TRIBRAIN: A THREE-SYSTEM EMBODIED FOUNDATION MODEL FOR SCALABLE ROBOT MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language-action (VLA) models have become a leading approach to generalist robot control, yet unifying understanding, prediction, and control within an multi-embodiment foundation model remains challenging. In particular, staged VLA training separates vision-language adaptation from continuous-action learning, limiting the co-adaptation of semantic representations and continuous control. Moreover, integrating world models into VLA pipelines for data augmentation or visual subgoal prediction, without coupling future-state prediction with task-progress evaluation, limits their effectiveness in guiding long-horizon manipulation. To address these issues, we present TRIBRAIN, a multi-embodiment VLA pretraining framework. The model adopts a three-system architecture: SYSTEM 1 converts multimodal context into continuous robot actions; SYSTEM 2 performs scene understanding and task decomposition; and SYSTEM 3 predicts future observations and estimates task progress to guide action generation. TRIBRAIN jointly trains a pretrained VLM backbone and a continuous-action expert while incorporating multimodal understanding objectives, mitigating optimization fragmentation. To balance semantic preservation and motor adaptation during joint training, Soft Knowledge Insulation attenuates action gradients entering the vision-language backbone. During policy post-training, a pretrined World Value Model (SYSTEM 3) provides predicted subgoal images and task-progress estimates used to derive binary progress labels. The subgoal images and progress labels serve as policy inputs and are independently dropped during training. Trained on over 37,000 hours of heterogeneous embodied experience, TRIBRAIN surpasses $\pi_{0.5}$ in real-world manipulation, achieving 89.2% and 74.1% success on complex tasks with AgileX PiPER and Maker H01, respectively. In simulation, it outperforms $\pi_{0.5}$ by 9.0% and 5.22% on RoboTwin 2.0 and EBench, respectively. In embodied vision-language evaluation on MiMo-Embodied, it also achieves SOTA performance with an overall score of 0.4621. Code, datasets and pretrained weights will be released.

1 INTRODUCTION

Vision-language-action (VLA) models have emerged as a leading approach to generalist robot control, adapting pretrained vision-language models (VLMs) to translate visual observations and language instructions into executable actions (Brohan et al., 2023; Kim et al., 2024; Black et al., 2024; GigaBrain Team et al., 2025). Recent advances in training scale and data diversity have improved real-world manipulation and transfer across tasks, environments, and embodiments (Physical Intelligence et al., 2025; Yu et al., 2026; Yuan et al., 2026; Xiaomi Robotics Team et al., 2026; Wu et al., 2026), while complementary work incorporates world models to predict future states and evaluate action outcomes (Lv et al., 2026; Team et al., 2026). These developments motivate a unified multi-embodiment foundation model in which task understanding, future-state prediction and evaluation, jointly guide continuous action generation. However, there are two challenges remain.

First, representation–control fragmentation in VLA pretraining. Most generalist models follow multi-stage pipelines (Figure 1(a)): they append action decoders only during post-training (Kim et al., 2025; Physical Intelligence et al., 2025), preventing early co-adaptation between multimodal semantics and continuous control. Conversely, naive one-stage pretraining passes unconstrained action gradients ($\nabla \mathcal{L}_{\mathrm{FM}}$) directly into the VLM backbone (Figure 1(b)), causing catastrophic semantic degradation (Driess et al., 2025). We resolve this trade-off with *Soft Knowledge Insulation (Soft KI)*

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

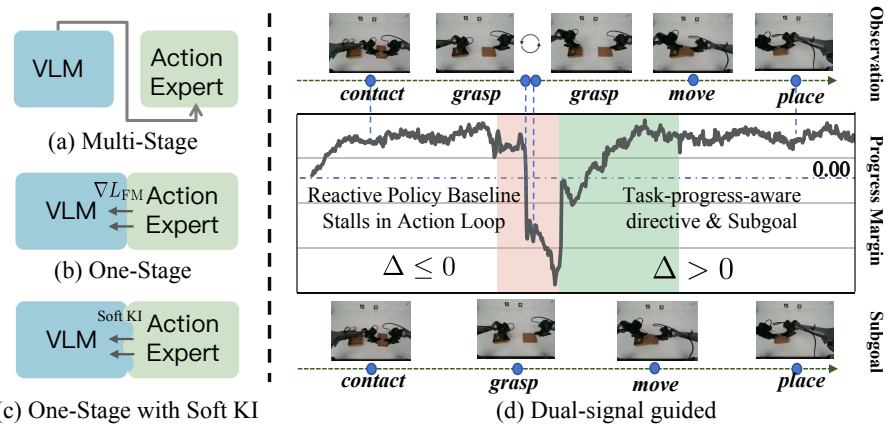


Figure 1: **Left:** One-stage VLM-action pretraining with Soft KI, attenuating action gradients into the vision-language trunk. **Right:** Dual-signal guidance. During post-training, SYSTEM 3 couples prospective visual subgoals with temporal progress evaluation to guidance actions generation.

(Figure 1(c)). Soft KI attenuates action gradients entering the visual trunk ($\alpha_{\text{KI}} \nabla \mathcal{L}_{\text{FM}}$), allowing continuous motor feedback to calibrate visual representations while preserving open-vocabulary semantics. **Second, reactive stalling and unverified execution progress.** Existing generalist policies remain largely reactive, lacking mechanisms to verify execution progress. In practice, minor execution deviations can easily induce cyclic action loops ($\Delta \leq 0$, Figure 1(d), red region). Recent works introduce predictive signals to mitigate this issue, such as prospective visual subgoals (Kim et al., 2026) or scalar progress values (Intelligence et al., 2025). However, these approaches provide only one-sided guidance. Furthermore, steerable policies (Physical Intelligence et al., 2026) introduce multi-modal prompts, yet they depend heavily on external human-curated inputs. To bridge this gap, TRIBRAIN couples prospective visual subgoals ($\mathcal{G}_t$) with temporal progress directives ($\mathcal{P}_t$) derived from an autonomous World Value Model (SYSTEM 3) to guide continuous action generation. Additionally, to prevent over-reliance on either modality, we independently drop both signals during post-training, improving policy robustness against predictive noise. This dual guidance effectively breaks behavioral deadlocks and restores forward task progression (Figure 1(d), green region).

To address these challenges, we introduce TRIBRAIN, a three-system embodied foundation model for control across diverse robot embodiments. It separates task understanding and planning, predictive evaluation, and action generation into three specialized systems. SYSTEM 1 (action and control) combines a 3B PaliGemma2 backbone (Steiner et al., 2024) with a 0.5B continuous Action Expert in a layer-wise dual-stream Transformer. SYSTEM 2 (understanding and planning) grounds a high-level instruction in current and historical observations and predicts the immediate subtask. SYSTEM 3 (prediction and evaluation) is a 5B World Value Model based on GigaWorld-1 (Team et al., 2026); it provides predicted subgoal images and compact task-progress evaluation to the SYSTEM 1.

Under this architectural factorization, we mitigate representation–action fragmentation through single-stage joint pretraining of SYSTEM 1 and SYSTEM 2. It supports two complementary prediction pathways: an autoregressive next-token prediction pathway for language, subtask, and discrete-action outputs, and a flow-matching pathway for continuous action chunks. This formulation enables semantic representations and continuous control to co-adapt throughout pretraining. Meanwhile, Soft Knowledge Insulation (KI) balances semantic preservation with motor adaptation by attenuating action gradients entering the vision-language backbone while retaining unattenuated gradients within the Action Expert. To address limited temporal foresight, we incorporate SYSTEM 3 as a pretrained World Value Model during policy post-training. It provides predicted subgoal images and binary progress labels derived from task-progress estimates. These complementary signals couple future-state prediction with progress evaluation, providing predictive guidance for action generation. Its future predictions span the target subtask, extending beyond a single action chunk. Additionally, we independently drop the subgoal and progess-aware guidance during training to reduce reliance on either signal. We construct a heterogeneous training corpus comprising over 37,000 hours of embodied trajectories spanning 16 robot types, together with large-scale vision-language data.

Our contributions are three-fold:

- We present TRIBRAIN, a multi-embodiment foundation model. The model adopts a three-system architecture: SYSTEM 1 converts multimodal context into continuous robot actions; SYSTEM 2 performs scene understanding and task decomposition; and SYSTEM 3 predicts future observations and estimates task progress to guide action generation.
- During pretraining, we achieve single-stage co-adaptation of multimodal semantics and continuous control using Soft KI, which attenuates action gradients to prevent catastrophic forgetting. During post-training, a pretrained World Value Model (SYSTEM 3) supplies prospective visual subgoals and temporally aligned progress directives with independent dropout, guiding continuous actions toward verified task completion.
- We construct a heterogeneous training corpus with over 37,000 hours of embodied trajectories spanning 16 robot types and over 270 million vision-language samples. TRIBRAIN outperforms $\pi_{0.5}$ on real-world manipulation, RoboTwin 2.0, EBench, and RoboColiseum. It also achieves SOTA performance on the MiMo-Embodied vision-language benchmark.

2 RELATED WORK

VLA architectures. VLA architectures adapt pretrained vision-language representations for robot control. They mainly differ in action representation and in how the action module connects to the VLM. We distinguish three families: autoregressive VLM-as-Actor, serial VLM–Action Expert, and parallel Mixture-of-Transformers (MoT) models. Autoregressive models such as RT-2 and OpenVLA represent continuous actions as discrete token sequences (Brohan et al., 2023; Kim et al., 2024). Compact and learned action tokenizers improve this representation (Pertsch et al., 2025; Lee et al., 2024; Dong et al., 2026; Galaxea Team, 2026), but sequential decoding becomes costly with higher control rates, longer horizons, and larger action spaces. Serial models use the VLM as a context encoder and a separate Action Expert for continuous control, as in GR00T N1, Gemini Robotics, and Qwen-RobotManip (NVIDIA et al., 2025b; Gemini Robotics Team et al., 2025; Yuan et al., 2026). This separation supports motor specialization but makes performance depend on the VLM–action interface. Parallel MoT models couple a continuous Action Expert to the VLM at multiple layers (Black et al., 2024; Physical Intelligence et al., 2025; GigaBrain Team et al., 2025; Yu et al., 2026). TRIBRAIN extends this design with separate interfaces for task understanding and planning, and for future-state prediction and evaluation.

Joint vision-language-action learning. Most VLAs initialize from pretrained vision-language models and add robot action supervision, differing in how these objectives are coupled during optimization. Multi-stage methods pretrain or align VLM and action modules separately before joint training. RT-2 and OpenVLA adapt multimodal representations via action-token prediction; $\pi_{0.5}$ introduces a flow-matching Action Expert during post-training; and Xiaomi-Robotics-0/1 separate VLM pretraining from embodiment alignment (Brohan et al., 2023; Kim et al., 2024; Physical Intelligence et al., 2025; Cai et al., 2026; Xiaomi Robotics Team et al., 2026). This stabilizes optimization and preserves semantic knowledge, but can weaken module coupling and cause interface mismatch. In contrast, one-stage methods jointly optimize the full model from the outset (GigaBrain Team et al., 2025; Yu et al., 2026; Yuan et al., 2026; Wu et al., 2026). While enabling end-to-end co-adaptation, this approach complicates optimization, increases sensitivity to data mixtures, and risks cross-modal interference. TRIBRAIN adopts this paradigm and introduces Soft Knowledge Insulation (KI). Unlike the original Knowledge Insulation (Driess et al., 2025) that strictly freezes the visual backbone and impedes continuous motor adaptation, Soft KI applies an asymmetric gradient attenuation mechanism. This allows the VLM backbone to receive calibrated motor feedback without compromising its open-vocabulary representations.

World models for VLA. World models provide a complementary approach to improving VLAs by learning predictive structure from large-scale visual experience. Existing methods use world models for several purposes, including embodied data generation, future-dynamics modeling, unified world-action prediction, policy conditioning, value estimation, and closed-loop policy refinement. For example, GigaBrain-0, DreamGen, and related approaches use video world models to synthesize diverse embodied experiences and expand the data available for policy training (GigaBrain Team et al., 2025; Jang et al., 2025). Other methods (Cheang et al., 2024; Ye et al., 2026; Zhu et al., 2025; NVIDIA et al., 2025a) jointly model future visual dynamics and robot actions within a unified

generative model. World-VLA-Loop further explores iterative refinement between a world model and a VLA policy through policy-generated failure rollouts (Liu et al., 2026). Additionally, distinct from $\pi^*_{0.6}$ and $\pi_{0.7}$ (Intelligence et al., 2025; Physical Intelligence et al., 2026) which solely rely on scalar ratings or human-labeled failures, and unlike RL−oriented world models (e.g., Viva, GigaBrain-0.5M* (Lv et al., 2026; GigaBrain Team et al., 2026)) used for rollout generation, TRIBRAIN couples spatial target foresight with temporal progress evaluation in SYSTEM 3. To ensure robustness against predictive noise, we independently drop the subgoal and progress-aware guidance during post-training, preventing catastrophic reliance on either modality.

3 METHODS

Given a task instruction ℓ, recent visual observations $o_{t-k:t}$, proprioceptive state s_t, and embodiment identifier e, the goal is to generate an action chunk $a_{t:t+H}$. The model factors this mapping into three functional systems. SYSTEM 2 provides a semantic subtask $\hat{\ell}_t$; SYSTEM 3 supplies a future visual target $\mathcal{G}_t$ and progress-aware guidance $\mathcal{P}_t$; and SYSTEM 1 implements the action policy

$$a_{t:t+H} \sim \pi_\theta\Big(\cdot \mid o_{t-k:t}, s_t, e, \ell, \hat{\ell}_t, \mathcal{G}_t, \mathcal{P}_t\Big)\,. \tag{1}$$

The systems are functionally coordinated rather than serially chained: SYSTEM 1 and 2 are jointly trained as a unified VLA, with SYSTEM 2 providing scene understanding and subtask instructions, while SYSTEM 3 is independently pretrained and then frozen. During VLA post-training, SYSTEM 3 supplies complementary future-observation and task-progress predictions to SYSTEM 1 without receiving gradients. The complete three-stage training and inference protocol is given in Appendix A.

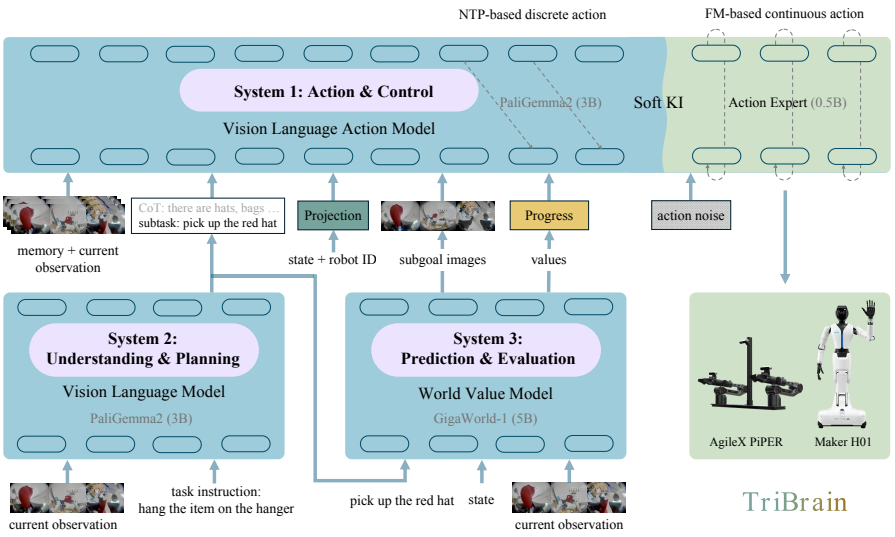


Figure 2: **TRIBRAIN:** SYSTEM 1 generates continuous actions, guided by SYSTEM 2's subtask understanding and SYSTEM 3's future-state and progress evaluation.

3.1 ACTION GENERATION AND PLANNING

SYSTEM 1: dual-stream action generation. SYSTEM 1 pairs the vision-language stream with a dedicated Action Expert. At layer l, let $\mathbf{H}_l^{\mathrm{VL}}$ and $\mathbf{H}_l^{\mathrm{AE}}$ denote the hidden states of the vision-language and Action Expert streams, respectively, and let $\mathbf{O}_l^{\mathrm{VL}}$ and $\mathbf{O}_l^{\mathrm{AE}}$ denote their attention outputs.

$$\mathbf{O}_l^{\mathrm{VL}} = \mathrm{CausalAttn}_l\left(\mathbf{H}_l^{\mathrm{VL}}; \mathbf{H}_l^{\mathrm{VL}}\right), \qquad \mathbf{O}_l^{\mathrm{AE}} = \mathrm{Attn}_l\left(\mathbf{H}_l^{\mathrm{AE}}; [\mathbf{H}_l^{\mathrm{VL}}, \mathbf{H}_l^{\mathrm{AE}}]\right). \tag{2}$$

Each stream then applies its own feed-forward network. The VLM stream therefore performs causal self-attention over its own states, whereas Action Expert queries use both semantic and motor states at every layer. This directional coupling injects evolving semantic information into continuous action generation without feeding motor tokens into the VLM stream, allowing the VLM to retain its autoregressive prediction role while the Action Expert models continuous action chunks.

SYSTEM 2: understanding and planning. Existing VLAs either predict actions directly from the task instruction and visual context, as in autoregressive action-token models (Brohan et al., 2023; Kim et al., 2024), or use a cascaded architecture in which a language model provides context to a separately parameterized motor module (NVIDIA et al., 2025b; Gemini Robotics Team et al., 2025; Yuan et al., 2026). We make the intermediate subtask an explicit prediction of SYSTEM 2. SYSTEM 2 uses the PaliGemma2 vision-language model to perform scene understanding, reason about the current task stage, and decompose long-horizon instructions into subtasks. The predicted subtask guide both SYSTEM 1 for action generation and SYSTEM 3 for future-observation and task-progress prediction. During VLA pre-training, we mix examples that provide both the task instruction and its aligned subtask with examples that provide only the task instruction, requiring SYSTEM 2 to infer the current subtask before action generation. This mixed supervision allows the model to use explicit subtasks when available and to recover them from the task instruction otherwise.

3.2 PREDICTION AND EVALUATION

SYSTEM 3: World Value Model. Prior work has shown that action generation can benefit from signals describing future task states and execution quality. For example, $\pi^*_{0.7}$ uses execution-quality annotations, while $\pi^*_{0.6}$ derives progress signals from a VLM-based value function (Intelligence et al., 2025; Physical Intelligence et al., 2026). Motivated by this works, SYSTEM 3 provides SYSTEM 1 with two complementary signals: a visual subgoal predicted by an independently pretrained world model and a binary progress signal derived from temporal value differences. Specifically, SYSTEM 3 is initialized from GigaWorld-1 (Team et al., 2026) and pretrained in two stages. First, it learns robot-centric future-observation prediction from embodied context, including the current observation, proprioceptive state, and subtask instruction produced by SYSTEM 2. Second, we add a value head that estimates task progress from current and historical observations. The value head is trained using trajectory-level completion annotations. SYSTEM 3 is frozen during policy post-training and does not receive gradients from SYSTEM 1.

Joint Target and Progress Guidance with Independent Dropout. Driven by the robot state, current observation, and subtask context, SYSTEM 3 jointly predicts a prospective visual subgoal $\mathcal{G}_t$ and estimates progress values to derive a binary task-progress-aware directive as follow,

$$\mathcal{P}_t = \mathbb{I}\left[\hat{V}_{t+\Delta} - \hat{V}_t > 0\right] \in \{0, 1\}. \tag{3}$$

Furthermore, to avoid over-reliance on external predictions, $\mathcal{G}_t$ and $\mathcal{P}_t$ are independently dropped during post-training with probabilities $p_g = 0.50$ and $p_A = 0.15$, respectively. The higher dropout on $\mathcal{G}_t$ prevents overfitting to high-dimensional visual targets, whereas the lower dropout on $\mathcal{P}_t$ maintains continuous exposure to progress signals while preserving unguided execution. At deployment, we set $\mathcal{P}_t = 1$ as an active forward prior. This encourages the Action Expert to advance toward the visual subgoal and accelerate task completion, rather than repeating stalled behaviors.

3.3 ONE-STAGE TRAINING OBJECTIVE

We jointly pre-train SYSTEM 1and SYSTEM 2using a unified objective that couples semantic understanding, hierarchical task prediction, and continuous action generation. Training is conducted on an in-house heterogeneous corpus exceeding 37,000 hours. This dataset spans diverse sources, including real-robot, UMI, egocentric (EGO), simulation, world-model generated trajectories, and vision-language data (Appendix B). For target sequence $y_{1:M}$ and multimodal context c, language, subtask, vision-language, and discrete-action targets use next-token prediction (NTP):

$$\mathcal{L}_{\text{NTP}} = -\mathbb{E}\left[\sum_{j=1}^{M} m_j \log p_\theta(y_j \mid c, y_{<j})\right]. \tag{4}$$

For continuous control, ground-truth action chunk a is interpolated with Gaussian noise ϵ at flow time τ as $a^\tau = \tau a + (1-\tau)\epsilon$. The Action Expert predicts the velocity field:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{a,\epsilon,\tau}\left[\|(v_\theta(a^\tau, c, \tau) - (a - \epsilon))\|_2^2\right], \quad \tau \in [0, 1], \tag{5}$$

Both objectives are active from the start, $\mathcal{L}_{\text{VLA}} = \mathcal{L}_{\text{NTP}} + \mathcal{L}_{\text{FM}}$, avoiding a representation-to-control handoff between pretraining stages.

Continuous motor gradients can nevertheless damage the VLM. Soft Knowledge Insulation scales only their contribution at the backbone boundary:

$$\nabla_{\theta_{\mathrm{VL}}} \mathcal{L}_{\mathrm{VLA}} = \nabla_{\theta_{\mathrm{VL}}} \mathcal{L}_{\mathrm{NTP}} + \alpha_{\mathrm{KI}} \nabla_{\theta_{\mathrm{VL}}} \mathcal{L}_{\mathrm{FM}}, \quad 0 < \alpha_{\mathrm{KI}} < 1. \tag{6}$$

The Action Expert receives the full flow-matching gradient. This controlled asymmetry lets visual features become action-aware without surrendering open-vocabulary semantics.

4 EXPERIMENTS

To comprehensively evaluate TRIBRAIN, our experiments span three core dimensions: foundation capabilities (Sec. 4.2), real-world post-training (Sec. 4.3), and simulation benchmarks (Sec. 4.4). Following prior real-robot manipulation evaluations (GigaBrain Team et al., 2025; 2026), we evaluate each setting over 20 physical rollouts and report the mean normalized task-completion score under the task-specific rubric described in Appendix C. To assess whether the observed gains arise from architectural decoupling rather than parameter scaling or execution scheduling, we further provide capacity-matched architecture and runtime comparisons in Appendix D and E.

4.1 EXPERIMENTAL SETUP

Pretraining Data and Baseline Evaluation. We pretrain TRIBRAIN on 37,256.98 trajectory hours spanning 16 robot embodiments and 272M vision-language samples (Appendix B). We compare against leading open-source VLAs, including $\pi_{0.5}$ (Physical Intelligence et al., 2025), GigaBrain-0.1 (GigaBrain Team et al., 2025), Galaxea G0.5 (Galaxea Team, 2026), and Xiaomi-Robotics-1 (Xiaomi Robotics Team et al., 2026). For post-training, all methods use the same downstream training dataset. Baselines are initialized from official public checkpoints and fine-tuned using their recommended configurations. Simulation evaluations follow the standard protocol of each benchmark, with matched camera views, observation inputs, and control frequencies across methods.

Cross-Embodiment Evaluation. To evaluate transfer from multi-embodiment pretraining to downstream control, we initialize separate post-training runs for AgileX PiPER and Maker H01 from a common TriBrain pretrained checkpoint, using the corresponding action-space masks. The adapted policies are evaluated on matched instruction-following and contact-rich manipulation suites under consistent observation and control protocols.

Simulation and Embodied Understanding. In simulation, policies are evaluated on RoboTwin 2.0 (Chen et al., 2026), EBench (Gao et al., 2026), and RoboColiseum (rob, 2026) to assess domain randomization, mobile manipulation, and Real2Sim2Real transfer. In addition, we evaluate open-vocabulary spatial and affordance reasoning on the MiMo-Embodied benchmark (Hao et al., 2026).

4.2 FOUNDATION MODEL PERFORMANCE

Effects of Data Scaling and Source Diversity. Increasing pretraining data reduces validation loss and improves real-robot performance under a fixed model configuration (Figure 3). The benefits are task-dependent: fruit manipulation reaches 100% success at 60% of the robot-data scale, whereas clothes folding improves from 70% to 100% as the scale increases from 60% to 100%. Additional robot experience therefore continues to benefit deformable-object manipulation after performance on the simpler task saturates. We further examine source composition by augmenting a 3,000-hour robot dataset with 3,000 hours of UMI or EGO demonstrations. Both additions improve performance before and after post-training, with UMI consistently outperforming EGO at the same total data volume. Combining all three sources yields the strongest overall performance, reaching 100% success on both tasks after post-training. The gains from UMI and EGO augmentation persist through task-specific adaptation, demonstrating benefits for both out-of-the-box control and downstream task performance.

In-Domain (ID) and Out-of-Domain (OOD) Generalization. Figure 4 shows that our pretrained policy acquires reusable manipulation primitives, but out-of-the-box transfer remains physics-dependent. On AgileX PiPER, the policy achieves 84.4% success on OOD fruit grasping and retains 60.0% on unseen cloths, demonstrating robust visual grounding and closed-loop tracking. However, performance drops sharply on tasks requiring fine-grained physical contact: button pressing yields 0% across both embodiments, and gift wrapping fails completely on PiPER (0%). This

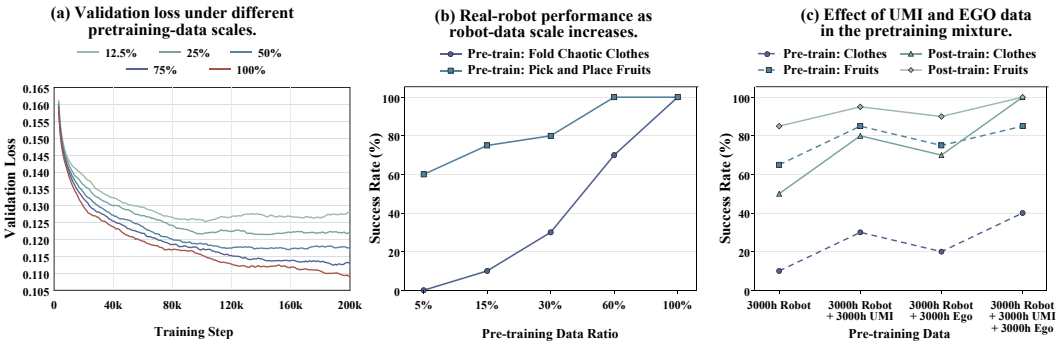
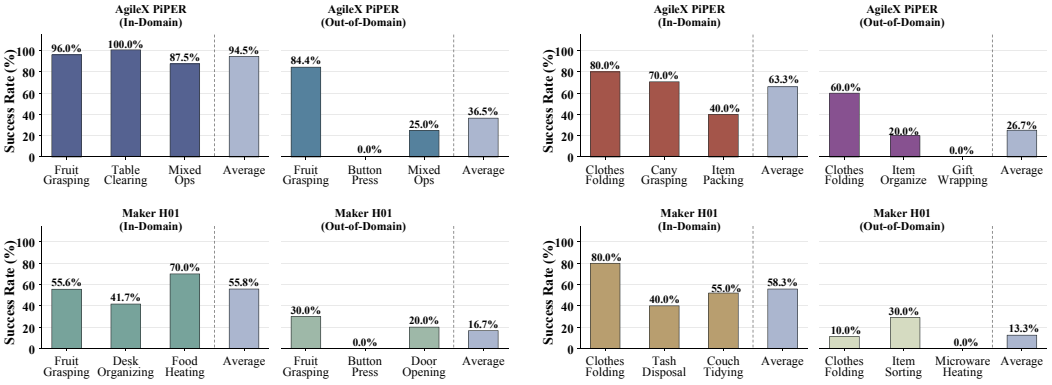


Figure 3: Effects of pretraining data scale and source composition.

failure stems from two fundamental limits of vision-only VLAs. First, button pressing demands sub-centimeter accuracy under severe end-effector occlusion; without force-torque or tactile feedback, position-based delta actions easily slip or fail to overcome mechanical spring resistance. Second, gift wrapping involves complex elastoplastic paper deformation; while SYSTEM 3 provides coarse spatial subgoals, video world models struggle to synthesize thin-sheet micro-creases accurately, compounding spatial errors over long execution horizons. Furthermore, the sharp gap in cloth folding between PiPER (60.0%) and H01 (10.0%) highlights that policy transfer on deformable media is highly sensitive to gripper morphology and compliance. While large-scale pretraining establishes strong semantic-spatial representations, closed-loop force sensing and high-fidelity physical modeling remain critical for contact-rich manipulation (Appendix I.2 for qualitative OOD cloth-folding).



(a) Language-following tasks on PiPER and H01.

(b) Complex manipulation tasks on PiPER and H01.

Figure 4: **Out-of-the-box foundation behavior.** The same pretrained checkpoint is evaluated without task-specific adaptation. ID and OOD settings vary objects, layouts, and task composition.

4.3 POST-TRAINING EVALUATION

Real-robot Performance. Table 1, We evaluate six instruction-conditioned manipulation tasks on AgileX PiPER and Maker H01, covering three complementary language-following capabilities: color discrimination (button push and spoon grasping), target discrimination (bowl and tableware pick-and-place), and direction discrimination (fork and wood-block pick-and-place). As Table 1 showing, TRIBRAIN achieves 91.5% on PiPER and 84.2% on H01, exceeding $\pi_{0.5}$ by 2.7 and 9.0 percentage points, respectively (qualitative examples in Appendix I.3). The task-level results reveal different sources of improvement. On PiPER, both models already achieve 100% on the two color tasks, with gains concentrated in target and direction discrimination. On H01, color discrimination accounts for the largest improvements, although tableware performance decreases. These contrasting patterns indicate that the remaining challenges in instruction-conditioned manipulation depend on both the task and the embodiment.

As shown in Table 2, TRIBRAIN achieves mean task-completion scores of 89.2% on Agilex PiPER and 74.1% on Maker H01 in complex manipulation, outperforming $\pi_{0.5}$ by 18.8 and 28.9 percentage

Table 1: Post-training multi-task language-following evaluation on AgileX PiPER and Maker H01. Values are normalized mean task-completion scores (%) over 20 trials.

Robot	Model	Color		Target		Direction		Avg.
		Button Push	Spoon Grasping	Bowl Pick-Place	Tableware Pick-Place	Fork Pick-Place	Wood Block Pick-Place	
AgileX PiPER	$\pi_{0.5}$	**100.0**	**100.0**	90.0	87.5	75.0	**80.0**	88.8
	GigaBrain-0.1	66.7	77.8	85.0	72.2	80.0	75.0	76.1
	G0.5	88.9	94.4	50.0	**100.0**	**85.0**	70.0	81.4
	Xiaomi-Robotics-1	66.7	83.3	50.0	98.6	75.0	60.0	72.3
	TRIBRAIN	**100.0**	**100.0**	**95.0**	88.9	**85.0**	**80.0**	**91.5**
Maker H01	$\pi_{0.5}$	66.7	34.4	**100.0**	**66.7**	83.3	**100.0**	75.2
	GigaBrain-0.1	44.4	34.4	**100.0**	55.6	83.3	**100.0**	69.6
	G0.5	72.2	12.5	25.0	31.9	12.5	0.0	25.7
	Xiaomi-Robotics-1	**100.0**	31.3	**100.0**	50.0	70.8	0.0	58.7
	TRIBRAIN	**100.0**	**56.3**	**100.0**	61.1	**87.5**	**100.0**	**84.2**

points, respectively. On Agilex PiPER, the largest gains occur in tableware arrangement and sweeping, while performance on sorting and storage remains unchanged. On H01, improvements span six of the seven tasks, including food preparation and dish cleanup. These gains extend across multiple forms of manipulation, suggesting broader task competence. However, sweeping success remains below 50% on both robots, highlighting a persistent weakness despite the overall improvements.

Table 2: Post-training complex manipulation evaluation on AgileX PiPER and Maker H01. Values are normalized mean task-completion scores (%) over 20 trials.

Robot	Model	Cube Sorting	Play-Dough Kneading	Tableware Arranging	Household Item Storage	Food Prep Heating	Dish Cleanup Washing	Rice Sweeping	Avg.
AgileX PiPER	$\pi_{0.5}$	**95.0**	95.0	77.8	**95.0**	50.0	60.0	20.0	70.4
	GigaBrain-0.1	90.0	95.0	38.9	75.0	50.0	60.0	25.0	62.0
	G0.5	45.0	20.0	38.9	85.0	20.0	20.0	0.0	32.7
	Xiaomi-Robotics-1	35.0	85.0	44.4	75.0	40.0	40.0	5.0	46.3
	TRIBRAIN	**95.0**	**100.0**	**94.4**	**95.0**	**100.0**	**100.0**	**40.0**	**89.2**
Maker H01	$\pi_{0.5}$	8.3	50.0	51.1	**91.7**	50.0	40.0	25.0	45.2
	GigaBrain-0.1	20.8	41.7	45.6	79.2	40.0	50.0	25.0	43.2
	G0.5	0.0	0.0	27.8	0.0	20.0	10.0	0.0	8.3
	Xiaomi-Robotics-1	8.3	20.8	34.4	54.2	30.0	40.0	41.7	32.8
	TRIBRAIN	**41.7**	**79.2**	**75.6**	**91.7**	**85.0**	**100.0**	**45.8**	**74.1**

4.4 SIMULATION EVALUATION

Simulation benchmarks. Table 3 compares representative publicly available models. Under RoboTwin 2.0, one policy is adapted jointly on all 50 tasks. TRIBRAIN is slightly below $\pi_{0.5}$ in the clean setting but is 21.9 points stronger under domain randomization, producing the best overall result. On EBench it reaches 33.30% success and 46.1 progress score, improving on $\pi_{0.5}$ by 5.22 points and 4.1 score. RoboColiseum uses reconstructed Real2Sim2Real environments; TRIBRAIN leads instruction following, spatial reasoning, robustness, and general manipulation. The consistent advantage under distribution shift complements the real-robot findings.

Table 3: **Simulation Result.** Results on RoboTwin 2.0, EBench, and RoboColiseum.

Model	RoboTwin 2.0			EBench		RoboColiseum			
	Easy	Hard	Overall	SR	Score	Instr.	Spatial	Robust.	Manip.
π_0	46.42	16.34	31.38	23.59	37.0	.3680	.1300	.3130	.3470
Xiaomi-Robotics-0	62.9	18.2	40.55	–	–	.6460	.2230	.5460	.3070
$\pi_{0.5}$	**70.7**	46.0	58.35	28.08	42.0	.7460	.3560	.6130	.5820
TRIBRAIN	66.8	**67.9**	**67.35**	**33.30**	**46.1**	**.8166**	**.4729**	**.6800**	**.6092**

4.5 ABLATIONS STUDY

SYSTEM 3 Ablation. We ablate visual foresight and task-progress guidance across three manipulation tasks (Figure 5). Visual subgoals (+subImage) provide spatial targets beyond the reactive action

horizon. This establishes feasible long-range geometric guidance, enabling valid packaging in gift wrapping ($0\% \rightarrow 20\%$). Meanwhile, progress directives (+value) inject directional momentum to escape local deadlocks, boosting gift wrapping to 60% and cube sorting to 45%. Combining both signals achieves the best performance (80% and 55%), confirming that spatial target specification and temporal progress tracking are orthogonal and mutually reinforcing (a qualitative rollout of stall recovery is detailed in Appendix I.1). Beyond binary completion, dual guidance substantially improves execution efficiency (Figure 5 (b), (c)). Even in clothes folding where success saturates at 100%, the combined signals increase the process score ($68.3\% \rightarrow 88.3\%$) and shorten completion time ($107\,\text{s} \rightarrow 75\,\text{s}$) by eliminating trajectory hesitation.

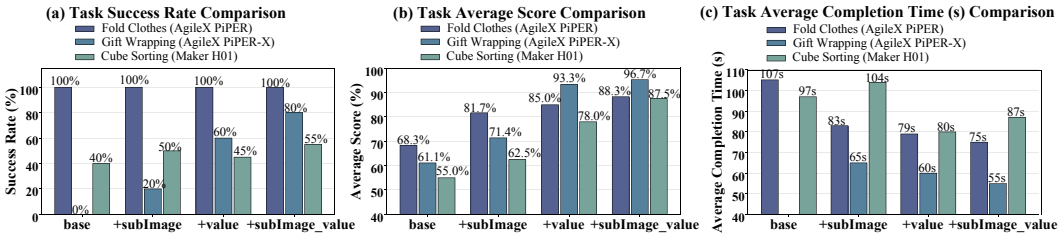


Figure 5: Ablation of the prediction and evaluation interfaces in SYSTEM 3.

Furthermore, the robustness of SYSTEM 3 are evaluated separately in Tables 12 and 13. Additionally, under identical FLOPs, our decoupled 3B+3B design outperforms a capacity-matched 6.8B single-backbone baseline by +0.74% on RoboTwin 2.0 and +9.1% on physical tasks (Appendix D, Table 10). These results demonstrate that the performance gains arise from structural decoupling, aligning with previous observations in OpenWAM (Wang et al., 2026).

Vision-language understanding with Soft KI. Soft KI attenuates action-loss gradients entering the VLM to support robot learning while limiting interference with pretrained vision-language knowledge. We evaluate embodied understanding on MiMo-Embodied (Hao et al., 2026), which includes spatial understanding and affordance reasoning. As shown in Table 4, TRIBRAIN achieves an overall score of 0.4621, compared with 0.3916 for G0.5. This improvement is driven by stronger spatial understanding (0.5215 vs. 0.3890), while affordance reasoning remains lower (0.3669 vs. 0.3956).

Table 4: **Vision-language understanding and robot control with Soft KI.** S/A/O denote Spatial/Affordance/Overall; O averages 13 benchmarks. Left: selected reported results.

(a) Reported MiMo results

Model	S	A	O
Xiaomi-Robotics-0	.1372	.0687	.1108
Spirit-v1.5	.3524	.2995	.3320
Wall-OSS-0.5	.2273	.0367	.1540
G0.5	.3890	.3956	.3916
Hy-Embodied-0.5-VLA-UMI	.2532	.0000	.1558
TRIBRAIN	**.5215**	**.3669**	**.4621**

(b) The ablation about α of Soft KI

Setting	S	A	O	PiPER	H01
Initial VLM	.4012	.3815	.3986	–	–
$\alpha = 0$	.3985	.3792	.3950	.3071	.2500
$\alpha = 0.01$	.4520	.3450	.4180	.6333	.5880
$\alpha = 0.1$	**.5215**	**.3669**	**.4621**	**.8750**	**.8019**
$\alpha = 0.5$	.2617	.1300	.2148	.5000	.4355
$\alpha = 1$	.1007	.0493	.0826	.3588	.2000

5 CONCLUSION

We introduced TRIBRAIN, a three-system multi-embodiment foundation model organized around two obstacles to VLA scaling: fragmented understanding-to-action interfaces and reactive decision making. A dual-stream, multi-embodiment policy jointly learns semantic and continuous control objectives while Soft Knowledge Insulation protects general representations. A World Value Model augments the policy with predicted future observations and progress estimates. Scaling the resulting architecture yields strong out-of-the-box behavior, cross-embodiment post-training, and robustness across real and simulated evaluations. More broadly, the results indicate that embodied scaling depends not only on collecting more experience, but on architectural interfaces that preserve, align, and prospectively evaluate that experience.

REFERENCES

Robocoliseum: A comprehensive, multi-dimensional benchmark for evaluating core capabilities of robotic models. https://robocoliseum.ai/, 2026. Accessed: 2026-08-14.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A Vision-Language-Action Flow Model for General Robot Control, 2024. URL https://arxiv.org/abs/2410.24164. arXiv preprint arXiv:2410.24164.

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

Rui Cai, Jun Guo, Xinze He, Piaopiao Jin, Jie Li, Bingxuan Lin, Futeng Liu, Wei Liu, Fei Ma, Kun Ma, Feng Qiu, Heng Qu, Yifei Su, Qiao Sun, Dong Wang, Donghao Wang, Yunhong Wang, Rujie Wu, Diyun Xiang, Yu Yang, Hangjun Ye, Yuan Zhang, and Quanyun Zhou. Xiaomi-Robotics-0: An Open-Sourced Vision-Language-Action Model with Real-Time Execution, 2026. URL https://arxiv.org/abs/2602.12684. arXiv preprint arXiv:2602.12684.

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, Hanbo Zhang, and Minzhao Zhu. GR-2: A Generative Video-Language-Action Model with Web-Scale Knowledge for Robot Manipulation, 2024. URL https://arxiv.org/abs/2410.06158. arXiv preprint arXiv:2410.06158.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=itonej9GIV.

Zibin Dong, Yicheng Liu, Shiduo Zhang, Baijun Ye, Yifu Yuan, Fei Ni, Jingjing Gong, Xipeng Qiu, Hang Zhao, Yinchuan Li, and Jianye Hao. ActionCodec: What Makes for Good Action Tokenizers, 2026. URL https://arxiv.org/abs/2602.15397.

Danny Driess, Jost Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, and Sergey Levine. Knowledge Insulating Vision-Language-Action Models: Train Fast, Run Fast, Generalize Better. In *Advances in Neural Information Processing Systems*, volume 38, pp. 102867–102888. Curran Associates, Inc., 2025. doi: 10.52202/085713-3439. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/94e936034d12bcd04834ec2773f02aff-Paper-Conference.pdf.

Galaxea Team. Galaxea G0.5 Technical Report. Technical report, Galaxea AI, 2026. URL https://opengalaxea.github.io/G05/Galaxea_G0_5.pdf.

Ning Gao, Jinliang Zheng, Xing Gao, Haoxiang Ma, Hanqing Wang, Yukai Wang, Jiantong Chen, Zanxin Chen, Shujie Zhang, Mingda Jia, Xuekun Jiang, Zihou Zhu, Xinyu Li, Shuai Wang, Hao Li, Wenzhe Cai, Yuqiang Yang, Xudong Xu, Zhaoyang Lyu, Yao Mu, Tai Wang, Jiangmiao Pang, Jia Zeng, Weinan Zhang, and Chunhua Shen. Ebench: Elemental diagnosis of generalist mobile manipulation policies, 2026. URL https://arxiv.org/abs/2606.18239.

Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, Steven Bohez, Konstantinos Bousmalis, Anthony Brohan, Thomas Buschmann, Arunkumar Byravan, Serkan Cabi, Ken Caluwaerts, Federico Casarini, Oscar Chang, Jose Enrique Chen, Xi Chen, Hao-Tien Lewis Chiang, Krzysztof Choromanski, David D'Ambrosio, Sudeep Dasari, Todor Davchev, Coline Devin, Norman Di Palo, Tianli Ding, Adil Dostmohamed, Danny Driess, Yilun Du, Debidatta Dwibedi, Michael Elabd, Claudio Fantacci, Cody Fong, Erik Frey, Chuyuan Fu, Marissa Giustina, Keerthana Gopalakrishnan, Laura Graesser, Leonard Hasenclever, Nicolas Heess, Brandon Hernaez, Alexander Herzog, R. Alex Hofer, Jan Humplik, Atil Iscen, Mithun George Jacob, Deepali Jain, Ryan Julian, Dmitry Kalashnikov, M. Emre Karagozler, Stefani Karp, Chase Kew, Jerad Kirkland, Sean Kirmani, Yuheng Kuang, Thomas Lampe, Antoine Laurens, Isabel Leal, Alex X. Lee, Tsang-Wei Edward Lee, Jacky Liang, Yixin Lin, Sharath Maddineni, Anirudha Majumdar, Assaf Hurwitz Michaely, Robert Moreno, Michael Neunert, Francesco Nori, Carolina Parada, Emilio Parisotto, Peter Pastor, Acorn Pooley, Kanishka Rao, Krista Reymann, Dorsa Sadigh, Stefano Saliceti, Pannag Sanketi, Pierre Sermanet, Dhruv Shah, Mohit Sharma, Kathryn Shea, Charles Shu, Vikas Sindhwani, Sumeet Singh, Radu Soricut, Jost Tobias Springenberg, Rachel Sterneck, Razvan Surdulescu, Jie Tan, Jonathan Tompson, Vincent Vanhoucke, Jake Varley, Grace Vesom, Giulia Vezzani, Oriol Vinyals, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Fei Xia, Ted Xiao, Annie Xie, Jinyu Xie, Peng Xu, Sichun Xu, Ying Xu, Zhuo Xu, Yuxiang Yang, Rui Yao, Sergey Yaroshenko, Wenhao Yu, Wentao Yuan, Jingwei Zhang, Tingnan Zhang, Allan Zhou, and Yuxiang Zhou. Gemini Robotics: Bringing AI into the Physical World, 2025. URL https://arxiv.org/abs/2503.20020. arXiv preprint arXiv:2503.20020.

GigaBrain Team, Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Haoyun Li, Jie Li, Jiagang Zhu, Lv Feng, Peng Li, Qiuping Deng, Runqi Ouyang, Wenkang Qin, Xinze Chen, Xiaofeng Wang, Yang Wang, Yifan Li, Yilong Li, Yiran Ding, Yuan Xu, Yun Ye, Yukun Zhou, Zhehao Dong, Zhenan Wang, Zhichao Liu, and Zheng Zhu. GigaBrain-0: A World Model-Powered Vision-Language-Action Model, 2025. URL https://arxiv.org/abs/2510.19430. arXiv preprint arXiv:2510.19430.

GigaBrain Team, Boyuan Wang, Bohan Li, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Jie Li, Jindi Lv, Jingyu Liu, Lv Feng, Mingming Yu, Peng Li, Qiuping Deng, Tianze Liu, Xinyu Zhou, Xinze Chen, Xiaofeng Wang, Yang Wang, Yifan Li, Yifei Nie, Yilong Li, Yukun Zhou, Yun Ye, Zhichao Liu, and Zheng Zhu. GigaBrain-0.5M*: a VLA That Learns From World Model-Based Reinforcement Learning, 2026. URL https://arxiv.org/abs/2602.12099. arXiv preprint arXiv:2602.12099.

Xiaoshuai Hao, Lei Zhou, Zhijian Huang, Zhiwen Hou, Yingbo Tang, Lingfeng Zhang, Guang Li, Zheng Lu, Shuhuai Ren, Xianhui Meng, Yuchen Zhang, Jing Wu, Jinghui Lu, Chenxu Dang, Jiayi Guan, Jianhua Wu, Zhiyi Hou, Hanbing Li, Shumeng Xia, Mingliang Zhou, Yinan Zheng, Zihao Yue, Shuhao Gu, Hao Tian, Yuannan Shen, Jianwei Cui, Wen Zhang, Shaoqing Xu, Bing Wang, Haiyang Sun, Zeyu Zhu, Yuncheng Jiang, Zibin Guo, Chuhong Gong, Chaofan Zhang, Wenbo Ding, Kun Ma, Guang Chen, Rui Cai, Diyun Xiang, Heng Qu, Fuli Luo, Hangjun Ye, and Long Chen. Mimo-embodied: X-embodied foundation model technical report, 2026. URL https://arxiv.org/abs/2511.16518.

Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachy Groom, Hunter Hancock, Karol Hausman, Gashon Hussein, Brian Ichter, Szymon Jakubczak, Rowan Jen, Tim Jones, Ben Katz, Liyiming Ke, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Yao Lu, Vishnu Mano, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Charvi Sharma, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Alex Swerdlow, James Tanner, Marcel Torne, Quan Vuong, Anna Walling, Haohuan Wang, Blake Williams, Sukwon Yoo, Lili Yu, Ury Zhilinsky, and Zhiyuan Zhou. $\pi^*_{0.6}$: a vla that learns from experience, 2025. URL https://arxiv.org/abs/2511.14759.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, Loic Magne, Ajay Mandlekar, Avnish Narayan,

You Liang Tan, Guanzhi Wang, Jing Wang, Qi Wang, Yinzhen Xu, Xiaohui Zeng, Kaiyuan Zheng, Ruijie Zheng, Ming-Yu Liu, Luke Zettlemoyer, Dieter Fox, Jan Kautz, Scott Reed, Yuke Zhu, and Linxi Fan. DreamGen: Unlocking Generalization in Robot Learning through Video World Models, 2025. URL https://arxiv.org/abs/2505.12705. arXiv preprint arXiv:2505.12705.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An Open-Source Vision-Language-Action Model, 2024. URL https://arxiv.org/abs/2406.09246. arXiv preprint arXiv:2406.09246.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-Tuning Vision-Language-Action Models: Optimizing Speed and Success, 2025. URL https://arxiv.org/abs/2502.19645. arXiv preprint arXiv:2502.19645.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos Policy: Fine-Tuning Video Models for Visuomotor Control and Planning, 2026. URL https://arxiv.org/abs/2601.16163. arXiv preprint arXiv:2601.16163.

Seungjae Lee, Yibin Wang, Haritheja Etukuru, H. Jin Kim, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Behavior Generation with Latent Actions, 2024. URL https://arxiv.org/abs/2403.03181.

Xiaokang Liu, Zechen Bai, Hai Ci, Kevin Yuchen Ma, and Mike Zheng Shou. World-vla-loop: Closed-loop learning of video world model and vla policy, 2026. URL https://arxiv.org/abs/2602.06508.

Jindi Lv, Hao Li, Jie Li, Fankun Kong, Yang Wang, Pengfei Yi, Yifei Nie, Xiaofeng Wang, Zheng Zhu, Chaojun Ni, Qiuping Deng, Hengtao Li, Jiancheng Lv, and Guan Huang. Viva: A video-generative value model for robot reinforcement learning. https://arxiv.org/abs/2604.08168, 2026. Accessed: 2026-08-16.

NVIDIA, :, Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, Daniel Dworakowski, Jiaojiao Fan, Michele Fenzi, Francesco Ferroni, Sanja Fidler, Dieter Fox, Songwei Ge, Yunhao Ge, Jinwei Gu, Siddharth Gururani, Ethan He, Jiahui Huang, Jacob Huffman, Pooya Jannaty, Jingyi Jin, Seung Wook Kim, Gergely Klár, Grace Lam, Shiyi Lan, Laura Leal-Taixe, Anqi Li, Zhaoshuo Li, Chen-Hsuan Lin, Tsung-Yi Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Arsalan Mousavian, Seungjun Nah, Sriharsha Niverty, David Page, Despoina Paschalidou, Zeeshan Patel, Lindsey Pavao, Morteza Ramezanali, Fitsum Reda, Xiaowei Ren, Vasanth Rao Naik Sabavat, Ed Schmerling, Stella Shi, Bartosz Stefaniak, Shitao Tang, Lyne Tchapmi, Przemek Tredak, Wei-Cheng Tseng, Jibin Varghese, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Xinyue Wei, Jay Zhangjie Wu, Jiashu Xu, Wei Yang, Lin Yen-Chen, Xiaohui Zeng, Yu Zeng, Jing Zhang, Qinsheng Zhang, Yuxuan Zhang, Qingqing Zhao, and Artur Zolkowski. Cosmos World Foundation Model Platform for Physical AI, 2025a. URL https://arxiv.org/abs/2501.03575. arXiv preprint arXiv:2501.03575.

NVIDIA, :, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An Open Foundation Model for Generalist Humanoid Robots, 2025b. URL https://arxiv.org/abs/2503.14734. arXiv preprint arXiv:2503.14734.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient Action Tokenization for Vision-Language-Action Models, 2025. URL https://arxiv.org/abs/2501.09747.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization, 2025. URL https://arxiv.org/abs/2504.16054. arXiv preprint arXiv:2504.16054.

Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, Vedant Choudhary, Foster Collins, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Maitrayee Dhaka, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachlan Groom, Haroun Habeeb, Hunter Hancock, Karol Hausman, Gashon Hussein, Victor Hwang, Brian Ichter, Connor Jacobsen, Szymon Jakubczak, Rowan Jen, Tim Jones, Gregg Kammerer, Ben Katz, Liyiming Ke, Mairbek Khadikov, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Brendon LeCount, Sergey Levine, Xinyu Li, Adrian Li-Bell, Vladislav Lialin, Zhonglin Liang, Wallace Lim, Yao Lu, Enyu Luo, Vishnu Mano, Nandan Marwaha, Aikys Mongush, Liam Murphy, Suraj Nair, Tyler Patterson, Karl Pertsch, Allen Z. Ren, Gavin Schelske, Charvi Sharma, Baifeng Shi, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Jiaming Tang, Jimmy Tanner, Shalom Tekeste, Marcel Torne, Kyle Vedder, Quan Vuong, Anna Walling, Haohuan Wang, Jason Wang, XuDong Wang, Chris Whalen, Samuel Whitmore, Blake Williams, Charles Xu, Sukwon Yoo, Lili Yu, Wuming Zhang, Zhuoyang Zhang, and Ury Zhilinsky. $\pi_{0.7}$: a Steerable Generalist Robotic Foundation Model with Emergent Capabilities, 2026. URL https://arxiv.org/abs/2604.15483. arXiv preprint arXiv:2604.15483.

Andreas Steiner, André Susano Pinto, Michael Tschannen, Daniel Keysers, Xiao Wang, Yonatan Bitton, Alexey Gritsenko, Matthias Minderer, Anthony Sherbondy, Shangbang Long, Siyang Qin, Reeve Ingle, Emanuele Bugliarello, Sahar Kazemzadeh, Thomas Mesnard, Ibrahim Alabdulmohsin, Lucas Beyer, and Xiaohua Zhai. PaliGemma 2: A Family of Versatile VLMs for Transfer, 2024. URL https://arxiv.org/abs/2412.03555. arXiv preprint arXiv:2412.03555.

GigaWorld Team, Angyuan Ma, Boyuan Wang, Bohan Li, Chaojun Ni, Guo Li, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, Jingyu Liu, Jiwen Lu, Qiuping Deng, Tingdong Yu, Xuancheng Xu, Xinyu Zhou, Xiuwei Xu, Xinze Chen, Xiaofeng Wang, Xiaoyu Tian, Yang Wang, Yifan Chang, Yukun Zhou, Yun Ye, Zhenyu Wu, Zhanqian Wu, and Zheng Zhu. Gigaworld-1: A roadmap to build world models for robot policy evaluation, 2026. URL https://arxiv.org/abs/2607.02642.

Yuran Wang, Siqiao Huang, Mingleyang Li, Chenhao Zhang, Jiaqi Liang, Weiyang Jin, Yue Chen, Xuemin Chi, Donghao Zhou, Qize Yu, et al. Openwam: An open, modular exploration towards systematic world-action model pretraining. *arXiv preprint arXiv:2609.07398*, 2026.

Wei Wu, Fangjing Wang, Fan Lu, He Sun, Shi Liu, Yunnan Wang, Yibin Yan, Yong Wang, Shuailei Ma, Xinyang Wang, Yibin Liu, Shuai Yang, Tianxiang Zhou, Kejia Zhang, Lei Zhou, Cheng Su, Nan Xue, Bin Tan, Han Zhang, Youchao Zhang, Fei Liao, Xing Zhu, Yujun Shen, and Kecheng Zheng. From Foundation to Application: Improving VLA Models in Practice, 2026. URL https://arxiv.org/abs/2607.06403. arXiv preprint arXiv:2607.06403.

Xiaomi Robotics Team, Jun Guo, Piaopiao Jin, Jason Li, Peiyan Li, Yingyan Li, Futeng Liu, Wanli Peng, Optimus Qin, Yifei Su, Nan Sun, Qiao Sun, Runze Suo, Heyun Wang, Yunhong Wang, Rujie Wu, Caoyu Xia, Lina Zhang, Jack Zhao, Guoliang Chen, Wenlong Chen, Xinze He, Bin Li, Qing Li, Zhuorong Li, Heng Qu, Wenxuan Song, Diyun Xiang, Yifan Xie, Peiran Xu, Hangjun Ye, Wen Ye, Han Zhao, and Quanyun Zhou. Xiaomi-Robotics-1: Scaling Vision-Language-Action Models with over 100K Hours of Real-World Trajectories, 2026. URL https://arxiv.org/abs/2607.15330. arXiv preprint arXiv:2607.15330.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan,

Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi "Jim" Fan, and Joel Jang. World Action Models are Zero-shot Policies, 2026. URL https://arxiv.org/abs/2602.15922. arXiv preprint arXiv:2602.15922.

Ryan Yu, Pushi Zhang, Starrick Liu, Brae Liu, Miracle Kang, Shalfun Li, Lights Shi, Ellie Ma, Ping Yang, Chris Pan, Jerry Chen, Dongxiu Liu, Rain Sun, Miles Guo, Byron Zhang, Hugo Zhou, Zach Xu, Vincent Chen, Harrison Huang, James Wang, Dance Kuzi, Andy Zhai, Hang Su, Roy Gan, Lucy Liang, Hao Wang, and Qian Wang. Wall-OSS-0.5 Technical Report, 2026. URL https://arxiv.org/abs/2605.30877. arXiv preprint arXiv:2605.30877.

Haoqi Yuan, Zhixuan Liang, Anzhe Chen, Ye Wang, Haoyang Li, Pei Lin, Yiyang Huang, Zixing Lei, Tong Zhang, Jiazhao Zhang, Jie Zhang, Jingyang Fan, Gengze Zhou, Qihang Peng, Chenxu Lv, Xiaoyue Chen, An Yang, Fei Huang, Junyang Lin, Dayiheng Liu, Jingren Zhou, Chenfei Wu, and Xiong-Hui Chen. Qwen-RobotManip Technical Report: Alignment Unlocks Scale for Robotic Manipulation Foundation Models, 2026. URL https://arxiv.org/abs/2606.17846. arXiv preprint arXiv:2606.17846.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified World Models: Coupling Video and Action Diffusion for Pretraining on Large Robotic Datasets. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.015. URL https://www.roboticsproceedings.org/rss21/p015.html.

A Model Training and Inference Protocol

TriBrain uses three coordinated optimization and execution stages. SYSTEM 1 and SYSTEM 2 are jointly optimized during VLA pretraining and policy post-training. SYSTEM 3 is pretrained separately and remains frozen during policy post-training. Table 5 summarizes the modules trained at each stage and the signals they provide. Specifically, the model adopts a three-stage training protocol, with the participation status of each system summarized in Table 5:

- **Stage 1 (Joint VLA Pretraining):** System 1 and System 2 are jointly optimized via combined next-token prediction (NTP) and flow-matching objectives. Soft Knowledge Insulation (Soft KI, with $\alpha_{KI} = 0.1$) is applied to attenuate action gradients entering the VLM backbone.
- **Stage 2 (System 3 Pretraining):** System 3 is pretrained independently on future-video prediction and task-progress value estimation.
- **Stage 3 (Guided VLA Post-training):** System 1 and System 2 are further fine-tuned with a frozen System 3. In this stage, System 2 predicts semantic subtasks, while System 3 provides visual subgoals and teacher-derived binary progress signals. During deployment, the progress directive $\mathcal{P}_t$ is fixed to 1.

Table 5: Three-stage training protocol of TriBrain. "Train" denotes active gradient updates, "–" denotes an unused module, and "Frozen" denotes inference without gradient updates.

Stage	SYSTEM 1	SYSTEM 2	SYSTEM 3
1. Joint VLA Pretraining	Train	Train	–
2. SYSTEM 3 Pretraining	–	–	Train
3. Guided VLA Post-training	Train	Train	Frozen

Table 6 summarizes the pre-training hyperparameters and multi-embodiment configurations. We pre-train TriBrain for 400k steps across a cluster of 256 NVIDIA A800 GPUs using PyTorch FSDP-v2. Soft Knowledge Insulation with $\alpha = 0.1$ is employed to enable effective action learning while safeguarding general VLM representations. Across heterogeneous datasets, embodiment-specific loss weighting and shuffled cycle sampling are applied for balanced multi-embodiment training.

Table 6: Pre-training hyperparameters and multi-embodiment training configurations.

Category	Shared Setup / Value
VLM Backbone	PaliGemma2-3B (224×224)
Action Chunk / Flow Horizon	50 / 50
Max Action Dimension	32
Action Normalization	Quantile Normalization
Compute Hardware	8×32 NVIDIA A800 (256 GPUs)
Global Batch Size	8192 ($8 \times 32 \times 32$)
Pre-training Steps	400k
Optimizer	AdamW (7.5×10^{-5})
Learning Rate Schedule	Warmup Cosine (Warmup: 1k, Decay: 200k)
Weight Decay / Max Grad Norm	10^{-5} / 1.0
Model Weight EMA	Enabled
Soft KI Gradient Weight (α)	0.1
Distributed Framework	PyTorch FSDP-v2
Multi-Embodiment Setup	
Visual Input	3-Camera RGB (224×224)
State Input Mode	Proprioceptive Anchor
Action Representation	Delta Joint / 6D Dual-Arm Pose
Oversampling Strategy	Shuffled Cycle (without replacement)
Loss Weight Balancing	UMI rot: 0.1; EGO rot & grip: 0.01

Table 7 summarizes the pre-training hyperparameters and multi-embodiment training configurations for SYSTEM 3 (Giga-World-1). We train the 5.0B world model across a cluster of 24×8 NVIDIA

Table 7: Pre-training hyperparameters and configurations for SYSTEM 3 (Value World Model).

Category	**Shared Setup / Value**
Base Model Architecture	Wan2.2-TI2V-5B (DiT-based World Model)
Latent Video DiT Dim / Layers	3072 / 30 layers (24 heads, head dim 128)
Value DiT Dim / Layers	1024 / 30 layers (24 heads, head dim 128)
Compute Hardware	24×8 NVIDIA A800 (192 GPUs)
Global Batch Size	3072 ($24 \times 8 \times 16$)
Pre-training Steps	200K
Precision	BF16 Mixed Precision
Optimizer	AdamW ($\beta_1 = 0.9, \beta_2 = 0.999$, weight decay 0.01)
Base Learning Rate	1.0×10^{-4}
Learning Rate Schedule	Cosine Annealing
Multi-Embodiment Setup	
Visual Resolution	Multi-Camera RGB (240×320, 3 views concatenated)
Video Frame Configuration	5 frames, Global Sample Stride 24 (at 30 FPS)
State / Action Dimension	32 (Padded fixed dimension, proprioceptive)
Text Embedding	Pre-cached embeddings (Context length: 128)
State Normalization	Z-score normalization
Value Target Mode	Rectified value progression

A800 GPUs using the AdamW optimizer with a cosine learning rate scheduler under BF16 mixed precision. Conditioned on text directives and proprioceptive states, the model synthesizes future multi-view visual frames and predicts execution values via Flow Matching, using a composite objective with video loss weight 1.0 and value loss weight 0.1.

The detailed post-training hyperparameters and platform-specific configurations for both AgileX PIPER and Maker H01 are summarized in Table 8.

Table 8: Post-training hyperparameters on AgileX PIPER and Maker H01.

Category	**Shared Setup / Value**
Action Chunk / Flow Horizon	50 / 50
Max Action Dimension	32
Action Normalization	Quantile Normalization
Global Batch Size	256 (8×32, 8-GPU)
Training Steps	30k
Optimizer	AdamW (2.5×10^{-5})
Model Weight EMA	Enabled
Platform-Specific Setup	
AgileX PIPER	Visual: 3-Camera RGB (224×224) Action: Delta Joint
Maker H01	Visual: 3-Camera Repacked RGB (224×224) Action: Delta Joint (with H01 Mask)

B Training Data Composition, Processing, and Data Isolation

This appendix describes the data mixture, trajectory processing pipeline, and separation between the training corpora assembled for this work and the downstream evaluation sets. The pretraining corpus contains embodied trajectories and vision-language (VL) supervision. Embodied trajectories provide state-action sequences and multi-robot control supervision, while VL data support visual-semantic, spatial, and affordance capabilities during joint action optimization.

B.1 Corpus Composition

The cleaned embodied trajectory corpus contains 37,256.98 hours from real robots, Universal Manipulation Interface (UMI) demonstrations, egocentric human interaction (EGO), physics-based simulation, and world-model-generated (WM) trajectories. The VL corpus contains 271,976,674 image-text and question-answering samples, including 15,234,327 self-collected embodied annotations. Table 9 summarizes the embodied corpus.

Table 9: Composition of the cleaned embodied trajectory corpus.

Data Source	**Duration (Hours)**	**Proportion (%)**
Real robot	20,535.65	55.12%
Universal Manipulation Interface (UMI)	8,251.83	22.15%
Egocentric human interaction (EGO)	2,862.36	7.68%
Physics-based simulation	1,453.92	3.90%
World-model generated (WM)	4,153.22	11.15%
Total	**37,256.98**	**100.00%**

The real-robot portion spans 16 robot types, 1,810,101 episodes, and 2,077,837,071 frames. Major platforms include AgileX PIPER, Maker H01, Maker M01, Agibot-G1, Ark, Galaxea R1 Lite, Franka Emika Panda, and UR5.

B.2 Embodied-Trajectory Standardized Processing

The data sources differ in storage format, degrees of freedom (DoF), coordinate conventions, camera naming, control frequency, and language granularity. We standardize trajectories to LeRobot v3.0 format using the following steps:

1. **Unified state-action layout:** Each embodiment is mapped to a common joint and end-effector layout (left arm, right arm, head, waist, and base). Unavailable DoFs are marked with embodiment-specific binary validity masks so they do not contribute artificial zero-valued targets to the loss.
2. **Continuous rotation representation:** Rotations use a continuous 6D representation to avoid the discontinuities of Euler angles and quaternions.
3. **Hierarchical annotation:** Raw human labels are normalized with an LLM-assisted rewriting pipeline and checked by human spot reviews. Atomic subtasks are aligned with temporal visual boundaries to guide SYSTEM 2.

C Real-Robot Evaluation Protocols and Completion Scoring

All physical tasks reported in Tables 1 and 2 use the normalized task-completion score defined below. Binary success, milestone completion, and continuous physical progress are treated as task-specific instances of this common scoring framework. We report

$$\text{Score} = 100 \times \frac{1}{N} \sum_{i=1}^{N} s_i, \qquad N = 20. \tag{7}$$

For binary tasks, $s_i \in \{0, 1\}$ and this reduces to the conventional success rate. For sequential or continuous-progress tasks, the rubric assigns fractional credit, so the reported score need not be an integer percentage. Averages over multiple tasks are unweighted arithmetic means of the task-level scores.

Task-specific scoring rubrics. The rubric is fixed for each task before evaluation and applied identically to all compared methods:

- **Binary completion.** Tasks with an unambiguous terminal criterion assign $s_i = 1$ only when the criterion is fully satisfied and $s_i = 0$ otherwise.

- **Sequential milestone completion.** For a task with M ordered milestones, $s_i = \sum_{m=1}^{M} w_m r_{i,m}$, where $r_{i,m} \in \{0, 1\}$ indicates whether milestone m is completed, $w_m \geq 0$, and $\sum_m w_m = 1$. Tableware Arranging uses weights 0.33, 0.33, and 0.34 for bowl placement, plate setting, and utensil alignment. Food Prep Heating scores microwave-door opening, tray insertion, and button activation as ordered milestones.
- **Continuous or multi-level physical progress.** Tasks involving deformable objects or granular materials use a normalized progress score in $[0, 1]$. Rice Sweeping measures the fraction of scattered rice transferred into the pan, Cube Sorting measures the fraction of cubes placed in their matching bins, and Spoon Grasping awards partial credit for an initial grasp and full credit for a sustained, slip-free lift. Button Push evaluates multi-color selection by awarding proportional credit for each correctly identified and depressed color button corresponding to the instruction

The physical task results in this appendix are descriptive summaries. We do not report a significance test for the deployment-time progress-signal comparison because the available results do not include a pre-specified statistical analysis for that comparison.

D Shared, Decoupled, and Capacity-Matched Models

We compare the proposed decoupled design with shared-backbone models to assess whether the observed gains can be explained by parameter count alone. The decoupled model has 6.5B active parameters across SYSTEM 1 and SYSTEM 2, compared with 3.5B for the shared 3B-backbone baseline. We evaluate the following configurations:

1. **Shared 3B backbone (3.5B):** A single PaliGemma2-3B trunk predicts subtask tokens and provides context to the 0.5B continuous Action Expert, using $\alpha_{KI} = 0.01$.
2. **Capacity-matched shared model (6.8B):** A 7B-class VLM backbone (approximately 6.3B non-embedding parameters) is paired with the same 0.5B Action Expert. It uses the same data mixture and number of training steps as the decoupled model; its training FLOPs are $1.86\times$ the shared 3B baseline.
3. **Decoupled 3B+3B (TriBrain, 6.5B):** SYSTEM 2 (3.0B) and SYSTEM 1 (3.5B) use separate backbones.

Table 10: Shared-backbone and decoupled architecture comparison. RoboTwin 2.0 reports mean and standard deviation across three training seeds. PIPER complex-task results report mean and 95% confidence interval across 20 trials.

Architecture	Active Params	Training FLOPs	Subtask Acc. (%)	RoboTwin 2.0 (SR %)	PIPER Complex (SR %)
Shared 3B backbone	3.5B	$1.0\times$	73.2 ± 1.1	58.40 ± 0.01	72.4 ± 3.8
Capacity-matched shared	6.8B	$1.86\times$	78.6 ± 0.9	60.85 ± 0.03	75.8 ± 3.4
Decoupled 3B+3B (TriBrain)	**6.5B**	$1.82\times$	$\mathbf{88.6 \pm 0.8}$	$\mathbf{61.59 \pm 0.00}$	$\mathbf{84.9 \pm 2.9}$
Difference from capacity-matched shared	$-0.3B$	$-0.04\times$	$+10.0$	$+0.74$	$+9.1$

To isolate the contribution of the S1/S2 architectural decoupling, Table 10 excludes System 3 from all compared configurations; System 3 ievaluated separately in Table 12 and 13. In Table 10, scaling the shared backbone from 3.5B to 6.8B parameters increases RoboTwin 2.0 performance by 2.45 points and PIPER performance by 3.4%. The decoupled model obtains a further 0.74% gain on RoboTwin 2.0 and a 9.1% gain on PIPER relative to the capacity-matched shared model. These results suggest that the decoupled architecture contributes beyond parameter scaling in the evaluated settings.

E Runtime Profile and Asynchronous Execution

We measure component inference latency using FP16 precision on a single workstation GPU. A serialized implementation runs the policy loop at approximately $\sim$2.0 Hz, with a 523 ms loop latency. In the evaluated PIPER tasks, this execution mode has a mean duration of 124.5 s and a mean completion score of 54.0%.

Table 11: Runtime profile and execution-paradigm comparison. Latencies are measured with FP16 precision on one workstation GPU. SYSTEM 1 generates action chunks at 10 Hz, while SYSTEM 2 and SYSTEM 3 refresh asynchronously at approximately 5 Hz/4 Hz.

(a) Component-level inference latency.

Module	**Deployment role**	**Params**	**Frequency**	**Latency**
SYSTEM 2	Subtask planning ($\hat{l}_t$)	3.0B	$\sim$5.0 Hz	190 ms
SYSTEM 3	Visual subgoal synthesis ($\mathcal{G}_t$)	5.0B	$\sim$4.0 Hz	240 ms
SYSTEM 1	Action chunk generation ($\mathcal{P}_t = 1$)	3.5B	$\sim$10.0 Hz	**93 ms**

(b) Execution-paradigm comparison on AgileX PIPER.

Execution paradigm	**Loop rate**	**Timing constraint**	**Mean duration**	**Mean completion**
Sequential synchronous (blocking)	$\sim$1.9 Hz	Misses 100 ms target	124.5 s	54.0%
Multi-rate asynchronous (ours)	**10.0 Hz**	**Real-time satisfied**	**75.2 s**	**84.9%**

SYSTEM 1 predicts an action chunk of horizon $H = 50$ in 93 ms. At a target control interval of 100 ms ($1/10.0$ Hz), this leaves a 10 ms margin for low-level robot joint interpolation and bus communication within each step. SYSTEM 2 and SYSTEM 3 run as background workers at approximately 1 Hz to asynchronously refresh the subtask plan and visual subgoal. During deployment, SYSTEM 1 receives the fixed positive-progress directive $\mathcal{P}_t = 1$, bypassing per-step value re-evaluations. However, because SYSTEM 3 continuously synthesizes prospective visual subgoals in the background, fixing $\mathcal{P}_t = 1$ does not eliminate the operational latency of SYSTEM 3 from the overall framework.

F Robustness to World-Model Guidance Errors

We evaluate the same trained policy with System 3 guidance degraded or omitted at deployment. The visual-subgoal and progress-signal inputs are independently dropped during post-training with probabilities $p_g = 0.50$ and $p_A = 0.15$.

Table 12: Deployment-time robustness and progress-signal comparisons on three real-robot tasks. All settings use the same trained checkpoint and 20 physical trials per task.

(a) System 3 guidance omission and corruption.

Guidance setting	**Signal**	**Folding**	**Wrapping**	**Sorting**	**Mean completion**	**Time (s)**
Full guidance (ours)	Clean $\mathcal{G}_t$, $\mathcal{P}_t = 1$	100.0%	80.0%	55.0%	**78.3%**	**74.8**
Guidance omission						
Subgoal omitted	$\mathcal{P}_t = 1$ only	100.0%	60.0%	45.0%	68.3%	79.5
Progress signal omitted	Clean $\mathcal{G}_t$ only	100.0%	20.0%	50.0%	56.7%	97.2
Both omitted	No System 3 guidance	100.0%	0.0%	40.0%	46.7%	107.0
Guidance corruption						
Corrupted subgoal	Latent visual noise ($\sigma = 0.3$)	90.5%	69.5%	50.0%	70.0%	83.1
Negative progress directive	$\mathcal{P}_t = 0$	90.0%	55.0%	45.0%	63.3%	96.8

(b) Deployment-time ablation of the progress directive.

Deployment progress input	**Refresh**	**Folding**	**Wrapping**	**Sorting**	**Mean completion**	**Time (s) [95% CI]**
Fixed $\mathcal{P}_t = 1$ (ours)	Constant	**100.0%**	**80.0%**	**55.0%**	**78.3%**	**74.8** $\pm$ 2.8
Online value-derived $\hat{\mathcal{P}}_t$	$\sim$1 Hz	100.0%	75.0%	50.0%	75.0%	91.2 $\pm$ 4.3
Fixed $\mathcal{P}_t = 0$	Constant	90.0%	55.0%	45.0%	63.3%	96.8 $\pm$ 5.1
No progress input	None	100%	0%	40.0%	46.7%	107.0 $\pm$ 6.2

Omitting both System 3 signals yields a mean completion score of 46.7%, close to the reactive baseline. Under the tested visual corruption and negative progress directive, the policy retains nonzero completion across all three tasks. These results demonstrate graceful degradation under

degraded guidance, while confirming that full performance relies on uncorrupted visual and progress signals.

Progress Directive Mechanism and Stall Resolution Dynamics. In Table 12 (b), we analyze the progress interface by comparing dynamically inferred directives ($\hat{\mathcal{P}}_t$) with a constant forward prior ($\mathcal{P}_t = 1$) using the same trained checkpoint. Conditioning policy execution on progress indicators provides bidirectional behavior modeling: exposing the policy to both advancing transitions ($\mathcal{P}_t = 1$) and stalled states ($\mathcal{P}_t = 0$) during post-training prevents representation collapse, enabling the Action Expert to distinguish productive motor transitions from unprogressive behaviors. The necessity of this bidirectional supervision is validated in Table 13, where training solely on constant positive labels (A_{fixed}) drops Gift Wrapping success from 80.0% to 45.0% due to the lack of progress discrimination.

As shown in Table 10(b), fixing $P_t = 1$ at deployment achieves a comparable mean completion score to online value-derived directives (78.3% vs. 75.0%), while reducing mean execution time from 91.2 s to 74.8 s (18.0%). These results support using a fixed positive-progress directive for efficient execution. In real-world manipulation, unexpected contact resistance or execution perturbations can halt task progress ($\Delta V \leq 0$). Feeding the dynamically inferred directive $\hat{\mathcal{P}}_t = 0$ back to the policy reinforces stationary behaviors, thereby perpetuating the stall. Conversely, setting $\mathcal{P}_t = 1$ at deployment acts as an active forward prior: when physical progress halts, prescribing $\mathcal{P}_t = 1$ prompts the Action Expert to sample progress-advancing action trajectories toward the prospective visual subgoal $\mathcal{G}_t$, autonomously breaking action loop. Meanwhile, closed-loop environmental grounding remains fully preserved through the dynamically refreshed visual subgoals from System 3.

Importantly, clamping $\mathcal{P}_t = 1$ does not compromise physical safety in unmodeled collision scenarios: the policy remains strictly grounded in real-time visual observations o_t and proprioceptive states s_t, while low-level joint torque limits and servo-level compliance thresholds on the robot platforms provide robust protection against excessive contact forces.

G System- and Component-Level Ablations

We evaluate component ablations on RoboTwin 2.0 and representative real-robot tasks. Table 13 reports RoboTwin 2.0 results as the mean and standard deviation across three training seeds. Real-robot task values are means over 20 trials.

Table 13: System- and component-level ablations on RoboTwin 2.0 and representative real-robot tasks. RoboTwin 2.0 reports mean and standard deviation across three seeds; physical task results report mean completion over 20 trials.

Configuration	SYSTEM 1	SYSTEM 2	SYSTEM 3	RoboTwin 2.0 (%)	Gift wrapping (%)	Cube sorting (%)
(1) Vanilla VLA (base)	3.5B	–	–	48.62 ± 0.00	0.0	40.0
(2) + Semantic planner	3.5B	3.0B	–	57.80 ± 0.03	0.0	45.0
(3) + Visual subgoal only	3.5B	3.0B	5.0B ($\mathcal{G}_t$ only)	61.59 ± 0.01	20.0	50.0
Progress-signal training ablations						
(4) Random progress labels	3.5B	3.0B	5.0B ($\mathcal{G}_t + A_{\text{rand}}$)	61.80 ± 0.04	20.0	45.0
(5) Constant positive label	3.5B	3.0B	5.0B ($\mathcal{G}_t + A_{\text{fixed}=1}$)	63.40 ± 0.01	45.0	50.0
(5b) Shuffled progress labels	3.5B	3.0B	5.0B ($\mathcal{G}_t + A_{\text{shuffled}}$)	62.85 ± 0.03	40.0	48.3
(6) TriBrain	**3.5B**	**3.0B**	**5.0B ($\mathcal{G}_t + \mathcal{P}_t$)**	$\mathbf{67.35 \pm 0.02}$	**80.0**	**55.0**

Temporally aligned progress signals. During post-training, the teacher-derived progress signal is computed as

$$\mathcal{P}_t = \mathbb{I}\left[\hat{V}_{t+\Delta} - \hat{V}_t > 0\right] \in \{0, 1\}. \tag{8}$$

This is a binary signal derived from the value difference between the current state and a model-imagined future state; At deployment, we use fixed $\mathcal{P}_t = 1$ as a positive-progress directive. Thus, the value-derived signal provides temporally aligned supervision during post-training, while the fixed value specifies the desired progress direction during execution.

To test whether alignment with the trajectory step matters, the shuffled-label baseline preserves the empirical label distribution ($P(\mathcal{P}_t = 1) \approx 62\%$) but permutes labels across trajectory samples within each training batch. The shuffled-label model scores 62.85% on RoboTwin 2.0 and 40.0%

on gift wrapping, compared with 67.35% and 80.0% for TRIBRAIN. The constant-positive-label model scores 63.40% on RoboTwin 2.0 and 45.0% on gift wrapping. The results support the use of trajectory-aligned progress supervision over randomly assigned or constant labels in the evaluated tasks.

H TRAIN-TEST ISOLATION AND DISTRIBUTION-SHIFT PROTOCOLS

We use the following protocols to separate the training corpora assembled for this work from downstream evaluation data and to define the evaluated distribution shifts:

1. **Physical evaluation-environment isolation:** The AgileX PIPER and Maker H01 evaluation arenas, including table surfaces, wall backgrounds, and third-person camera extrinsics, were constructed in separate testing cells. No training trajectories were collected in these evaluation cells.
2. **OOD instance screening:** Physical object instances used in OOD evaluation (including cloth textures, tableware geometries, and gift boxes) were screened against the training corpus using perceptual hashes and CLIP image embeddings. No exact image matches were found for the evaluated instances.
3. **Distribution-shift definitions:**
   - *In-domain (ID):* Familiar object categories and task compositions evaluated with randomized initial poses and distractor configurations.
   - *Out-of-domain (OOD):* Unseen object instances with different visual geometry or physical properties, as well as composite instructions requiring subtask chaining.

Benchmark data exclusion. No RoboTwin 2.0, EBench, RoboColiseum, or MiMo-Embodied benchmark episodes or examples were included in the embodied or VL corpora assembled for this work, SYSTEM 3 training data, or policy post-training data. This exclusion also covers simulator and world-model-generated samples derived from those benchmark episodes.

I QUALITATIVE RESULTS

I.1 QUALITATIVE ANALYSIS OF PROGRESS EVALUATION

To qualitatively examine how SYSTEM 3 evaluates task progress, we visualize a representative rollout involving a grasping failure and subsequent recovery. Figure 6 shows the corresponding observations, predicted future observations, and progress margin over time.

During the failure window, the robot repeatedly attempts to grasp the object without making meaningful progress. The predicted future observations remain close to the stalled configuration, while the progress margin decreases sharply. After the execution leaves the grasping loop, the predicted future observations depict continued task execution, and the progress margin recovers toward positive values. This temporal correspondence provides an interpretable example of how SYSTEM 3 combines future-state prediction with progress evaluation to identify stalled execution and recovery.

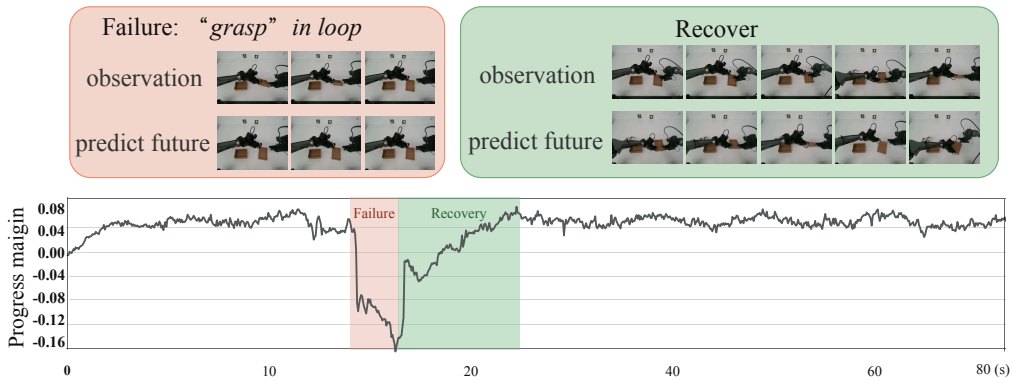


Figure 6: **Qualitative analysis of progress evaluation in SYSTEM 3.** The top panels show observed frames and future observations predicted from the corresponding current context. During the failure window, the robot remains in a repeated grasping loop and the progress margin decreases sharply. After recovery, the predicted future returns to a task-progressing trajectory and the progress margin increases. Shaded regions indicate the failure and recovery windows.

I.2 Qualitative OOD Generalization of the Pretrained Foundation Policy

We evaluate the same pretrained checkpoint without task-specific post-training. The OOD cloth-folding setting changes both the cloth instance and its initial configuration. According to the train–test isolation protocol, the evaluated cloth instances do not have exact matches in the training corpus.

Figure 7 presents representative executions on AgileX PiPER and Maker H01. On PiPER, the policy interacts with different cloth instances, establishes contact, reorients the deformable object, and places it into the container. On H01, the policy performs a comparable multi-stage folding sequence with a different embodiment and gripper configuration. These rollouts provide qualitative evidence that the pretrained policy transfers reusable interaction patterns beyond memorized object appearances or fixed initial states.

At the same time, the two embodiments exhibit different execution behaviors. This contrast is consistent with the quantitative OOD results and highlights the sensitivity of deformable-object manipulation to embodiment morphology, gripper geometry, and compliance. The qualitative examples therefore illustrate both the transferability of the learned manipulation primitives and the remaining challenges in cross-embodiment foundation-policy transfer.

I.3 Qualitative Post-Training Results

We further visualize representative executions after task-specific post-training. The selected examples cover language-conditioned manipulation and complex multi-stage interaction across AgileX PiPER and Maker H01.

Language-conditioned manipulation. Figures 8 and 9 illustrate three forms of instruction grounding: color-based target selection, object-identity grounding, and relative spatial reasoning. On PiPER, the policy follows the instruction "Grab the orange spoon and put it on the plate" and transfers the block on the left side of the plate into the bowl. On H01, the policy selects the language-specified colored target, moves the fork to the basket, and resolves the relative spatial instruction referring to the fork on the left side of the plate.

These rollouts show that post-training changes the selected object and subsequent action sequence according to the instruction, rather than merely replaying a fixed manipulation trajectory. The same qualitative behavior is observed across the two embodiments despite their different camera views, kinematics, and action spaces.

Complex manipulation. Figures 10 and 11 show tasks that require sustained contact, deformable-object handling, tool-mediated interaction, and multi-stage execution. On PiPER, the policy maintains

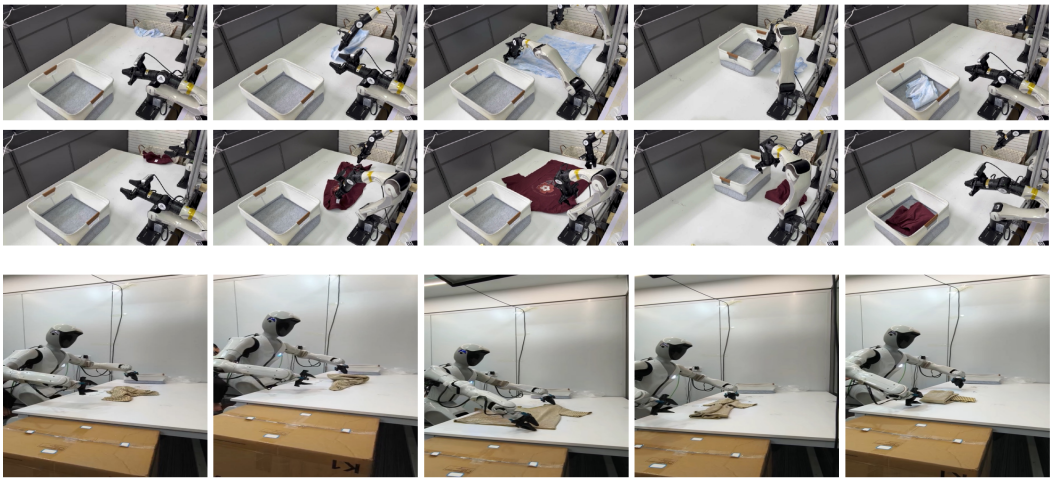


Figure 7: **OOD cloth-folding with the pretrained foundation policy.** The top panel shows two representative AgileX PiPER rollouts with different cloth instances, and the bottom panel shows a representative Maker H01 rollout. Each row contains temporally ordered observations. The evaluated cloth are initialized in unstructured configurations and are not exact matches to the training instances.

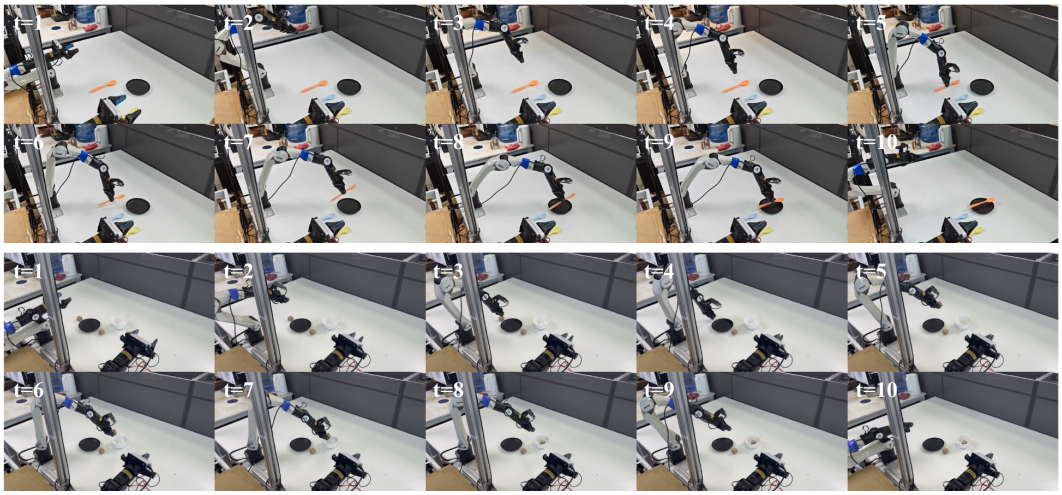


Figure 8: **Post-training language following on AgileX PiPER.** The top rollout demonstrates color-conditioned object selection for "Grab the orange spoon and put it on the plate." The bottom rollout demonstrates relative spatial grounding for "Grab the block on the left side of the plate and put it in the bowl."

contact while manipulating play-dough and uses a sweeping tool to gather scattered material. On H01, the policy performs sequential household-item organization and repeated block sorting, requiring object selection, transport, placement, and adaptation as the scene changes.

The examples are consistent with the quantitative evaluation: the policy shows strong behavior on several deformable-object and household-organization tasks, while sweeping and sorting remain more challenging. Overall, the rollouts suggest that post-training improves sustained interaction and multi-stage task execution across heterogeneous embodiments.

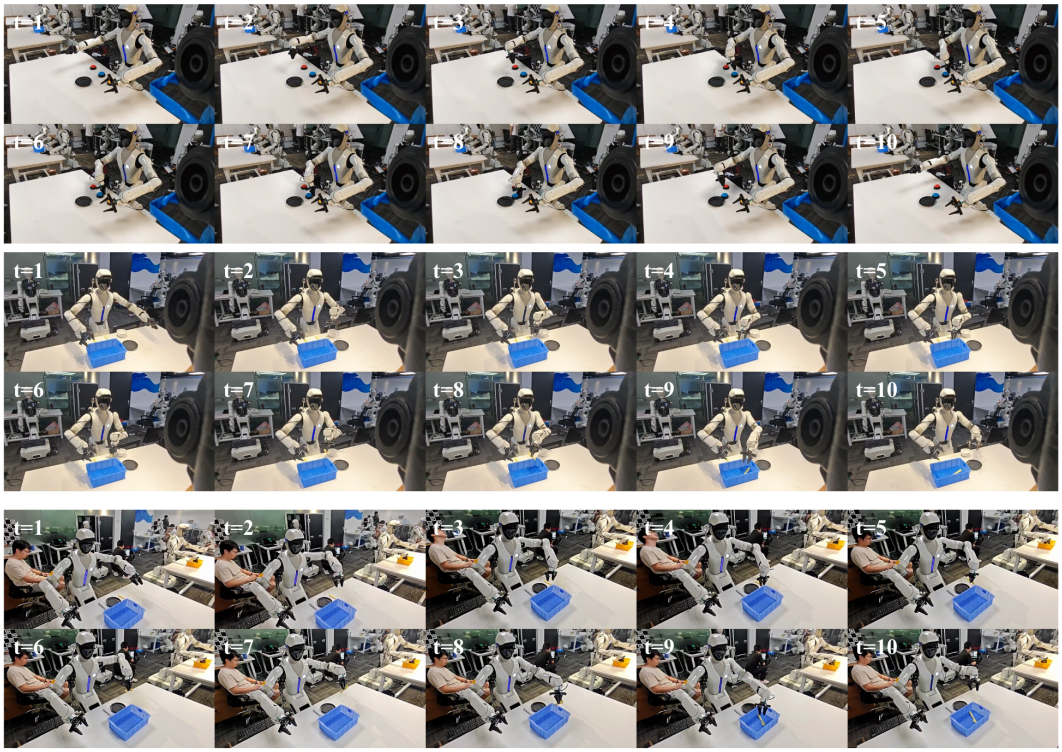


Figure 9: **Post-training language following on Maker H01.** The top rollout demonstrates color-conditioned target selection. The middle rollout follows the instruction "Pick up the fork and move it to the basket." The bottom rollout resolves the relative spatial instruction "Grab the fork on the left side of the plate and put it in the basket."

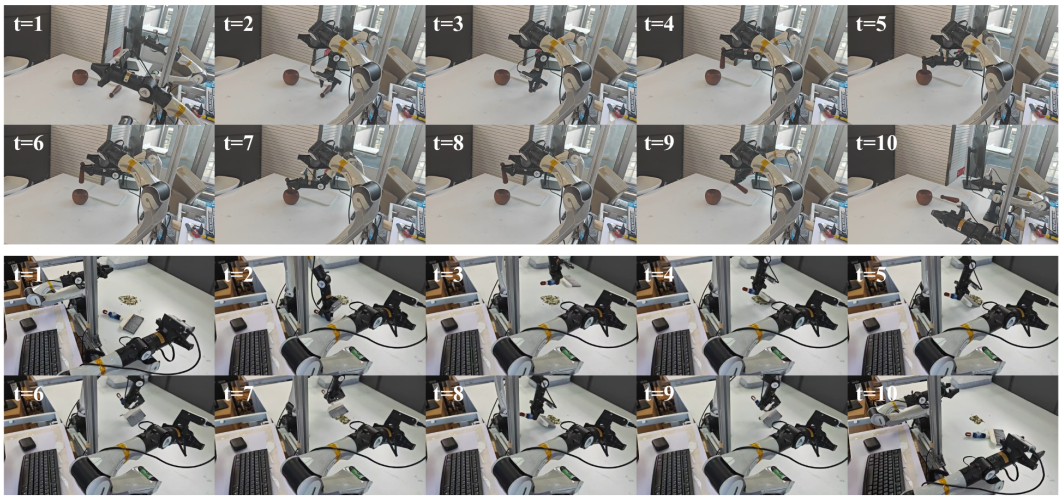


Figure 10: **Post-training complex manipulation on AgileX PiPER.** The top rollout shows deformable-object manipulation with play-dough. The bottom rollout shows tabletop sweeping, which requires sustained tool–object contact and coordinated motion over multiple stages.

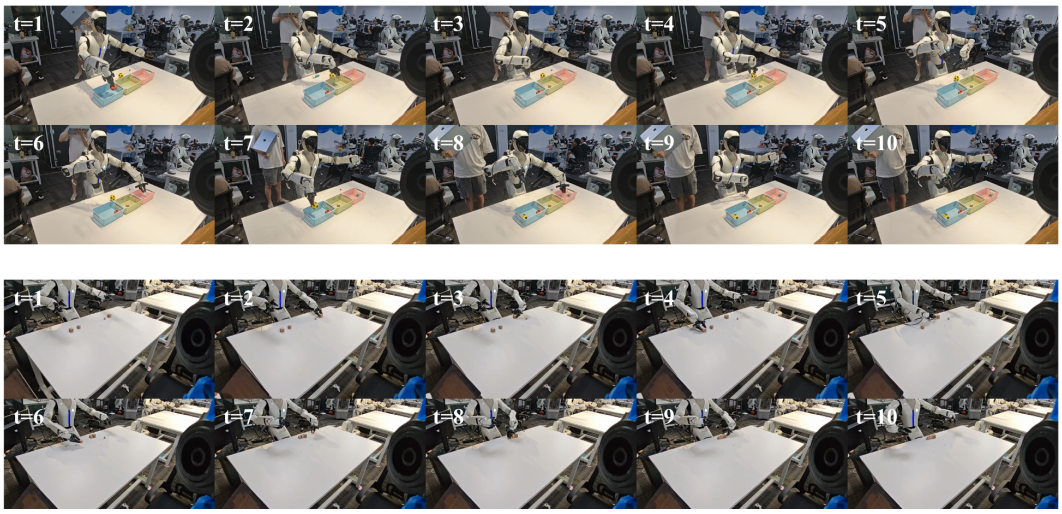


Figure 11: **Post-training complex manipulation on Maker H01.** The top rollout shows household-item organization involving sequential object selection and placement. The bottom rollout shows repeated block sorting across successive manipulation stages.