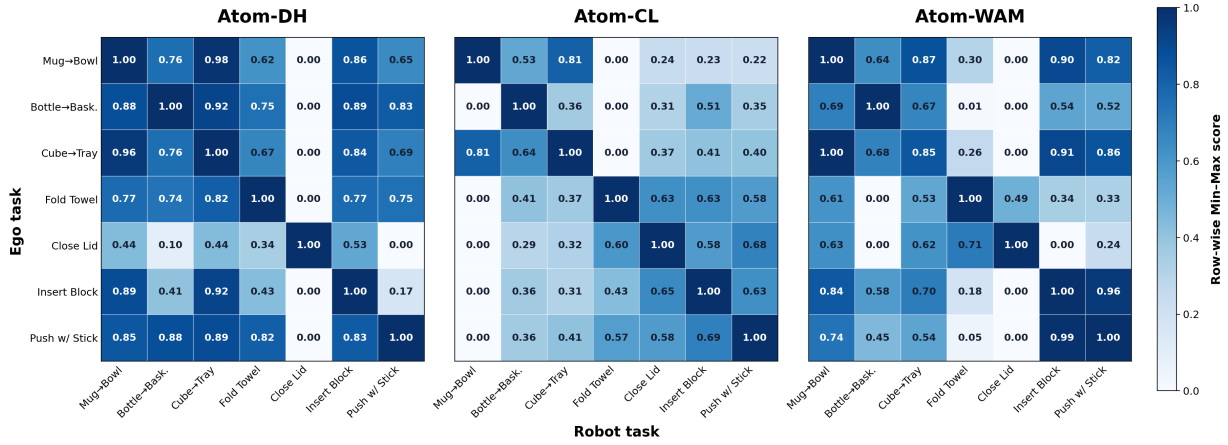

Figure 7: Row-normalized language-conditioned Ego-to-Robot task similarity for Atom-DH (top left), Atom-CL (top right), and Atom-WAM (bottom). Rows denote Ego tasks and columns denote Robot tasks.

5.2 Language-Conditioned Cross-Embodiment Representation Evaluation

To further examine cross-embodiment representation learning, we evaluate whether ego and robot videos of the same task are encoded similarly while remaining distinguishable from other tasks. We use seven task-matched sets from the alignment dataset rather than the real-robot evaluation suite, ensuring that the ego and robot videos share the same task semantics.

Protocol and metrics. We evaluate three post-trained models—Atom-DH, Atom-CL, and Atom-WAM—on 62 ego and 62 robot videos. For each video, we uniformly sample 32 front-view frames, pair them with the task instruction, and aggregate the resulting language-conditioned frame features into a clip representation z using temporal-change weighting. For ego task p and robot task q, we compute

$$S_{pq} = \frac{1}{|\mathcal{E}_p||\mathcal{R}_q|} \sum_{i \in \mathcal{E}_p} \sum_{j \in \mathcal{R}_q} \cos(z_i, z_j), \tag{8}$$

where $\mathcal{E}_p$ and $\mathcal{R}_q$ denote the corresponding ego and robot video sets. The resulting 7×7 matrix is visualized in Figure 7 after row-wise Min–Max normalization. Table 5 reports the mean raw diagonal similarity for matched tasks and the mean normalized off-diagonal similarity for different tasks. Together, these metrics measure cross-embodiment task consistency and task discrimination.

Results. Atom-DH achieves the highest matched-task similarity but also the highest normalized different-task similarity, indicating strong cross-embodiment consistency alongside relatively weak task separation. Atom-CL exhibits slightly lower matched-task similarity but substantially better task separation, while Atom-WAM lies between the two VLA methods on both metrics. This intermediate position suggests that joint visual prediction captures some shared cross-embodiment structure but does not produce the strongest task selectivity. More broadly, the representation ordering does not directly match the real-robot performance ordering. Useful cross-embodiment alignment must therefore preserve action-relevant task structure rather than simply bringing ego and robot representations closer.

5.3 Result Summary and Takeaways

We now return to our initial question: How can we use egocentric data effectively during pre-training? Our experiments yield the following takeaways.

Takeaway 1: Effective alignment requires more than direct data mixing. Both real-robot and representation results show that simply bringing ego and robot features closer does not guarantee useful transfer. Atom-DH achieves high matched-task similarity but weak task separation, whereas progressive embodiment alignment yields the best control performance. Effective alignment must therefore preserve task-discriminative structure and connect human experience to executable robot actions.

Takeaway 2: Ego data particularly benefit OOD generalization. Both ego-pretrained VLA variants improve OOD success from 22.86% to 42.86%, suggesting that diverse human interactions provide transferable priors beyond the robot training distribution. However, Atom-CL achieves substantially higher ID performance than Atom-DH, indicating that structured alignment is necessary to convert this diversity into precise robot control.

Takeaway 3: WAM-based transfer requires stronger alignment mechanisms. Direct ego–robot co-training with Fast-WAM [36] reduces combined success from 28.57% to 20.71%. Future-video prediction can capture embodiment-specific appearance and motion without learning action-relevant shared structure. Thus, effectively exploiting ego data in WAMs may require action-conditioned world objectives, domain-balanced optimization, or progressive alignment.

6 Limitation and future work

While this study provides a controlled comparison of representative ego–robot co-training paradigms, several directions remain open. First, due to resource constraints, the scale of our pre-training corpus remains limited. Second, our scope prioritizes a systematic evaluation of alternative paradigms rather than specialized algorithmic development for any single route. In the future, we plan to further investigate the most effective approach and scale its training to substantially larger datasets toward a more capable embodied foundation model.

7 Conclusion

We presented AtomEgo, a unified framework for studying scalable ego–robot co-training for embodied foundation models. Using a curated corpus of robot and egocentric interaction data, we compared three complementary paradigms: direct co-training with domain-specific action heads, progressive embodiment alignment, and joint video–action modeling. Our results show that egocentric data can improve policy generalization, but their effectiveness depends critically on the cross-embodiment alignment strategy. These findings provide a practical foundation for scaling embodied models beyond robot-only supervision.

Acknowledgements

We gratefully acknowledge the computational support provided by the Lionrock Artificial Intelligence Laboratory at the China Merchants Research Institute of Advanced Technology.

Author Contributions

Data collection and processing: Di Wu.

Algorithm and training: Junhe Sheng, Zhongxing Wei, Xiaoquan Sun, Junyang Zheng, Di Wu, Songxin Zhang and Zejian Xie.

Evaluation: Dongchen Zheng.

Writing: Di Wu and Dongchen Zheng.

Advisor: Jiayu Chen, Jiaxing Zhang, Zhuoyang Song.

References

[1] Niket Agarwal, Arslan Ali, Jon Allen, Martin Antolini, Adeline Aubame, Alisson Azzolini, Junjie Bai, Maciej Bala, Yogesh Balaji, Josh Bapst, et al. Cosmos 3: Omnimodal world models for physical ai. *arXiv preprint arXiv:2606.02800*, 2026.

[2] AgiBot-World-Contributors, Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.

[3] Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 35101–35113, 2026.

[4] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[5] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

[6] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

[7] Jialei Huang, Shuo Wang, Fanqi Lin, Yihang Hu, Chuan Wen, and Yang Gao. Tactile-VLA: Unlocking vision-language-action model's physical knowledge for tactile generalization. *arXiv preprint arXiv:2507.09160*, 2025.

[8] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. pi05: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

[9] Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, et al. pi0.7: a steerable generalist robotic foundation model with emergent capabilities. *arXiv preprint arXiv:2604.15483*, 2026.

[10] Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. Egomimic: Scaling imitation learning via egocentric video. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13226–13233. IEEE, 2025.

[11] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

[12] Ji Woong Kim, Ke Wang, Zipeng Fu, Sirui Chen, Jeff Lai, Chelsea Finn, et al. Ego-pi: Vla fine-tuning for ego-centric human and robot data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1515–1524, 2026.

[13] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[14] Yantao Liu, Zijun Yao, Rui Min, Yixin Cao, Lei Hou, and Juanzi Li. Rm-bench: Benchmarking reward models of language models with subtlety and style. In *International Conference on Learning Representations*, volume 2025, pages 44323–44355, 2025.

[15] Hao Luo, Yicheng Feng, Wanpeng Zhang, Sipeng Zheng, Ye Wang, Haoqi Yuan, Jiazheng Liu, Chaoyi Xu, Qin Jin, and Zongqing Lu. Being-h0: vision-language-action pretraining from large-scale human videos. *arXiv preprint arXiv:2507.15597*, 2025.

[16] Hao Luo, Wanpeng Zhang, Yicheng Feng, Sipeng Zheng, Haiweng Xu, Chaoyi Xu, Ziheng Xi, Yuhui Fu, and Zongqing Lu. Being-h0. 7: A latent world-action model from egocentric videos. *arXiv preprint arXiv:2605.00078*, 2026.

[17] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

[18] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya 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.

[19] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4172–4182. IEEE, 2023.

[20] Ryan Punamiya, Simar Kareer, Zeyi Liu, Josh Citron, Ri-Zhao Qiu, Xiongyi Cai, Alexey Gavryushin, Jiaqi Chen, Davide Liconti, Lawrence Y Zhu, et al. Egoverse: An egocentric human dataset for robot learning from around the world. *arXiv preprint arXiv:2604.07607*, 2026.

[21] Wenxuan Song, Jiayi Chen, Xiaoquan Sun, Huashuo Lei, Yikai Qin, Wei Zhao, Pengxiang Ding, Han Zhao, Tongxin Wang, Pengxu Hou, et al. Rethinking the practicality of vision-language-action model: A comprehensive benchmark and an improved baseline. *arXiv preprint arXiv:2602.22663*, 2026.

[22] Xiaoquan Sun, Zetian Xu, Chen Cao, Zonghe Liu, Yihan Sun, Jingrui Pang, Ruijian Zhang, Zhen Yang, Kang Pang, Dingxin He, et al. Atomvla: Scalable post-training for robotic manipulation via predictive latent world models. *arXiv preprint arXiv:2603.08519*, 2026.

[23] Xiaoquan Sun, Ruijian Zhang, Chen Cao, Yihan Sun, Jiahui Chen, Zetian Xu, Bo Chen, Haijier Chen, Zhen Yang, Jiarun Zhu, et al. Himem-wam: Hierarchical memory-gated world action models for robotic manipulation. *arXiv preprint arXiv:2606.10363*, 2026.

[24] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[25] RA Team et al. Joyai-ra 0.5: Scaling robot manipulation learning via dual action alignment. *arXiv preprint arXiv:2608.05674*, 2026.

[26] Ye Wang, Pei Lin, Xiong-Hui Chen, Haoqi Yuan, Zhixuan Liang, Yiyang Huang, Anzhe Chen, Zixing Lei, Jie Zhang, Tao Zhang, et al. Ego2robot: Scalable robot data synthesis from egocentric human data. *arXiv preprint arXiv:2608.02580*, 2026.

[27] Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, et al. Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation. *arXiv preprint arXiv:2412.13877*, 2024.

[28] Shihan Wu, Xuecheng Liu, Shaoxuan Xie, Pengwei Wang, Xinghang Li, Bowen Yang, Zhe Li, Kai Zhu, Hongyu Wu, Yiheng Liu, et al. Robocoin: An open-sourced bimanual robotic data collection for integrated manipulation. *arXiv preprint arXiv:2511.17441*, 2025.

[29] Wei Wu, Fangjing Wang, Fan Lu, He Sun, Shi Liu, Yunnan Wang, Yibin Yan, Yong Wang, Shuailei Ma, Xinyang Wang, et al. From foundation to application: Improving vla models in practice. *arXiv preprint arXiv:2607.06403*, 2026.

[30] Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022.

[31] Ruihan Yang, Qinxi Yu, Yecheng Wu, Rui Yan, Borui Li, An-Chieh Cheng, Xueyan Zou, Yunhao Fang, Xuxin Cheng, Ri-Zhao Qiu, et al. Egovla: Learning vision-language-action models from egocentric human videos. *arXiv preprint arXiv:2507.12440*, 2025.

[32] Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. In *International Conference on Learning Representations*, volume 2025, pages 28213–28239, 2025.

[33] Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

[34] Yifan Ye, Yankai Fu, Yaoxu Lv, Bohan Hou, Jun Cen, Lingdong Kong, Duo Zheng, Tianxing Chen, Jiaming Liu, Ziang Cao, et al. Data pyramid for embodied manipulation. *arXiv preprint arXiv:2607.24744*, 2026.

[35] Haoqi Yuan, Zhixuan Liang, Anzhe Chen, Ye Wang, Haoyang Li, Pei Lin, Yiyang Huang, Zixing Lei, Tong Zhang, Jiazhao Zhang, et al. Qwen-robotmanip technical report: Alignment unlocks scale for robotic manipulation foundation models. *arXiv preprint arXiv:2606.17846*, 2026.

[36] Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026.

[37] Ruijie Zheng, Dantong Niu, Yuqi Xie, Jing Wang, Mengda Xu, Yunfan Jiang, Fernando Castañeda, Fengyuan Hu, You Liang Tan, Letian Fu, et al. Egoscale: Scaling dexterous manipulation with diverse egocentric human data. *arXiv preprint arXiv:2602.16710*, 2026.

[38] Yifan Zhong, Fengshuo Bai, Shaofei Cai, Xuchuan Huang, Zhang Chen, Xiaowei Zhang, Yuanfei Wang, Shaoyang Guo, Tianrui Guan, Ka Nam Lui, et al. A survey on vision-language-action models: An action tokenization perspective. *arXiv preprint arXiv:2507.01925*, 2025.

Appendix Contents

A Evaluation Details

A.1 Real-Robot Task Definitions

The ID configurations follow the nominal setups used in our seven-task real-robot evaluation. Each paired OOD condition preserves the task objective while perturbing one primary factor, allowing the resulting performance change to be associated with a specific type of distribution shift. Task initializations, perturbation types and execution timing, maximum rollout durations, and complete-task success criteria are fixed before evaluation and shared across checkpoints. Table 6 lists the seven evaluated task configurations and success criteria.

Table 6: Seven-task real-robot evaluation protocol. Each OOD condition changes one primary factor while holding other task-relevant variables fixed whenever applicable. Symbolic quantities denote protocol parameters fixed before execution, including illumination, mass, displacement, and stability duration.

Task	ID setup	OOD type	OOD perturbation	OOD specification and controls	Success criterion
Cook Bread	Standard pot and bread poses	Spatial	Rotate pot	Pot yaw $= 45^\circ$ CCW; pot center and other object poses unchanged	Bread is placed in the pot, and the pot is placed on the cloth
Hang Cup	Standard cup mass	Physical	Increase cup mass	Mass $m_{\mathrm{ID}} \rightarrow m_{\mathrm{OOD}}$; cup appearance and initial pose unchanged	Handle is inserted and the cup hangs stably for the predefined duration
Place Fruits	Standard target set	Appearance	Add visually similar non-target fruit as a distractor	Distractor identity and pose fixed; target identities and initial poses unchanged	All three target fruits are placed in the basket while the distractor remains outside
Place Plate	Standard rack pose	Spatial	Translate rack	Rack displacement $\Delta\mathbf{x}_{\mathrm{rack}}$ fixed; plate pose and rack orientation unchanged	Plate is aligned, inserted, released, and remains stable in the rack
Place Pot	Standard pot pose	Spatial	Rotate pot	Pot yaw $= 60^\circ$; pot center and cloth pose unchanged	Pot is grasped by the handle and placed on the cloth
Press Button	Standard illumination	Appearance	Reduce illumination	Illuminance $L_{\mathrm{ID}} \rightarrow L_{\mathrm{OOD}}$; camera and button poses unchanged	Button activation is registered and the end effector reaches the predefined retreat region
Push Box	Standard box orientation	Spatial	Rotate box	Box yaw $= 90^\circ$; box center and box–target distance unchanged	Box reaches the predefined target region

Figure 8 provides qualitative task-completion progressions for the same seven-task evaluation suite.

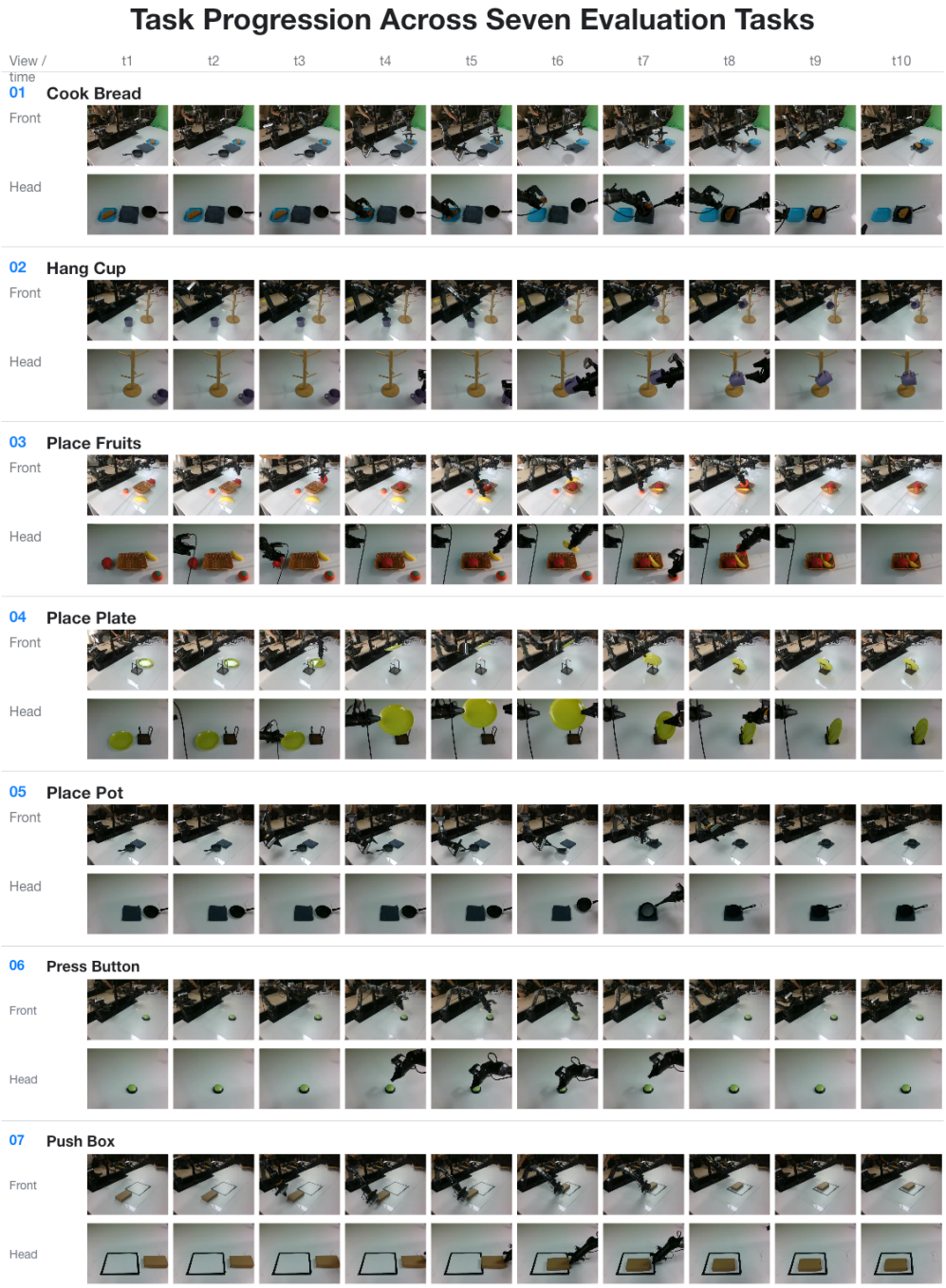

Figure 8: Uniformly sampled task-completion progressions for the seven real-robot evaluation tasks. Each row pair shows front and head views at ten time points from a complete task execution.

A.2 Cross-Embodiment Evaluation Instructions

The cross-embodiment representation evaluation uses the following seven fixed task instructions in the same order as the evaluation matrix. The Ego and Robot videos associated with the same task receive exactly the same instruction.

1. `pink_mug_to_green_bowl`: "Pick up the pink mug and place it in the green bowl."
2. `water_bottle_to_basket`: "Pick up the water bottle and place it in the basket."
3. `red_cube_into_blue_tray`: "Pick up the red cube and place it in the blue tray."
4. `fold_towel`: "Use both hands to grasp the two designated corners along the lower edge of the towel and fold the towel once from bottom to top."
5. `close_box_lid`: "Use both hands to grasp the two sides of the box lid and close the lid."
6. `insert_block`: "Pick up the wooden block and insert it into the hole of the white foam block."
7. `push_cube_with_stick`: "Pick up the wooden stick, then use it to push the red cube completely into the white target area."

A.3 Per-Task Real-Robot Success Rates

To complement the aggregate results in Table 4, Table 7 provides the task-level breakdown for all evaluated checkpoints. ID and OOD results are reported in separate columns, and each entry gives the number of successful trials out of ten. Following the main evaluation, the average row summarizes performance across all seven tasks, and the best result for each task can be highlighted in bold.

Table 7: Per-task real-robot results under ID and OOD conditions. Task rows report successful trials out of ten; the overall rows aggregate all seven tasks (70 trials per condition). Best results for each task and condition are shown in bold, including ties; all-zero rows are left unbolded.

	VLA						WAM			
	Baseline		Atom-DH		Atom-CL		Baseline		Atom-WAM	
Task	ID	OOD	ID	OOD	ID	OOD	ID	OOD	ID	OOD
Overall (7 tasks)	30/70	16/70	31/70	30/70	42/70	30/70	20/70	20/70	14/70	15/70
Success rate	42.86%	22.86%	44.29%	42.86%	60.00%	42.86%	28.57%	28.57%	20.00%	21.43%
Cook Bread	**10/10**	0/10	1/10	0/10	2/10	0/10	2/10	**1/10**	0/10	0/10
Hang Cup	1/10	0/10	0/10	0/10	**10/10**	**10/10**	0/10	0/10	0/10	0/10
Place Fruits	**10/10**	**10/10**	**10/10**	**10/10**	**10/10**	**10/10**	**10/10**	**10/10**	**10/10**	**10/10**
Place Plate	0/10	0/10	0/10	0/10	0/10	0/10	0/10	0/10	0/10	0/10
Place Pot	**4/10**	**2/10**	0/10	0/10	0/10	0/10	0/10	0/10	0/10	0/10
Press Button	0/10	0/10	**10/10**	**10/10**	**10/10**	**10/10**	8/10	9/10	4/10	5/10
Push Box	5/10	4/10	**10/10**	**10/10**	**10/10**	0/10	0/10	0/10	0/10	0/10

B Training Hyperparameters

Table 8 summarizes the key settings for robot pre-training and subsequent multi-task post-training. Pre-training uses 34 audited in-house and open-source robot datasets, while post-training adapts the model to two in-house Piper multi-task datasets. Under the reported hardware configuration, pre-training takes approximately four days, while post-training takes about 12 hours.

Table 8: Key hyperparameters for robot pre-training and multi-task post-training.

Hyperparameter	Robot pre-training	Multi-task post-training
Training data	34 robot datasets; 150.1M source frames; EgoVerse excluded	Two in-house Piper multi-task datasets
Action representation	Unified80	Unified80
Hardware	3×8 NVIDIA B200 GPUs	1×8 NVIDIA B200 GPUs
FSDP devices	4	4
Global batch size (train / validation)	1,536 / 96	512 / 96
Training steps	97,728	20,000
Warmup / decay steps	5,000 / 97,728	1,000 / 30,000
Peak / decay learning rate	1×10^{-6} / 1×10^{-7}	2.5×10^{-5} / 2.5×10^{-6}
Evaluation / checkpoint interval	1,000 / 10,000	1,000 / 5,000
Action MSE	Disabled	Enabled
Initialization	Base PaliGemma VLM	Final robot pre-training checkpoint

C Optimal-Transport Alignment Details

The domain-specific heads in Section 4.2 isolate human and robot action decoding, but they do not explicitly constrain the shared action representation across domains. We therefore examine a training-only optimal-transport (OT) regularizer that brings human and robot latents closer when their end-effector trajectories describe similar interactions.

Data pools and alignment supervision. Because data with meaningful human–robot correspondence constitute only a small fraction of the full corpus, unconstrained sampling would rarely place sufficient paired examples in the same batch. We therefore divide the training mixture into a general co-training pool and an alignment pool, and reserve a fixed batch quota for each. The alignment pool consists exclusively of our in-house aligned data: wearable-camera human demonstrations and robot executions collected for the same task set. The general pool contains all remaining egocentric and robot datasets and preserves the scale and diversity of the complete corpus. Only samples from the alignment pool receive the additional OT supervision, while samples from the general pool retain the original flow-matching objective.

Trajectory-aware OT alignment. Human and robot demonstrations of the same interaction may progress at different speeds, making direct frame-by-frame latent matching unreliable. We first align their ground-truth Cartesian trajectories using Soft Dynamic Time Warping (Soft-DTW), a smooth sequence-alignment procedure that permits nonlinear temporal correspondence. This operation is applied only to end-effector positions and is detached from gradient computation. It produces a correspondence weight matrix W that indicates which portions of the human and robot trajectories should be compared.

We construct separate bridges for single-arm and bimanual interactions. The former uses the right end-effector position, while the latter jointly uses the left and right end-effectors. For each bridge, valid human–robot pairs are collected from the global batch. We then use the Sinkhorn algorithm, an entropy-regularized solver for soft optimal transport, to match their action-expert latents with pairwise cost

$$C_{ij} = \frac{1}{2}\overline{\|h_i - r_j\|_2^2} W_{ij}, \tag{9}$$

where h_i and r_j denote the human and robot latent trajectories, and the overline averages over temporal and feature dimensions. The valid single-arm and bimanual transport costs are summed to obtain $\mathcal{L}_{\mathrm{OT}}$.

The final training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{FM}} + \alpha \mathcal{L}_{\mathrm{OT}}, \tag{10}$$

where α controls the strength of latent alignment and flow matching remains the primary action objective. OT is used only during training; the output heads and inference procedure are unchanged, so the variant introduces no deployment-time computation.

Experiment result. We compare two Atom-DH variants that differ in whether the OT regularizer is enabled, using the same seven-task real-robot evaluation protocol used in the main paper. Each variant is evaluated with ten ID and ten OOD trials per task. Table 9 reports the resulting success rates.

Table 9: Ablation of the training-only OT regularizer on the seven-task real-robot evaluation. Each ID or OOD entry aggregates 70 trials; Combined pools all 140 trials.

Variant	ID success	OOD success	Combined success
Atom-DH (+OT)	38.57% (27/70)	31.43% (22/70)	35.00% (49/140)
Atom-DH (w/o OT)	**44.29% (31/70)**	**42.86% (30/70)**	**43.57% (61/140)**

Disabling OT improves ID, OOD, and combined success by 5.72, 11.43, and 8.57 percentage points, respectively. Under this protocol, explicit latent transport alignment does not improve downstream control and produces the largest degradation under OOD conditions. We therefore treat OT as an auxiliary training variant rather than a default component of Atom-DH.