DEVOL-ONE: ONE AUTOREGRESSIVE MIXTURE OF TRANSFORMERS TO UNIFY VISION-LANGUAGE-ACTION AND LATENT WORLD MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision Language Action (VLA) models condition actions directly on current visual and language context, without an explicit account of how the scene evolves under candidate actions. World Action Models (WAM) attempt to address this limitation by predicting future states, but existing designs keep prediction and policy learning architecturally separate, connecting them only through the predicted output, whether through pixel space video generation or a latent forecasting module trained independently of the policy. We present Devol-ONE, a Mixture of Transformers architecture that unifies vision language understanding, latent world dynamics prediction, and action generation within a single autoregressive framework. Instead of encoding vision language tokens once and feeding them to the action expert, Devol-ONE runs autoregressive prediction jointly across a vision language stream and a V-JEPA pretrained dynamics stream, attending to the vision language key-value cache at every layer to forecast future latent states under language guidance. The action expert is in turn shaped continuously by semantic reasoning and predicted physical dynamics rather than by a fixed representation computed in advance. Extensive experiments are conducted on LIBERO, LIBERO-PLUS, RoboTwin2.0 along with real-world evaluation on Flexiv single-arm and dual-arm setups. Ablation studies show the effectiveness of dynamic stream prediction and layer-wise unified attention to validate our model architectural coherency.

1 INTRODUCTION

Imitation learning (IL) is widely used to train robotic manipulation policies. Among existing approaches, Vision-Language-Action (VLA) models (Bu et al., 2024; Hu et al., 2024; Li et al., 2024a;b; 2023; Black et al., 2024; Intelligence et al., 2025) benefit from large-scale pretraining on visual observations, language prompts, and demonstration trajectories, supporting instruction following across multiple tasks. However, conventional VLAs (Lin et al., 2025a;b) encode the scene once and map the resulting representation directly to control, without modeling how the environment will evolve in response to an action.

World-Action Models (WAMs) incorporate predicted future states into action generation (Shen et al., 2026). Existing designs can be broadly grouped into two categories. Cascaded designs synthesize future frames or subgoals, then recover actions through a separate inverse-dynamics or goal-conditioned module (Du et al., 2024; 2023; Black et al., 2023; Bu et al., 2024; Tian et al., 2024). Joint designs denoise future video tokens and action tokens within a single backbone (Li et al., 2025c; Zhu et al., 2025; Cen et al., 2025b; Cheang et al., 2024). These approaches benefits from internet-scale video generation models that provide transferable dynamics which static image-text pretraining cannot supply and the prediction target that can be supervised with action-free video, allowing the training signal to scale beyond limited action labels. However, in WAMs, generating future observations in pixel space causes substantial inference costs, as well as lacking explicit text reasoning. On the other hand, even though VLA has its own vision-text semantic space, it does not take real-world physical environment into account.

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.

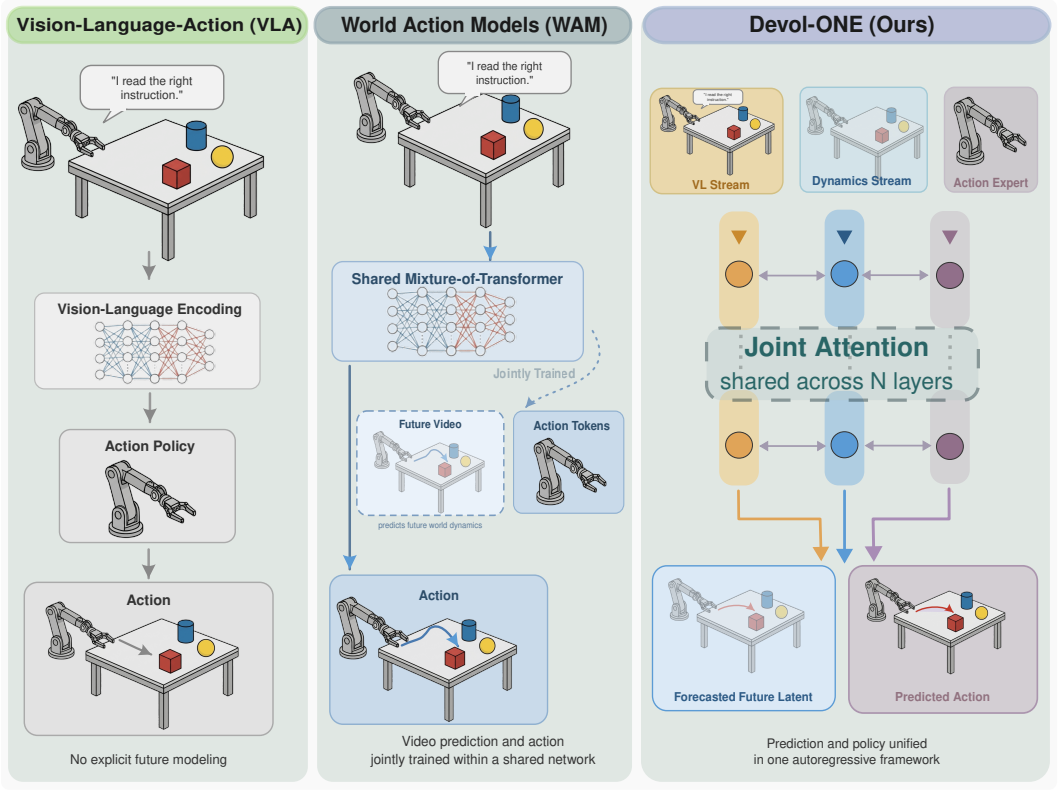


Figure 1: Comparison of VLA models, joint WAMs, and Devol-ONE. (**Left**) Standard VLA models map vision-language features directly to actions. (**Middle**) Joint WAMs use a shared transformer to predict future video and actions. (**Right**) Devol-ONE combines a VL stream, a JEPA dynamics stream initialized from a V-JEPA pretrained backbone, and an action expert through layerwise joint attention.

Can an embodied policy jointly reason about its task, predict future world states, and generate actions within a unified architecture? We present Devol-ONE, a Mixture-of-Transformers (MoT) architecture that integrates latent dynamics prediction with vision-language understanding and action generation. As illustrated in Figure 1, Devol-ONE decomposes computation into three parameter-disjoint streams: a VL stream providing task-related semantic features, a dynamics stream initialized from a V-JEPA pretrained backbone (Assran et al., 2025) that autoregressively forecasts future latent states $\hat{z}_{t+1}, \ldots, \hat{z}_{t+N}$ under language guidance, and a diffusion action expert that denoises continuous action chunks. In general, a Layerwise Joint MoT Block connects the streams at every transformer layer, allowing the action expert to attend to both semantic features and predicted dynamics throughout action decoding.

In VLA-JEPA (Sun et al., 2026), a related latent world model, the target latent is constructed from a ground-truth future frame and is therefore available only during training. At inference, the robot has no access to future observations and leave no choices but roll out its own latent predictions, conditioning only on the current state and image. This train–inference mismatch is related to exposure bias in sequence prediction (Bengio et al., 2015), as inference relies on model-generated latent states. We introduce Autoregressive Latent Rollout Training (ALRT), which exposes both the predictor and the action expert to model-generated multi-step latent rollouts during training.

We evaluate Devol-ONE on LIBERO (Liu et al., 2023), LIBERO-Plus (Fei et al., 2025), and RoboTwin 2.0 (Mu et al., 2025), together with real-world deployment on Flexiv single-arm and dual-arm platforms. Devol-ONE attains 98.4% average success on LIBERO and 71.4% on zero-shot LIBERO-Plus without fine-tuning on the perturbed data. On the 34 evaluated RoboTwin 2.0 tasks, the average success rate increases from 61.1% under the Clean setting to 63.7% under the

Randomized setting. These results suggest improved robustness under the evaluated perturbations. Our contributions are summarized as follows:

- **A unified autoregressive architecture.** Devol-ONE couples a vision-language stream, a V-JEPA latent dynamics stream, and a diffusion action expert through layerwise joint attention, grounding future-latent forecasting and action generation in a shared computation rather than connecting them through a predicted output.
- **Rollout training for the policy, not only the predictor.** To solve exposure bias in latent world models propagates to any action head conditioned on the rollout, and introduce ALRT, which trains both the dynamics stream and the action expert on self-generated K-step latent trajectories.
- **Validation under distribution shift and in the real world.** We evaluate Devol-ONE on simulation benchmarks, including zero-shot LIBERO-Plus and RoboTwin 2.0 under randomization, and compare its robustness with that of pixel-space WAMs. We further evaluate Devol-ONE on Flexiv single-arm and dual-arm platforms.

2 Related Work

Vision-Language-Action Models. Vision-Language-Action (VLA) models (Brohan et al., 2023; Zitkovich et al., 2023; Li et al., 2023; 2024b;a; Black et al., 2024; Intelligence et al., 2025; Liu et al., 2024; 2025a; Shukor et al., 2025; Bjorck et al., 2025; Wen et al., 2025; Zheng et al., 2025a) adapt pretrained vision-language backbones to robotic control by learning $p(a_{1:H} \mid o, l)$ from demonstration trajectories. Lightweight variants further emphasize computational efficiency (Lin et al., 2025a;b; 2026). Devol-ONE extends this formulation by incorporating predicted latent dynamics into action generation through layerwise joint attention.

World-Action Models. World-Action Models (WAMs) use predictions of future visual states to guide action generation (Wu et al., 2024; Bu et al., 2024; Hu et al., 2024; Li et al., 2025c; Cheang et al., 2024; Black et al., 2023; Du et al., 2024; Yang et al., 2023; Du et al., 2023; Zhu et al., 2024; Cen et al., 2025b; Zhu et al., 2025; Jang et al., 2025; Feng et al., 2025; Bharadhwaj et al., 2024; Huang et al., 2025a; Zhang et al., 2025b). A recent survey organizes this space by how much of the future a method is required to generate before the action is obtained (Shen et al., 2026). Cascaded WAMs follow an imagine-then-execute paradigm, introducing future visual observations $v_{1:T}$ as an intermediate variable:

$$p(a_{1:H} \mid o, l) = \int p(v_{1:T} \mid o, l)\, p(a_{1:H} \mid o, l, v_{1:T})\, dv_{1:T}, \tag{1}$$

first synthesizing future visual states then recovering action trajectories from these imagined outcomes via an inverse dynamics model (Tian et al., 2024). The predicted visual states provide future context for action selection. However, their strong dependency on future frame synthesis introduces considerable inference latency and potential error accumulation, motivating the latent alternatives discussed next.

Implicit World Modeling and Latent Action Representations. A growing body of work retains the benefits of dynamics-aware policy learning while avoiding explicit pixel-level future synthesis, encoding predicted futures as compact latent representations rather than rendered frames (Yuan et al., 2026; Huang et al., 2025b; Sun et al., 2026). One line augments the VLA objective with auxiliary future-prediction targets, such as multimodal world knowledge (Zhang et al., 2025d), aligned latent features of future observations (Zheng et al., 2025b), joint understanding-and-prediction pretraining (Zhang et al., 2025c), or visual and motion chain-of-thought reasoning (Zhao et al., 2025; Zheng et al., 2024; Zhong et al., 2025; 2026). A parallel line learns action-relevant latent dynamics directly from large-scale action-free video via latent action models (Schmidt & Jiang; Edwards et al., 2019; Ye et al., 2024; Chen et al., 2024b;a; Bu et al., 2025a; Chen et al., 2025; Yang et al., 2025; Liu et al., 2025b; Bi et al., 2025; Gao et al.; Zhang et al., 2026; Klepach et al., 2025; Nikulin et al., 2026; 2025; Garrido et al., 2026; Zhang et al., 2025a; Bu et al., 2025b), avoiding paired action labels. Our approach focuses on coupling latent dynamics prediction and action generation within the policy through layerwise joint attention.

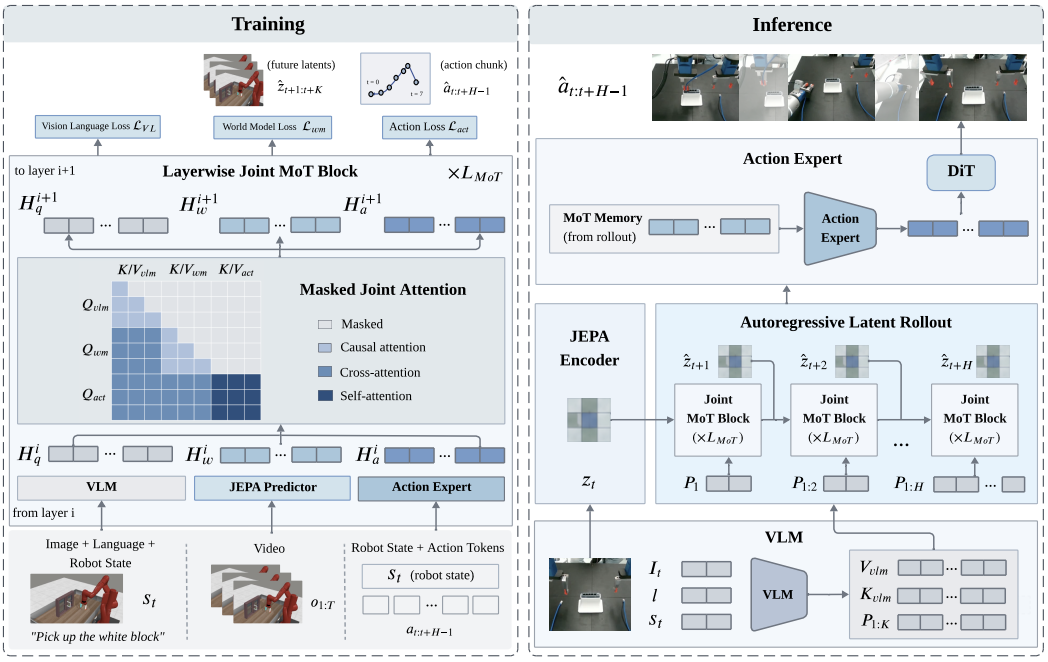


Figure 2: **Overview of Devol-ONE pipeline.** A VLM, World Model (WM), and Action Expert are trained as unified layerwise Mixture-of-Transformers blocks with masked joint attention, optimized with a world-model loss $\mathcal{L}_{\text{dyn}}$ and an action loss $\mathcal{L}_{\text{act}}$ (**left**). At inference (**right**), a JEPA encoder embeds the current observation, and the predictor is autoregressively rolled out through the shared MoT block to forecast future latents conditioned on the VLM. The resulting latent trajectory conditions the Action Expert, which uses a DiT head to generate the action chunk.

3 METHOD

Overview. Fig. 2 shows our model overall architecture. We formulate embodied manipulation as joint conditional generation over future latent dynamics and action chunks. At time step t, the policy receives observation $o_t = (I_t, c_t, S_t)$, where I_t is the RGB image, c_t is the language instruction, and S_t is the proprioceptive state. Devol-ONE learns two objectives as shown in below.

The action policy generates a continuous action chunk $A_t = (a_t, \ldots, a_{t+H-1})$ of horizon H from the current observation:

$$p(A_t \mid o_t). \tag{2}$$

The JEPA world model learns an autoregressive predictor over V-JEPA (Bardes et al., 2023; Assran et al., 2025) latents. Given latent history $Z_{t-L:t} = (z_{t-L}, \ldots, z_t)$ of $L{+}1$ steps encoded from past observations, the predictor is trained to predict one step ahead, i.e., $(z_{t-L+1}, \ldots, z_{t+1})$. At inference, it is rolled out for K steps using its own predictions as history to forecast future latents $Z_{t+1:t+K} = (z_{t+1}, \ldots, z_{t+K})$:

$$p(Z_{t+1:t+K} \mid Z_{t-L:t}, o_t). \tag{3}$$

Both objectives are jointly optimized, coupling action generation with latent dynamics prediction.

Layerwise Joint Mixture-of-Transformers with Directed Causal Attention. Let X^{VL}, X^{D}, and X^{A} denote the token sequences of the vision-language stream, the latent dynamics stream, and the action stream, respectively. Following the MoT formulation, every non-embedding parameter is decoupled by stream, and each stream $s \in \{\text{VL}, \text{D}, \text{A}\}$ maintains its own layer normalization, query/key/value projections, and feed-forward network. At layer ℓ, the projections are computed as

$$Q_s^{(\ell)} = \bar{X}_s^{(\ell)} W_{Q,s}^{(\ell)}, \qquad K_s^{(\ell)} = \bar{X}_s^{(\ell)} W_{K,s}^{(\ell)}, \qquad V_s^{(\ell)} = \bar{X}_s^{(\ell)} W_{V,s}^{(\ell)}, \tag{4}$$

where $\bar{X}_s^{(\ell)} = \text{LN}_s^{(\ell)}(X_s^{(\ell)})$, and d denotes the per-head dimension used for scaling below.

The three streams interact through a fixed directed dependency: the vision-language stream conditions the dynamics stream, and both condition the action stream. Later streams access earlier ones, whereas earlier streams do not access later ones. Specifically, the VL stream performs causal self-attention only, the dynamics stream attends to the concatenation of VL and dynamics tokens, and the action stream attends to the concatenation of VL, dynamics, and action tokens:

$$\tilde{X}^{(\mathrm{VL})} = \mathrm{Softmax}\left(\frac{Q^{(\mathrm{VL})}K^{(\mathrm{VL})\top}}{\sqrt{d}} + M^{(\mathrm{VL})}\right)V^{(\mathrm{VL})}, \tag{5}$$

$$\tilde{X}^{(\mathrm{D})} = \mathrm{Softmax}\left(\frac{Q^{(\mathrm{D})}[K^{(\mathrm{VL})}; K^{(\mathrm{D})}]^\top}{\sqrt{d}} + M^{(\mathrm{D})}\right)[V^{(\mathrm{VL})}; V^{(\mathrm{D})}], \tag{6}$$

$$\tilde{X}^{(\mathrm{A})} = \mathrm{Softmax}\left(\frac{Q^{(\mathrm{A})}[K^{(\mathrm{VL})}; K^{(\mathrm{D})}; K^{(\mathrm{A})}]^\top}{\sqrt{d}} + M^{(\mathrm{A})}\right)[V^{(\mathrm{VL})}; V^{(\mathrm{D})}; V^{(\mathrm{A})}], \tag{7}$$

where $[;]$ denotes concatenation along the token dimension, and the layer index ℓ is omitted for clarity. Masks $M^{(\mathrm{VL})}$ and $M^{(\mathrm{D})}$ are causal and enforce temporal ordering, while $M^{(\mathrm{A})}$ enforces the directed stream dependency; attention within the action stream is bidirectional over the action chunk. In cross-stream attention, an earlier-stream key is visible only when its temporal index does not exceed that of the query. Thus, VL accesses only its causal history, dynamics accesses the VL context and its own history, and action accesses the VL context, dynamics context, and its own action history. No stream accesses a later stream or a future rollout step.

The language instruction specifies what task the model should perform. Since the dynamics stream attends to VL keys and values at every layer, this conditioning is applied at each rollout step and helps keep the predicted trajectory aligned with the task. Each sublayer is applied residually within its own stream,

$$\hat{X}_s^{(\ell)} = X_s^{(\ell)} + \tilde{X}_s^{(\ell)}, \qquad X_s^{(\ell+1)} = \hat{X}_s^{(\ell)} + \mathrm{FFN}_s^{(\ell)}\left(\mathrm{LN}_s^{(\ell)}\left(\hat{X}_s^{(\ell)}\right)\right). \tag{8}$$

With stream-specific normalization, projection, and FFN parameters, and cross-stream communication mediated solely by the directed causal attention above, the model preserves specialization of vision-language reasoning, latent dynamics, and action generation while coupling them through shared layerwise computation.

Training time. The three streams are trained jointly with stream-specific objectives. The VL stream performs autoregressive next-token prediction over text tokens conditioned on the image with a causal mask, optimized by cross-entropy:

$$\mathcal{L}_{\mathrm{VL}} = -\sum_{i \in \mathrm{text}} \log p_\theta\left(x_i^{\mathrm{VL}} \mid x_{<i}^{\mathrm{VL}}, I_t\right). \tag{9}$$

The JEPA dynamics stream performs causal latent autoregressive prediction. Given the current latent z_t and its causal history, it predicts future latents $\hat{z}_{t+1}, \ldots, \hat{z}_{t+K}$ and is trained with an L1 loss against targets from a frozen V-JEPA encoder E:

$$\mathcal{L}_{\mathrm{dyn}} = \sum_{k=1}^{K} \left\|\hat{z}_{t+k} - E(I_{t+k})\right\|_1, \tag{10}$$

The action expert is a Diffusion Transformer (DiT) (Peebles & Xie, 2023) with bidirectional self-attention over action tokens. It denoises a noisy action chunk A_t^τ conditioned on the VL representations, predicted latent rollout, and proprioceptive state, trained with a denoising objective:

$$\mathcal{L}_{\mathrm{act}} = \mathbb{E}_{\epsilon,\tau}\left[\left\|\epsilon - \epsilon_\theta\left(A_t^\tau, \tau, X^{\mathrm{VL}}, \hat{Z}_{t+1:t+K}, S_t\right)\right\|_2^2\right], \tag{11}$$

where A_t^τ is the noisy action chunk at diffusion step τ. The overall training objective combines the three losses:

$$\mathcal{L} = \mathcal{L}_{\mathrm{VL}} + \lambda_{\mathrm{dyn}}\mathcal{L}_{\mathrm{dyn}} + \lambda_{\mathrm{act}}\mathcal{L}_{\mathrm{act}}, \tag{12}$$

with weights λ_{dyn} and λ_{act} reported in the Appendix.

Autoregressive Latent Rollout Training. The dynamics targets above come from encoded future frames, whereas at inference the model has only its own predictions. Under teacher forcing, the

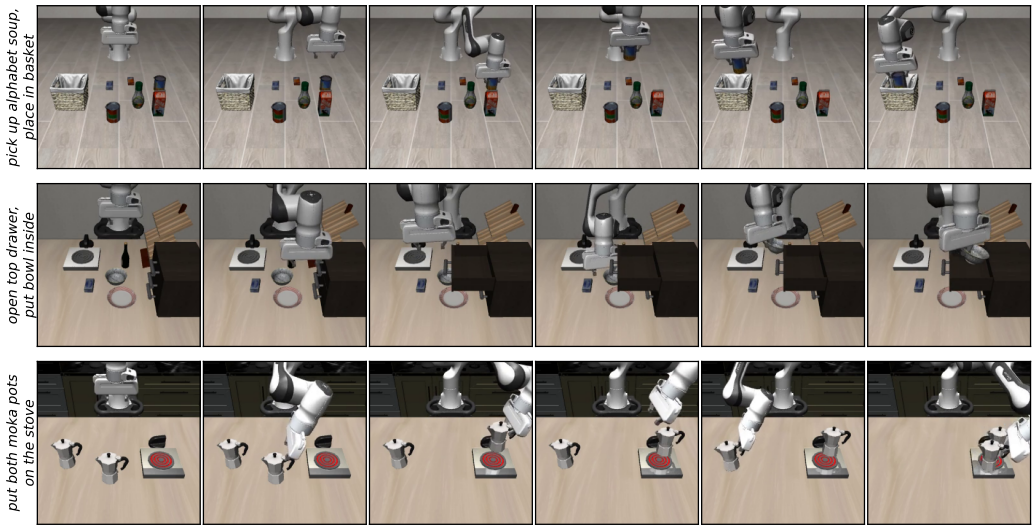


Figure 3: Example rollouts on LIBERO.

predictor never sees the latent distribution it encounters in closed loop, and the action expert inherits this mismatch through its conditioning on the rollout. ALRT addresses this gap by unrolling the dynamics stream during training as at inference. Training begins with a teacher-forced warm-up, in which each step receives the encoded target latent, and then switches to a free-running mode, in which the predictor consumes its own output at every step:

$$\tilde{z}_{t+k} = \begin{cases} E(I_{t+k}), & \text{warm-up,} \\ \hat{z}_{t+k}, & \text{free-running.} \end{cases} \tag{13}$$

Both $\mathcal{L}_{\text{dyn}}$ and $\mathcal{L}_{\text{act}}$ are evaluated on the self-generated trajectory after the switch. The transition is discrete rather than a sampling schedule (Bengio et al., 2015); thus training and inference rollouts match in the free-running phase.

Inference time. At inference, the dynamics stream is rolled out as above, starting from the latent z_t encoded from the current observation and conditioned on X^{VL}. The resulting latent trajectory $\hat{Z}_{t+1:t+K}$ conditions the action expert, which denoises an action chunk A_t through reverse diffusion. The robot executes the first $H_{\text{exec}} \leq H$ actions of the chunk and replans at the next step. Forecasting is performed in V-JEPA latent space, so inference requires neither future frame synthesis nor an inverse dynamics model.

Optional CoT branch. In addition to the JEPA predictor, the vision-language backbone supports inference-time autoregressive rollout, producing chain-of-thought text alongside the action. This branch annotates each step with the current subtask and makes task stages explicit, for dataset construction and failure analysis. It is optional and remains disabled in all reported results unless stated otherwise. Since the VL key-value cache is shared by the dynamics stream and the action expert, we either prefill all language tokens or decode them from the prompt, feeding generated tokens back into the model at the next autoregressive step.

4 Experiments

4.1 Experimental Setup

LIBERO & LIBERO-Plus. We continue training Devol-ONE from a robot-pretrained checkpoint. We evaluate the resulting checkpoint on the standard four-suite LIBERO benchmark (Liu et al., 2023). On LIBERO-Plus (Fei et al., 2025) we evaluate the checkpoint under two protocols. In the zero-shot setting, the checkpoint is evaluated directly on LIBERO-Plus without fine-tuning on its perturbed data. In the in-distribution (zero-shot) setting, we fine-tune on the LIBERO-Plus training

Table 1: **Success rates (%) on the LIBERO benchmark.**

Model	Spatial	Object	Goal	Long	Avg
Diffusion Policy (Chi et al., 2023)	78.3	92.5	68.3	50.5	72.4
Octo (Team et al., 2024)	78.9	85.7	84.6	51.1	75.1
OpenVLA (Kim et al., 2024)	84.7	88.4	79.2	53.7	76.5
SpatialVLA (Qu et al., 2025)	88.2	89.9	78.6	55.5	78.1
CoT-VLA (Zhao et al., 2025)	87.5	91.6	87.6	69.0	81.1
WorldVLA (512) (Cen et al., 2025b)	87.6	96.2	83.4	60.0	81.8
VLA-RFT (Li et al., 2025b) [1]	94.4	94.4	95.4	80.2	91.1
RynnVLA-002 (Disc.) (Cen et al., 2025a)[1]	94.2	96.8	94.6	87.6	93.3
GR00T N1 (Bjorck et al., 2025)	94.4	97.6	93.0	90.6	93.9
π_0 (Black et al., 2024)	96.8	98.8	95.8	85.2	94.2
UniVLA (Bu et al., 2025a)	96.5	96.8	95.6	92.0	95.2
$\pi_{0.5}$ (Intelligence et al., 2025)	98.8	98.2	98.0	92.4	96.9
OpenVLA-OFT (Kim et al., 2025)	97.6	98.4	97.9	94.5	97.1
VLA-JEPA (Sun et al., 2026)	96.2	99.6	97.2	95.8	97.2
RynnVLA-002 (Cont.) (Cen et al., 2025a)	**99.0**	**99.8**	96.4	94.4	97.4
Fast-WAM (Yuan et al., 2026)	97.0	98.2	**100.0**	95.2	97.6
Devol-ONE (Ours)	98.4	99.6	98.2	**97.0**	**98.4**

split before evaluation. Both protocols evaluate across seven perturbation dimensions: Camera, Robot, Language, Light, Background, Noise, and Layout, each spanning the four LIBERO task suites. **RoboTwin 2.0.** RoboTwin 2.0 (Mu et al., 2025) is a benchmark for bimanual manipulation with more than 50 tasks that require coordinated dual-arm control.

Real-world benchmarks. We evaluate single- and dual-arm manipulation on a Flexiv platform using human-teleoperated demonstrations. Each baseline is trained separately for each task. The task definitions and evaluation procedure are described below.

Additional training and evaluation details, including training schedules and prompt configurations, are provided in Appendix A.4.

4.2 Simulation Results

LIBERO. Table 1 compares Devol-ONE with published policies on the four LIBERO suites. Devol-ONE achieves an average success rate of 98.4%, the highest among the compared methods. On the Long suite, it reaches 97.0%, compared with 95.2% for Fast-WAM, 95.8% for VLA-JEPA, and 94.5% for OpenVLA-OFT. It also achieves success rates of 98.4% on Spatial, 99.6% on Object, and 98.2% on Goal. Figure 3 shows example rollouts.

LIBERO-Plus. Table 2 reports zero-shot results without fine-tuning on perturbed data. Devol-ONE achieves an overall success rate of 71.4%, the highest among the compared methods. It leads on Language, Light, and Layout, with success rates of 85.9%, 94.5%, and 80.5%, respectively.

RoboTwin 2.0. We compare against results from the RoboTwin 2.0 leaderboard (RoboTwin Team, 2025). Table 3 presents four representative dual-arm tasks under the Clean and Randomized settings. Devol-ONE achieves the highest Randomized success rate on all four tasks: 99.0% on Click Alarmclock, 93.0% on Dump Bin Bigbin, 68.5% on Place Bread Basket, and 67.6% on Place Can Basket. Fast-WAM achieves higher success rates under the Clean setting on three of the four tasks, but its success rates decrease to 60.0%, 0.0%, 0.0%, and 0.0%, respectively, under the Randomized setting. Per-task Clean/Randomized results are provided in Table 8.

[1] From public reports.

Table 2: **Zero-shot performance on LIBERO-Plus.** All methods are trained only on the standard LIBERO dataset without fine-tuning on LIBERO-Plus dataset.

Method	Camera	Robot	Language	Light	Background	Noise	Layout	Total
OpenVLA (Kim et al., 2024)	0.8	3.5	23.0	8.1	34.8	15.2	28.5	15.6
WorldVLA (Cen et al., 2025b)	0.1	27.9	41.6	43.7	17.1	10.9	38.0	25.0
NORA	2.2	37.0	65.1	45.7	58.6	12.8	62.1	39.0
UniVLA (Bu et al., 2025a)	1.8	46.2	69.6	69.0	81.0	21.2	31.9	42.9
Fast-WAM (Yuan et al., 2026)	44.5	**68.9**	60.7	53.7	37.7	16.4	78.2	51.5
π_0 (Black et al., 2024)	13.8	6.0	58.8	85.0	81.4	**79.0**	68.9	53.6
OpenVLA-OFT_w (Kim et al., 2025)	10.4	38.7	70.5	76.8	**93.6**	49.9	69.9	55.8
π_0-Fast[2] (Pertsch et al., 2025)	**65.1**	21.6	61.0	73.2	73.2	74.4	68.8	61.6
OpenVLA-OFT_m (Kim et al., 2025)	55.6	21.7	81.0	92.7	91.0	78.6	68.7	67.9
RIPT-VLA	55.2	31.2	77.6	88.4	91.6	73.5	74.2	68.4
OpenVLA-OFT (Kim et al., 2025)	56.4	31.9	79.5	88.7	93.3	75.8	74.2	69.6
Devol-ONE (Ours)	47.3	45.7	**85.9**	**94.5**	91.8	67.5	**80.5**	**71.4**

Table 3: Simulation results on the RoboTwin 2.0 leaderboard. We report success rates (%) under *Clean* and *Randomized* settings for four representative tasks.

	Click Alarmclock		Dump Bin Bigbin		Place Bread Basket		Place Can Basket	
RoboTwin	Clean	Randomized	Clean	Randomized	Clean	Randomized	Clean	Randomized
ACT (Zhao et al., 2023)	32.0	4.0	68.0	1.0	6.0	0.0	1.0	0.0
Diffusion Policy (Chi et al., 2023)	61.0	5.0	49.0	0.0	14.0	0.0	18.0	0.0
RDT (Liu et al., 2024)	61.0	12.0	64.0	32.0	10.0	2.0	19.0	6.0
π_0 (Black et al., 2024)	63.0	11.0	83.0	24.0	17.0	4.0	41.0	5.0
DP3 (Ze et al., 2024)	77.0	14.0	85.0	53.0	26.0	1.0	67.0	2.0
starVLA (StarVLA Community, 2026)	60.0	0.0	54.0	4.0	82.0	5.0	4.0	0.0
$\pi_{0.5}$ (Intelligence et al., 2025)	65.0	50.0	92.0	84.0	63.0	60.0	56.0	10.0
Abot-M0 (Yang et al., 2026b)	0.0	21.0	89.0	68.0	70.0	47.0	38.0	13.0
Xiaomi Robotics-0 (Cai et al., 2026b)	97.0	58.0	94.0	49.0	54.0	23.0	52.0	7.0
EventVLA (Yang et al., 2026a)	93.0	30.0	92.0	64.0	66.0	15.0	76.0	7.0
X-WAM (Guo et al., 2026)	95.0	61.0	88.0	40.0	82.0	40.0	63.0	15.0
X-VLA (Zheng et al., 2025a)	62.0	43.0	83.0	41.0	85.0	52.0	70.0	9.0
GalaxeaVLA (Liu et al., 2026)	79.0	7.0	88.0	40.0	77.0	32.0	67.0	6.0
AHA-WAM (Cai et al., 2026a)	98.0	17.0	90.0	27.0	54.0	3.0	61.0	2.0
Spatial Forcing (Li et al., 2025a)	89.0	28.0	83.0	70.0	84.0	53.0	**77.0**	10.0
Fast-WAM (Yuan et al., 2026)	**99.0**	60.0	94.0	0.0	**87.0**	0.0	77.0	0.0
GigaBrain-0.7 (Team et al., 2026)	98.0	96.0	90.0	66.0	60.0	68.0	20.0	22.0
Devol-ONE (Ours)	98.0	**99.0**	**95.0**	**93.0**	59.0	**68.5**	58.0	**67.6**

Table 4: **Flexiv manipulation tasks and demonstration counts.** Abbreviations are used in Table 7. Details on the task and task criteria are listed in Appendix A.5.1

Abbreviation	Task prompt (verbatim)	Episodes
Ethernet	Insert the Ethernet connector	118
2arm box	Stack realsense boxes with both arms	200
1arm box	Stack realsense boxes with only the left arm	200
Basket	Place the snacks into a basket	200
4cups	Stack four different-colored cups	200

4.3 Real-World Evaluation

We collect human-teleoperated demonstrations on a fixed Flexiv manipulation platform with two Flexiv Rizon (Flexiv Robotics, 2024) 7-DoF collaborative arms and three Intel RealSense D-series (Keselman et al., 2017) color cameras. One camera is mounted between the arms, and one is mounted on each wrist to capture multiple views of the workspace. Demonstrations are collected using a Meta Quest VR controller, with each episode starting from the same home position. Task prompts and demonstration counts are listed in Table 4. We train Devol-ONE and the baseline models on these demonstrations, evaluate each model over 20 rollout episodes, and record the success rate. Table 7 reports the success rates. Devol-ONE achieves the highest average success rate across

[2]FAST-tokenized variant.

Table 5: Action expert conditioning on LIBERO.

Conditioning	Spatial	Object	Goal	Long	Avg
VL only	96.2	98.4	96.8	92.5	96.0
JEPA only	97.1	98.8	97.5	94.2	96.9
Final-layer fusion	**98.4**	99.0	**98.6**	92.1	97.0
Layerwise (Ours)	**98.4**	**99.6**	98.2	**97.0**	**98.4**

Table 6: Effect of prediction horizon on LIBERO.

Steps N	Frames	Long	Avg
1	4	94.5	96.8
2	6	96.2	97.5
3	**8**	**97.0**	**98.4**

tasks, outperforming all baselines including $\pi_{0.5}$ and π_0, with particularly strong margins on the Basket and 4cups tasks. Further details can be referred to Appendix A.5.

Table 7: **Real-world rollout success rates (%).** Full task prompts are listed in Table 4.

Task	**Ethernet**	**2arm box**	**1arm box**	**Basket**	**4cups**	**Avg.**
GigaBrain-0.7Team et al. (2026)	0.0	80.0	90.0	20.0	70.0	52.0
DM0.5Yu et al. (2026)	0.0	40.0	45.0	25.0	20.0	26.0
Fast-WAMYuan et al. (2026)	0.0	55.0	60.0	0.0	0.0	23.0
π_0Black et al. (2024)	40.0	90.0	100.0	35.0	10.0	55.0
$\pi_{0.5}$Intelligence et al. (2025)	45.0	100.0	100.0	40.0	35.0	64.0
Devol-ONE (Ours)	55.0	100.0	100.0	50.0	75.0	**76.0**

4.4 Ablation Studies

We conduct ablation studies on LIBERO to examine three aspects of Devol-ONE for the contribution of semantic and dynamics features to action generation, effect of layerwise interaction and dynamics prediction horizon. Further details for LIBERO-PLUS are listed in Appendix.

Action expert conditioning. Table 5 compares four conditioning configurations. VL only and JEPA only condition the action expert on vision-language features and predicted dynamics features, respectively. The two remaining configurations use both streams, either through final-layer fusion or through the layerwise joint attention used in Devol-ONE. It is worth noting that last-layer fusion is basically equivalent to Groot-like models (Lin et al., 2025b; Sun et al., 2026), by feeding last layer of hidden features into action expert instead of uniting attentions. This comparison examines both the value of combining the two information sources and the effect of how they are integrated into action generation. The full model achieves 98.4% average success, compared with 96.0% for VL only and 96.9% for JEPA only, supporting the complementary value of semantic and dynamics features. Layerwise joint attention also outperforms final-layer fusion on average (98.4% vs. 97.0%), with increment on Long (97.0% vs. 92.1%).

Prediction horizon. Table 6 compares prediction horizons of four, six, and eight frames. Average success increases from 96.8% to 97.5% and 98.4%, while Long-suite success rises from 94.5% to 96.2% and 97.0%, respectively. These results suggest that longer forecasts benefit extended manipulation sequences, supporting the eight-frame horizon used in our configuration.

5 Conclusion

In this paper, we present Devol-ONE, a Mixture-of-Transformers architecture that integrates vision-language understanding, latent dynamics prediction, and action generation through layerwise joint attention. By forecasting future states in V-JEPA latent space, Devol-ONE provides predictive context for action generation without synthesizing future video frames. We further introduced Autoregressive Latent Rollout Training (ALRT) to expose the policy to model-generated latent predictions during training. Experiments on LIBERO, LIBERO-Plus, RoboTwin 2.0, and real-world manipulation tasks demonstrate the effectiveness of the framework, while ablations support the benefits of complementary semantic and dynamics features and layerwise conditioning. These results highlight the potential of integrating latent world modeling into embodied policies.

AI USE STATEMENT

In this work, we used generative AI tools to assist with writing, coding, and reproducibility-related engineering support. We have not used generative AI tools for generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, cleaning and reformatting datasets, supporting qualitative and thematic data analysis, or interpreting results; none of these tasks were AI-assisted. Assistance with translation is not applicable to this work.

Specifically, we used generative AI tools to draft and edit parts of the manuscript text and improve its readability, and to assist in writing, refactoring, and debugging code used in our experiments and reproducibility efforts. All AI-assisted text was reviewed and edited by the authors, and all AI-assisted code was reviewed and tested by the authors for correctness before use. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. V-jepa: Latent video prediction for visual representation learning. 2023.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks, 2015. URL https://arxiv.org/abs/1506.03099.

Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation. *arXiv preprint*, 2024.

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. *arXiv preprint arXiv:2512.13030*, 2025.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint*, 2025.

Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. *arXiv preprint*, 2023.

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*, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J. Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael S. Ryoo, Grecia Salazar, Pannag R. Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong T. Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-1: robotics transformer for real-world control at scale. In *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, 2023.

Qingwen Bu, Jia Zeng, Li Chen, Yanchao Yang, Guyue Zhou, Junchi Yan, Ping Luo, Heming Cui, Yi Ma, and Hongyang Li. Closed-loop visuomotor control with generative expectation for robotic manipulation. *arXiv preprint*, 2024.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint*, 2025a.

Xizhou Bu, Jiexi Lyu, Fulei Sun, Ruichen Yang, Zhiqiang Ma, and Wei Li. Laof: Robust latent action learning with optical flow constraints. *arXiv preprint arXiv:2511.16407*, 2025b.

Jisong Cai, Long Ling, Shiwei Chu, Zhongshan Liu, Jiayue Kang, Zhixuan Liang, Wenjie Xu, Yinan Mao, Weinan Zhang, Xiaokang Yang, Ru Ying, Ran Zheng, and Yao Mu. Aha-wam:asynchronous horizon-adaptive world-action modeling with observation-guided context routing, 2026a. URL https://arxiv.org/abs/2606.09811.

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, 2026b. URL https://arxiv.org/abs/2602.12684.

Jun Cen, Siteng Huang, Yuqian Yuan, Kehan Li, Hangjie Yuan, Chaohui Yu, Bohan Hou, Yuming Jiang, Jiayan Guo, Xin Li, Hao Luo, Fan Wang, Deli Zhao, and Hao Chen. Rynnvla-002: A unified vision-language-action and world model, 2025a. URL https://arxiv.org/abs/2511.17502.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. Worldvla: Towards autoregressive action world model. *arXiv preprint*, 2025b.

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint*, 2024.

Xiaoyu Chen, Junliang Guo, Tianyu He, Chuheng Zhang, Pushi Zhang, Derek Cathera Yang, Li Zhao, and Jiang Bian. Igor: Image-goal representations are the atomic control units for foundation models in embodied ai. *arXiv preprint arXiv:2411.00785*, 2024a.

Xiaoyu Chen, Hangxing Wei, Pushi Zhang, Chuheng Zhang, Kaixin Wang, Yanjiang Guo, Rushuai Yang, Yucen Wang, Xinquan Xiao, Li Zhao, et al. villa-x: Enhancing latent action modeling in vision-language-action models. *arXiv preprint*, 2025.

Yi Chen, Yuying Ge, Yizhuo Li, Yixiao Ge, Mingyu Ding, Ying Shan, and Xihui Liu. Moto: Latent motion token as the bridging language for robot manipulation. *arXiv preprint*, 2024b.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 2023.

Yilun Du, Mengjiao Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Brian Ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B Tenenbaum, et al. Video language planning. *arXiv preprint*, 2023.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *NeurIPS*, 2024.

Ashley Edwards, Himanshu Sahni, Yannick Schroecker, and Charles Isbell. Imitating latent policies from observation. In *International conference on machine learning*, pp. 1755–1763. PMLR, 2019.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

Yao Feng, Hengkai Tan, Xinyi Mao, Guodong Liu, Shuhe Huang, Chendong Xiang, Hang Su, and Jun Zhu. Generalist bimanual manipulation via foundation video diffusion models. *arXiv preprint*, 2025.

Flexiv Robotics. Rizon 4 adaptive robot. https://www.flexiv.com/product/rizon, 2024. Accessed: 2026-09-18.

Shenyuan Gao, Siyuan Zhou, Yilun Du, Jun Zhang, and Chuang Gan. Adaworld: Learning adaptable world models with latent actions. In *Forty-second International Conference on Machine Learning*.

Quentin Garrido, Tushar Nagarajan, Basile Terver, Nicolas Ballas, Yann LeCun, and Michael Rabbat. Learning latent action world models in the wild. *arXiv preprint arXiv:2601.05230*, 2026.

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *ICCV*, 2017.

Jun Guo, Qiwei Li, Peiyan Li, Zilong Chen, Nan Sun, Yifei Su, Heyun Wang, Yuan Zhang, Xinghang Li, and Huaping Liu. Unified 4d world action modeling from video priors with asynchronous denoising, 2026. URL https://arxiv.org/abs/2604.26694.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. *arXiv preprint*, 2024.

Siyuan Huang, Liliang Chen, Pengfei Zhou, Shengcong Chen, Zhengkai Jiang, Yue Hu, Yue Liao, Peng Gao, Hongsheng Li, Maoqing Yao, et al. Enerverse: Envisioning embodied future space for robotics manipulation. *arXiv preprint*, 2025a.

Yuhang Huang, JIazhao Zhang, Shilong Zou, XInwang Liu, Ruizhen Hu, and Kai Xu. Ladi-wm: A latent diffusion-based world model for predictive manipulation. *arXiv preprint*, 2025b.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. pi0.5: a vision-language-action model with open-world generalization. *arXiv preprint*, 2025.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, et al. Dreamgen: Unlocking generalization in robot learning through neural trajectories. *arXiv preprint*, 2025.

Leonid Keselman, John Iselin Woodfill, Anders Grunnet-Jepsen, and Achintya Bhowmik. Intel RealSense stereoscopic depth cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 1267–1276, 2017. doi: 10.1109/CVPRW.2017.167.

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*, 2024.

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*, 2024.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint*, 2025.

Albina Klepach, Alexander Nikulin, Ilya Zisman, Denis Tarasov, Alexander Derevyagin, Andrei Polubarov, Nikita Lyubaykin, and Vladislav Kurenkov. Object-centric latent action learning. *CoRR*, abs/2502.09680, 2025.

Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial forcing: Implicit spatial representation alignment for vision-language-action model, 2025a. URL https://arxiv.org/abs/2510.12276.

Hengtao Li, Pengxiang Ding, Runze Suo, Yihao Wang, Zirui Ge, Dongyuan Zang, Kexian Yu, Mingyang Sun, Hongyin Zhang, Donglin Wang, and Weihua Su. Vla-rft: Vision-language-action reinforcement fine-tuning with verified rewards in world simulators, 2025b. URL https://arxiv.org/abs/2510.00406.

Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint*, 2024a.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. *arXiv preprint*, 2025c.

Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, et al. Vision-language foundation models as effective robot imitators. *arXiv preprint*, 2023.

Xinghang Li, Peiyan Li, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Tao Kong, Hanbo Zhang, and Huaping Liu. Towards generalist robot policies: What matters in building vision-language-action models. *arXiv preprint*, 2024b.

Tao Lin, Gen Li, Yilei Zhong, Yanwen Zou, Yuxin Du, Jiting Liu, Encheng Gu, and Bo Zhao. Evo-0: Vision-language-action model with implicit spatial understanding, 2025a. URL https://arxiv.org/abs/2507.00416.

Tao Lin, Yilei Zhong, Yuxin Du, Jingjing Zhang, Jiting Liu, Yinxinyu Chen, Encheng Gu, Ziyan Liu, Hongyi Cai, Yanwen Zou, Lixing Zou, Zhaoye Zhou, Gen Li, and Bo Zhao. Evo-1: Lightweight vision-language-action model with preserved semantic alignment, 2025b. URL https://arxiv.org/abs/2511.04555.

Tao Lin, Yuxin Du, Jiting Liu, Nuobei Zhu, Yunhe Li, Yuqian Fu, Yinxinyu Chen, Hongyi Cai, Zewei Ye, Bing Cheng, Kai Ye, Yiran Mao, Yilei Zhong, MingKang Dong, Junchi Yan, Gen Li, and Bo Zhao. Evo-depth: A lightweight depth-enhanced vision-language-action model, 2026. URL https://arxiv.org/abs/2605.14950.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: benchmarking knowledge transfer for lifelong robot learning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *NeurIPS*, 2023.

Jiaming Liu, Hao Chen, Pengju An, Zhuoyang Liu, Renrui Zhang, Chenyang Gu, Xiaoqi Li, Ziyu Guo, Sixiang Chen, Mengzhen Liu, et al. Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model. *arXiv preprint*, 2025a.

Mingyu Liu, Jiuhe Shu, Hui Chen, Zeju Li, Canyu Zhao, Jiange Yang, Shenyuan Gao, Hao Chen, and Chunhua Shen. Stamo: Unsupervised learning of generalizable robot motion from compact state representation. *arXiv preprint arXiv:2510.05057*, 2025b.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint*, 2024.

Yicheng Liu, Zibin Dong, Baijun Ye, Tianyuan Yuan, Tao Jiang, Anqi Yang, Shicheng Cao, Haonan Liu, Yue Sun, Zihan Guo, Xiao Liu, Ke Dong, Changxun Pan, Chenru Wu, Tailai Cheng, Xiaoshu Ren, Xinlei Zhang, Jianning Cui, Zijie Zhao, Haoyu Zhang, Kaiming Xu, Haodong Yang, Bowen Zhang, Jiahui Niu, Shaoting Zhu, Shiduo Zhang, and Hang Zhao. G0.5: One autoregressive stream for robot reasoning and action, 2026. URL https://arxiv.org/abs/2608.11739.

Yao Mu, Tianxing Chen, Shijia Peng, Zanxin Chen, Zeyu Gao, Yude Zou, Lunkai Lin, Zhiqiang Xie, and Ping Luo. Robotwin: Dual-arm robot benchmark with generative digital twins (early version). In *ECCV*, 2025.

Alexander Nikulin, Ilya Zisman, Denis Tarasov, Nikita Lyubaykin, Andrei Polubarov, Igor Kiselev, and Vladislav Kurenkov. Latent action learning requires supervision in the presence of distractors. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025.

Alexander Nikulin, Ilya Zisman, Albina Klepach, Denis Tarasov, Alexander Derevyagin, Andrei Polubarov, Lyubaykin Nikita, and Vladislav Kurenkov. Vision-language models unlock task-centric latent actions, 2026.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

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. *arXiv preprint*, 2025.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint*, 2025.

RoboTwin Team. RoboTwin 2.0 Leaderboard. `https://robotwin-platform.github.io/leaderboard`, 2025. Accessed: 2026-09.

Dominik Schmidt and Minqi Jiang. Learning to act without actions. In *The Twelfth International Conference on Learning Representations*.

Qiuhong Shen, Shihua Zhang, Yue Liao, Qi Li, Zhenxiong Tan, Shizun Wang, Shuicheng Yan, and Xinchao Wang. World action models: A survey, 2026. URL `https://arxiv.org/abs/2606.20781`.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint*, 2025.

StarVLA Community. Starvla: A lego-like codebase for vision-language-action model developing, 2026. URL `https://arxiv.org/abs/2604.05014`.

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. Vla-jepa: Enhancing vision-language-action model with latent world model, 2026. URL `https://arxiv.org/abs/2602.10098`.

GigaBrain Team, Angen Ye, Axiang Sun, Can Jin, Chenxi Cheng, Chong Shi, Dengke Shang, Dingqian Zhang, Guan Huang, Guangqiang Wang, Guangqing Ding, Guo Li, Hangcong Li, Hengyu Zhong, Hongtao Lu, Jianbo Qin, Jiming Mao, Jing Zhu, Jindi Lv, Jingzhi Cui, Junjie Xie, Junyi Bao, Kai Liu, Lei Yuan, Limin Long, Lv Feng, Mingming Yu, Peng Li, Pengfei Yi, Qi Li, Qianli Zhang, Qingfang Li, Qitang Hu, Rui Zhang, Shaoyan Sun, Shibo Sun, Shiying Duan, Tenghui Chen, Tianze Liu, Weijie Ke, Wenyao Xue, Xiaofeng Wang, Xiaoyu Tian, Xinyu Liu, Xinze Chen, Yang Wang, Yankai Wang, Yejun Zeng, Yifan Li, Yifei Nie, Yilong Li, Yilong Liu, Yongchao Feng, Yumeng Wang, Yun Ye, Zhichao Liu, Ziheng He, Zonghai Yang, and Zheng Zhu. Gigabrain-0.7: Scaling embodied foundation models to emergent capabilities with a three-system architecture, 2026. URL `https://arxiv.org/abs/2608.15875`.

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*, 2024.

Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. *ICLR*, 2024.

Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Zhibin Tang, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, et al. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. *IEEE Robotics and Automation Letters*, 2025.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *ICLR*, 2024.

Ganlin Yang, Zhangzheng Tu, Yuqiang Yang, Sitong Mao, Junyi Dong, Tianxing Chen, Jiaqi Peng, Jing Xiong, Jiafei Cao, Jifeng Dai, Wengang Zhou, Yao Mu, and Tai Wang. Eventvla: Event-driven visual evidence memory for long-horizon vision-language-action policies, 2026a. URL https://arxiv.org/abs/2606.20092.

Jiange Yang, Yansong Shi, Haoyi Zhu, Mingyu Liu, Kaijing Ma, Yating Wang, Gangshan Wu, Tong He, and Limin Wang. Como: Learning continuous latent motion from internet videos for scalable robot learning. *arXiv preprint*, 2025.

Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint*, 2023.

Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Dekang Qi, Junjin Xiao, Haoyun Liu, Ronghan Chen, Yuzhi Chen, Dongjie Huo, Feng Xiong, Xing Wei, Zhiheng Ma, and Mu Xu. Abot-m0: Vla foundation model for robotic manipulation with action manifold learning, 2026b. URL https://arxiv.org/abs/2602.11236.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. *arXiv preprint*, 2024.

En Yu, Haoran Lv, Jianjian Sun, Kangheng Lin, Ruitao Zhang, Yukang Shi, Yuyang Chen, Ze Chen, Ziheng Zhang, Fan Jia, Kaixin Liu, Meng Zhang, Ruitao Hao, Saike Huang, Songhan Xie, Yu Liu, Zhao Wu, Bin Xie, Pengwei Zhang, Qi Yang, Xianchi Deng, Yunfei Wei, Enwen Zhang, Hongyang Peng, Jie Zhao, Kai Liu, Wei Sun, Yajun Wei, Yi Yang, Yunqiao Zhang, Ziwei Yan, Haitao Yang, Hao Liu, Haoqiang Fan, Haowei Zhang, Junwen Huang, Yang Chen, Yunchao Ma, Yunhuan Yang, Zhengyuan Du, Ziming Liu, Jiahui Niu, Yucheng Zhao, Daxin Jiang, Wenbin Tang, Xiangyu Zhang, Zheng Ge, Erjin Zhou, and Tiancai Wang. Dm0: An embodied-native vision-language-action model towards physical ai, 2026. URL https://arxiv.org/abs/2602.14974.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination?, 2026. URL https://arxiv.org/abs/2603.16666.

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy. *arXiv preprint*, 2024.

Chubin Zhang, Jianan Wang, Zifeng Gao, Yue Su, Tianru Dai, Cai Zhou, Jiwen Lu, and Yansong Tang. Clap: Contrastive latent action pretraining for learning vision-language-action models from human videos, 2026.

Chuheng Zhang, Tim Pearce, Pushi Zhang, Kaixin Wang, Xiaoyu Chen, Wei Shen, Li Zhao, and Jiang Bian. What do latent action models actually learn? *arXiv preprint arXiv:2506.15691*, 2025a.

Hongyin Zhang, Pengxiang Ding, Shangke Lyu, Ying Peng, and Donglin Wang. Gevrm: Goal-expressive video generation model for robust visual manipulation. *arXiv preprint*, 2025b.

Jianke Zhang, Yanjiang Guo, Yucheng Hu, Xiaoyu Chen, Xiang Zhu, and Jianyu Chen. Up-vla: A unified understanding and prediction model for embodied agent. *arXiv preprint*, 2025c.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, et al. Dreamvla: A vision-language-action model dreamed with comprehensive world knowledge. *arXiv preprint*, 2025d.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. *arXiv preprint*, 2025.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint*, 2023.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Ya-Qin Zhang, Jiangmiao Pang, Jingjing Liu, Tai Wang, and Xianyuan Zhan. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model, 2025a. URL https://arxiv.org/abs/2510.10274.

Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. *arXiv preprint*, 2024.

Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loic Magne, et al. Flare: Robot learning with implicit world modeling. *arXiv preprint*, 2025b.

Linqing Zhong, Yi Liu, Yifei Wei, Ziyu Xiong, Maoqing Yao, Si Liu, and Guanghui Ren. Acot-vla: Action chain-of-thought for vision-language-action models. *arXiv preprint arXiv:2601.11404*, 2026.

Zhide Zhong, Haodong Yan, Junfeng Li, Xiangchen Liu, Xin Gong, Tianran Zhang, Wenxuan Song, Jiayi Chen, Xinhu Zheng, Hesheng Wang, et al. Flowvla: Visual chain of thought-based motion reasoning for vision-language-action models. *arXiv preprint arXiv:2508.18269*, 2025.

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. *arXiv preprint*, 2025.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. Irasim: Learning interactive real-robot action simulators. *arXiv preprint*, 2024.

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

A Appendix

A.1 RoboTwin 2.0 per-task success rates

Table 8 reports per-task Clean/Randomized success rates on RoboTwin 2.0. We compare Devol-ONE with $\pi_{0.5}$ (Intelligence et al., 2025), X-VLA (Zheng et al., 2025a), Fast-WAM (Yuan et al., 2026), GigaBrain-0.7 (Team et al., 2026), and X-WAM (Guo et al., 2026).

A.2 RoboTwin 2.0 per-task success rates

Table 8 reports per-task Clean/Randomized success rates on RoboTwin 2.0. We compare Devol-ONE with $\pi_{0.5}$ (Intelligence et al., 2025), X-VLA (Zheng et al., 2025a), Fast-WAM (Yuan et al., 2026), GigaBrain-0.7 (Team et al., 2026), and X-WAM (Guo et al., 2026).

Table 8: Per-task success rates on all 50 RoboTwin 2.0 tasks (Clean/Randomized, %). Baselines are co-trained on all 50 tasks and taken from the official RoboTwin 2.0 leaderboard; Devol-ONE is trained with single-task fine-tuning. In each column, the best result is in **bold** and the second best is underlined. *Overall* is the mean of the Clean and Randomized averages.

Task	$\pi_{0.5}$	X-VLA	Fast-WAM	GigaBrain-0.7	X-WAM	**Devol-ONE (Ours)**
Adjust Bottle	**99**/77	**99**/33	**99**/0	98/**98**	92/13	98/**98**
Beat Block Hammer	93/23	87/9	82/0	**94**/74	92/21	86/**93**
Blocks Ranking RGB	72/44	3/4	81/0	68/**76**	**85**/36	55/64
Blocks Ranking Size	45/21	**56**/25	52/0	42/28	33/0	26/**39**
Click Alarmclock	65/50	62/43	**99**/60	98/96	95/61	98/**99**
Click Bell	31/35	**100**/63	95/7	98/84	**100**/52	**100/100**
Dump Bin Bigbin	92/84	83/41	94/0	90/66	88/40	**95/93**
Grab Roller	99/97	**100**/20	**100**/2	**100/100**	95/16	**100/100**
Handover Block	30/12	**77**/1.3	66/0	16/32	21/21	19/**39**
Handover Mic	98/6	98/1	98/0	**100/100**	97/8	98/87
Hanging Mug	17/14	19/2	14/0	**26/36**	17/0	13/25
Lift Pot	**98**/35	96/15	91/0	86/86	77/0	96/**98**
Move Can Pot	60/10	58/0	**96**/0	86/**82**	77/34	65/72
Move Pillbottle Pad	46/32	66/3	**92**/0	32/**58**	78/26	43/39
Move Playingcard Away	84/65	84/5	**94**/0	58/**82**	83/11	83/**82**
Move Stapler Pad	20/18	**42**/0	33/0	26/**40**	26/2	33/29
Open Laptop	93/68	85/16	84/0	78/88	72/6	**96/98**
Open Microwave	**86/59**	43/5	17/0	82/50	9/24	39/42
Pick Diverse Bottles	66/42	50/30	72/0	38/50	**74**/26	56/**62**
Pick Dual Bottles	72/**57**	41/51	87/1	26/56	**99**/55	34/48
Place A2B Left	**71**/49	53/15	69/0	68/**72**	70/18	48/44
Place A2B Right	**68**/39	49/13	60/1	58/**68**	59/21	16/11
Place Bread Basket	63/60	85/52	**87**/0	60/**68**	82/40	59/67
Place Bread Skillet	76/47	**89**/17	85/0	80/50	82/37	59/**71**
Place Burger Fries	82/86	90/35	**97**/0	82/86	93/36	85/**93**
Place Can Basket	56/10	70/9	**77**/0	20/22	63/15	58/**67**
Place Cans Plasticbox	31/**73**	93/60	**98**/0	8/18	93/36	37/43
Place Container Plate	92/72	94/39	**100**/0	94/**94**	95/52	95/89
Place Dual Shoes	**70**/45	30/12	67/0	68/**72**	38/7	58/54
Place Empty Cup	**96**/80	90/16	90/0	90/**90**	93/36	84/88
Place Fan	66/26	63/1	59/0	**78/74**	47/6	37/48
Place Mouse Pad	34/31	53/0	**55**/0	26/**44**	49/6	0/29
Place Object Basket	70/29	81/6	**82**/0	74/**64**	52/20	59/45
Place Object Scale	78/38	76/6	**83**/0	64/**54**	66/14	35/36
Place Object Stand	85/47	91/43	**94**/0	88/**76**	83/28	18/32
Place Phone Stand	65/37	84/5	**85**/0	36/**52**	35/4	35/45
Place Shoe	82/50	83/39	87/0	**88/92**	76/31	76/91
Press Stapler	79/63	87/73	**90**/3	**90**/74	85/20	**90/81**
Put Bottles Dustbin	**78**/50	47/20	55/0	68/**54**	71/21	33/43
Put Object Cabinet	39/22	41/21	42/0	**78/82**	28/16	15/23
Rotate QRcode	**90**/21	81/1	81/0	54/58	47/2	79/**64**
Scan Object	49/40	57/9	**68**/0	32/46	32/14	58/**69**
Shake Bottle Horizontally	99/**100**	**100**/78	**100**/9	**100**/92	97/66	90/93
Shake Bottle	**100/100**	**100**/76	**100**/10	**100**/98	96/67	94/96
Stack Blocks Three	77/25	0/0	**89**/0	60/**78**	72/26	0/10
Stack Blocks Two	91/43	10/2	**99**/0	80/**92**	97/76	44/50
Stack Bowls Three	**81**/52	77/9	80/0	70/**64**	69/38	53/60
Stack Bowls Two	**94**/70	85/15	**94**/0	90/74	93/67	82/**88**
Stamp Seal	53/21	50/2	30/0	52/**54**	**58**/16	20/18
Turn Switch	52/25	40/6	39/3	42/52	**68**/4	51/**53**
Average	70.7/46.0	68.0/20.9	**77.8**/1.9	66.8/**67.9**	70.0/25.8	58.0/62.2
Overall	58.3	44.5	39.8	**67.4**	47.9	60.1

Overall comparison. Averaged over all 50 tasks, Devol-ONE achieves 58.0% under the Clean setting and 62.2% under the Randomized setting, for an overall average of 60.1%. This places it second among the compared methods, behind GigaBrain-0.7 (67.4%) and ahead of $\pi_{0.5}$ (58.3%), X-WAM (47.9%), X-VLA (44.5%), and Fast-WAM (39.8%). The advantage comes mainly from the Randomized setting, where Devol-ONE's average is second only to GigaBrain-0.7 (67.9%) and exceeds $\pi_{0.5}$ (46.0%), X-WAM (25.8%), X-VLA (20.9%), and Fast-WAM (1.9%). At the task level, Devol-ONE obtains the best Randomized score on 20 of the 50 tasks and the second best on

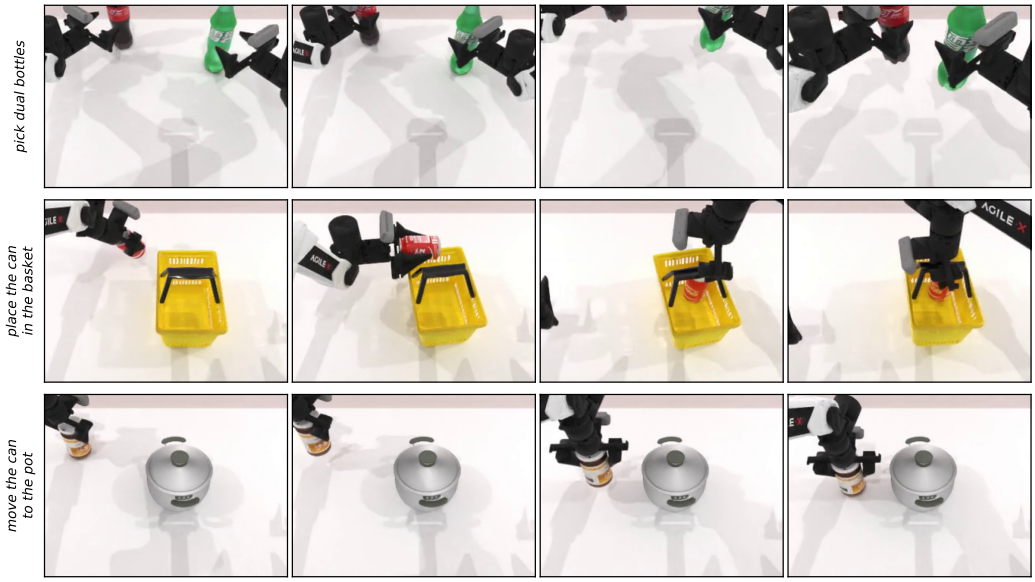


Figure 4: Example rollouts on RoboTwin 2.0.

another 17, placing it in the top two on 37 tasks; in 16 of the 17 tasks where it ranks second, it trails only GigaBrain-0.7. In head-to-head comparisons under the Randomized setting, Devol-ONE outperforms Fast-WAM, X-WAM, X-VLA, and $\pi_{0.5}$ on 50, 46, 46, and 37 tasks, respectively.

Sensitivity to domain randomization. The clearest difference between methods is the gap between the two settings. Most baselines degrade substantially under domain randomization: the average Clean-to-Randomized drop is 24.7 points for $\pi_{0.5}$, 44.1 for X-WAM, and 47.0 for X-VLA, while Fast-WAM, despite the highest Clean average (77.8%), falls to near-zero Randomized success on almost every task. Devol-ONE shows no such degradation; its Randomized average is 4.1 points above its Clean average, a pattern otherwise observed only for GigaBrain-0.7 (1.1 points). We note that Devol-ONE scores higher under Randomized than Clean on 34 of the 50 tasks. Because the Randomized setting is designed to be strictly harder, we do not interpret this as a benefit of randomization itself, and leave a closer analysis to future work.

Limitations. Devol-ONE has the lowest Clean average among the compared methods, with the best Clean score on 5 tasks and the second best on 7. Its failures concentrate on tasks that require precise placement relative to a target or fine-grained reasoning over multiple objects, such as Place Mouse Pad (0/29), Stack Blocks Three (0/10), Place A2B Right (16/11), Hanging Mug (13/25), Put Object Cabinet (15/23), Place Object Stand (18/32), and Stamp Seal (20/18). Finally, Devol-ONE is trained with single-task fine-tuning, whereas all baselines are co-trained on the full 50-task suite, so the per-task comparison is not a strictly controlled one.

A.3 Additional LIBERO-Plus Results

We report two additional evaluations on LIBERO-Plus (Fei et al., 2025): in-distribution fine-tuning and the action-expert conditioning ablation under the zero-shot protocol. All evaluations use the full test set of 10,030 episodes.

A.3.1 In-Distribution Fine-Tuning

We fine-tune Devol-ONE on the LIBERO-Plus training split and evaluate after 50k and 150k steps (Table 9). Fine-tuning raises the total success rate from 71.4% in the zero-shot setting to 82.2% and

84.4%, with the largest gains on Camera and Noise. Longer training improves every factor, most notably Robot (+5.1) and Language (+3.1).

Table 9: **In-distribution results on LIBERO-Plus (%).** Δ: change from 50k to 150k steps. #Ep.: evaluation episodes per factor. Robot denotes robot initial-state perturbations.

Factor	#Ep.	Zero-shot	SFT 50k	SFT 150k	Δ
Camera	1,599	47.3	92.1	93.9	+1.8
Robot	1,550	45.7	43.1	48.2	+5.1
Language	1,537	85.9	81.7	84.8	+3.1
Light	1,142	94.5	97.2	97.9	+0.7
Background	1,076	91.8	96.6	96.8	+0.2
Noise	1,601	67.5	92.9	94.3	+1.4
Layout	1,525	80.5	79.3	81.2	+1.9
Total	10,030	71.4	82.2	84.4	+2.2

A.3.2 Action-Expert Conditioning on LIBERO-Plus

We evaluate the conditioning variants of Table 5 on LIBERO-Plus under the zero-shot protocol (Table 10). The layerwise model achieves the best total (71.4%), exceeding VL only by 2.4 points and JEPA only by 6.3 points, and leads on five of the seven factors. The two single-source variants are complementary: JEPA only is the most robust to robot initial-state perturbations, whereas VL only is stronger on Language, Camera, and Noise.

Table 10: **Action-expert conditioning on LIBERO-Plus (zero-shot, %).** Best result per column in bold.

Conditioning	Camera	Robot	Language	Light	Background	Noise	Layout	Total
VL only	**48.7**	43.1	82.8	90.6	90.1	62.1	79.1	69.0
JEPA only	37.5	**55.4**	72.3	92.7	89.9	46.8	77.6	65.1
Layerwise (Ours)	47.3	45.7	**85.9**	**94.5**	**91.8**	**67.5**	**80.5**	**71.4**

A.4 Pretraining details

Backbones and world model. The dynamics-stream predictor consumes the V-JEPA2 encoder's features from the current/past observations and is trained to predict the encoder's features on *future* frames (Section 3); Table 11 lists the corresponding sizes.

Table 11: Backbone and world-model hyperparameters.

Hyperparameter	**Value**
Vision-language stream:	
Backbone	Qwen3-VL-2B-Instruct
Hidden size	2048
Attention implementation	SDPA
Dynamics stream:	
Encoder	V-JEPA2 ViT-L/16 (Assran et al., 2025)
Encoder checkpoint	`vjepa2-vitl-fpc64-256`
Predictor depth	12
Predictor attention heads	8
Temporal window (frames)	8
Camera views	3
Latent dynamics tokens / timestep	32

Action expert. The action expert is a DiT-Base (Peebles & Xie, 2023) flow-matching head; Table 12 lists its hyperparameters.

Table 12: Action expert hyperparameters.

Hyperparameter	**Value**
Action dimensions	14
State dimensions	16
Future action window size K	6
Past action window size	0
Action horizon H	7
Repeated diffusion steps	8
Flow-matching time sampling: Beta α	1.5
Flow-matching time sampling: Beta β	1.0
Flow-matching time sampling: rescale s	0.999
Timestep buckets	1000
Inference timesteps	4
Number of target vision tokens	32
Diffusion Parameters:	
Cross-attention width	2048
Norm type	`ada_norm`
Number of layers	16
Output dimensions	1024
Positional embeddings	none
Dropout	0.2

Data. The robot-action stream mixes AgiBotWorld-ReinforcementLearning archives (Home and Industry environments) with DROID (Khazatsky et al., 2024); both sources carry frame-level sub-task annotations, used as the CoT-conditioning instruction described in Section 3 rather than the coarser episode-level instruction. An auxiliary Something-Something V2 (Goyal et al., 2017) stream with no action labels is co-trained to expose the vision-language and dynamics streams to broader human motion. Table 13 lists the mixture composition.

Table 13: Pretraining data mixture.

Property	**Value**
Robot-action stream:	
AgiBotWorld-RL archives (Home + Industry)	78
AgiBotWorld-RL sample weight (per archive)	1.0
DROID sample weight	150
Composition (DROID : AgiBotWorld-RL)	$\approx$65.8% : 34.2%
Action representation	delta-position, delta-rotation-vector
Instruction conditioning	frame-level CoT sub-task label
Per-device batch size	24
Auxiliary video stream:	
Source	Something-Something V2 (Goyal et al., 2017)
Action labels	none
Loss terms contributed	$\mathcal{L}_{\mathrm{VL}}, \mathcal{L}_{\mathrm{dyn}}$ (no $\mathcal{L}_{\mathrm{act}}$)
Per-device batch size	16
Both streams:	
Model input resolution	224×224
Source decode resolution	256×256

Optimization. Table 14 lists the optimizer, learning-rate schedule, distributed-training, and loss-weighting hyperparameters (Section 3).

Table 14: Optimization hyperparameters.

Hyperparameter	Value
Optimizer (AdamW):	
β_1	0.9
β_2	0.95
ϵ	10^{-8}
Weight decay	10^{-8}
Learning rate:	
Peak LR: dynamics predictor & fusion params.	3×10^{-5}
Peak LR: pretrained VL backbone	1×10^{-5}
Peak LR: action expert	1×10^{-4}
Warmup steps	10,000 (10%)
Schedule	cosine decay
Decay floor	10^{-6}
Gradients & distributed training:	
Gradient clipping (global norm)	1.0
Gradient accumulation steps	1
GPUs	4
Effective batch size (robot-action / video)	96 / 64
Precision	mixed precision (bf16)
Memory optimization	gradient checkpointing, DeepSpeed ZeRO-2
Loss weights:	
Dynamics loss λ_{dyn}	0.1
CoT text loss λ_{VL}	0.1
Action loss λ_{act} (implicit)	1.0
Checkpoint interval (steps)	10,000

A.5 Details of real-world rollout

The robot rig used for teleoperation is a fixed, two-arm Flexiv manipulation cell consisting of two Flexiv Rizon (Flexiv Robotics, 2024) 7-DoF collaborative arms. It is controlled through Flexiv's RDK `flexivrdk` with Python bindings. The left arm is a Flexiv Rizon 4 arm, and the right arm is Flexiv Rizon 4R arm, a variant with force and torque sensing. The right arm's force and torque sensing abilities are not used for this experiment. The platform is positioned over a black tabletop and illuminated with artificial light. Natural light is blocked to reduce illumination changes across recording sessions.

Each arm uses a WRobot gripper, model `W-EPGC-50-150-B-C-NM-L5`. Each gripper is controlled through direct register writes using Modbus RTU over a USB-to-serial adapter at $115\,200\,\text{baud}$. The interface uses registers for target position, speed, and grip force, which are initialized upon connection. The interface represents the gripper state as a binary variable with hysteresis: it is set to the closed state at $< 30\,\%$ open and the open state at $> 90\,\%$ open.

Three Intel RealSense D-series (Keselman et al., 2017) color cameras are used, with a 640×480 pixel resolution at $30\,\text{Hz}$ frame rate. One camera is mounted between the two arms, and one is mounted on each wrist. The cameras are driven with `pyrealsense2`. The cameras return `bgr8` color frames without depth data. The frames are JPEG-encoded with a quality setting of 90. The cameras are calibrated using a checkerboard/ChArUco board with 6×9 squares, a square size of $30\,\text{mm}$, and a marker size of $20\,\text{mm}$.

A.5.1 Tasks and Task Criteria

The tasks detailed in Table 4 are as follows. Figures 5–9 show, for three sample episodes per task, the first and final frame of the egocentric view together with the corresponding wrist-camera view(s).

"Insert the Ethernet Connector" - This task is performed with the left arm only. A network switch with Ethernet receptacles is present at roughly the center of the table. Two Ethernet cables are placed on the table to the left and right of the switch; each cable has one end already secured in a fixture and the other end loose. The robot must successfully pick up the loose end of a cable and fully plug it into a receptacle on the switch. This is done twice, serially, with a single (left) arm.

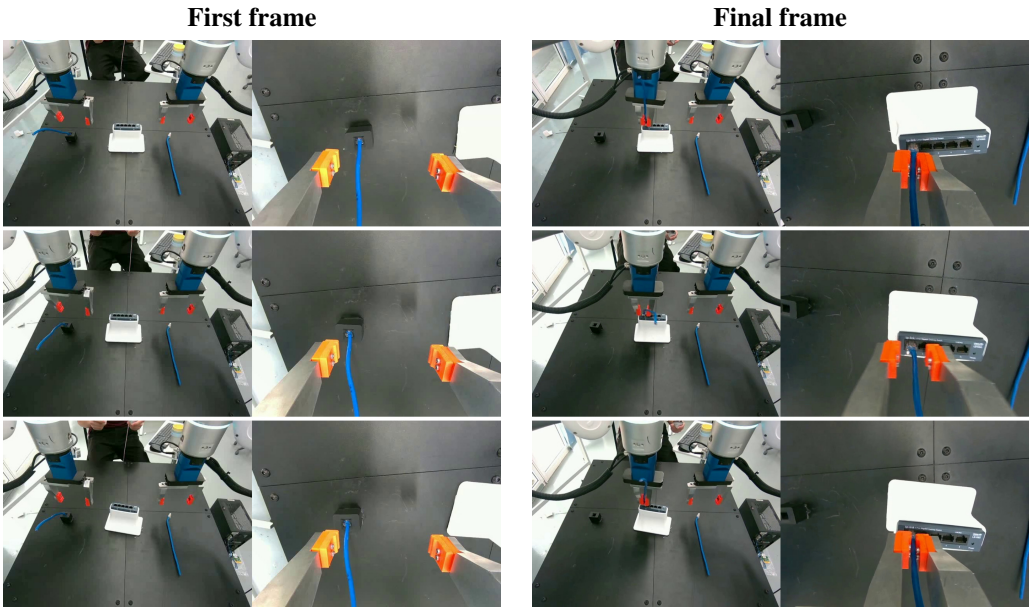


Figure 5: Three sample episodes of the **Ethernet** task (rows, top to bottom), each showing the egocentric view and the left-wrist view at the first and final frame.

"Stack realsense [*sic*] boxes with both arms/with only the left arm" - Three Intel RealSense boxes are scattered around the table, and the robot must successfully stack them on top of one another, either using both arms in coordination (**2arm box**) or using only the left arm (**1arm box**).

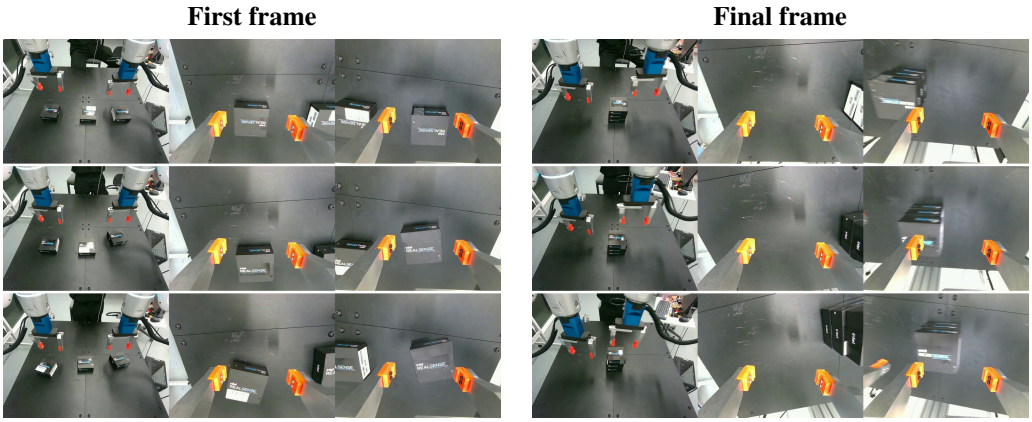


Figure 6: Three sample episodes of the **2arm box** task variant (dual-arm; rows, top to bottom), each showing the egocentric view and both wrist views at the first and final frame.

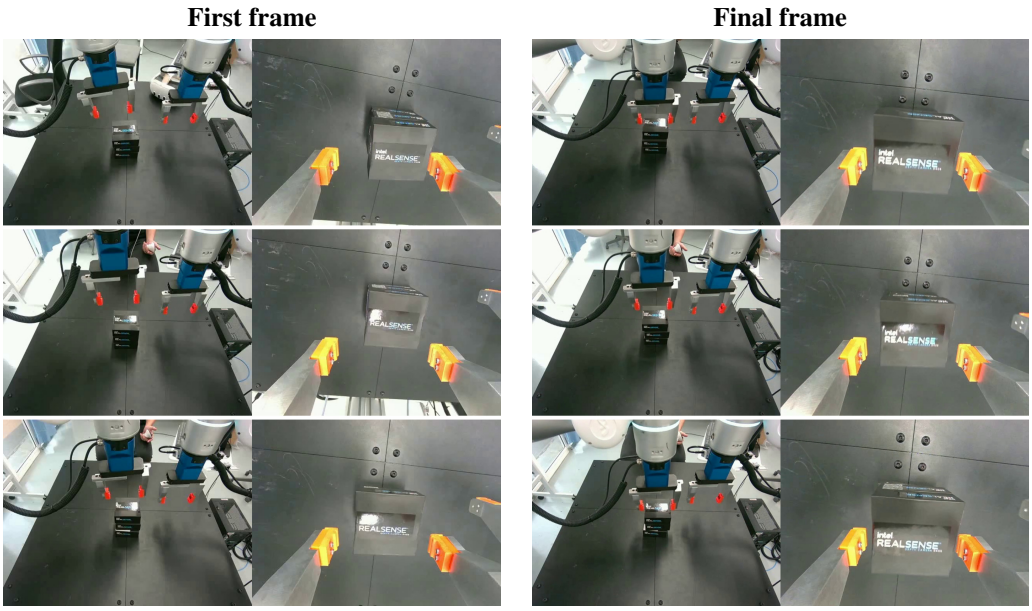


Figure 7: Three sample episodes of the **1arm box** task variant (single, left arm only; rows, top to bottom).

"Place the snacks into a basket" - Two snack bags are placed on the table next to an open plastic basket, and the robot must place the snacks into the basket using both arms. The task is considered successful if at least one of the snack bags is successfully placed in the bag.

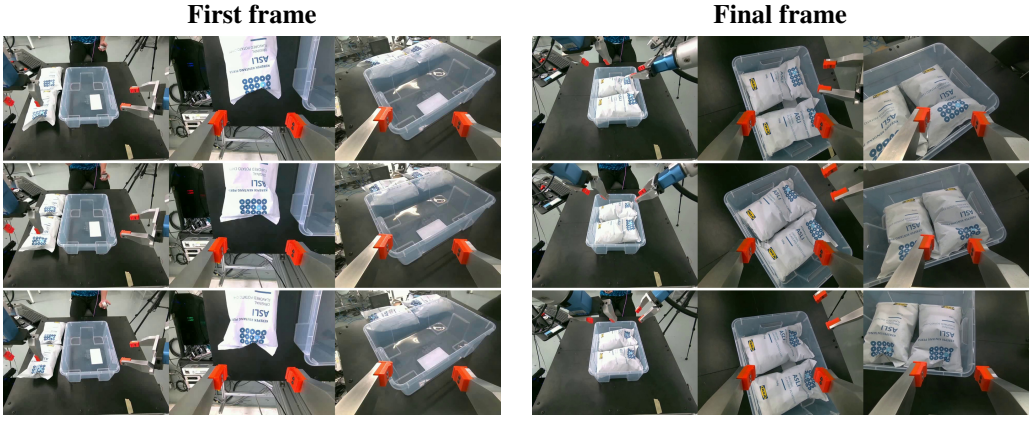


Figure 8: Three sample episodes of the **Basket** task (rows, top to bottom).

"Stack four different-colored cups" - Four different-colored cups are placed on the table, and the robot must stack them in no particular order, using both arms. The task is considered successful if at least three cups are stacked.

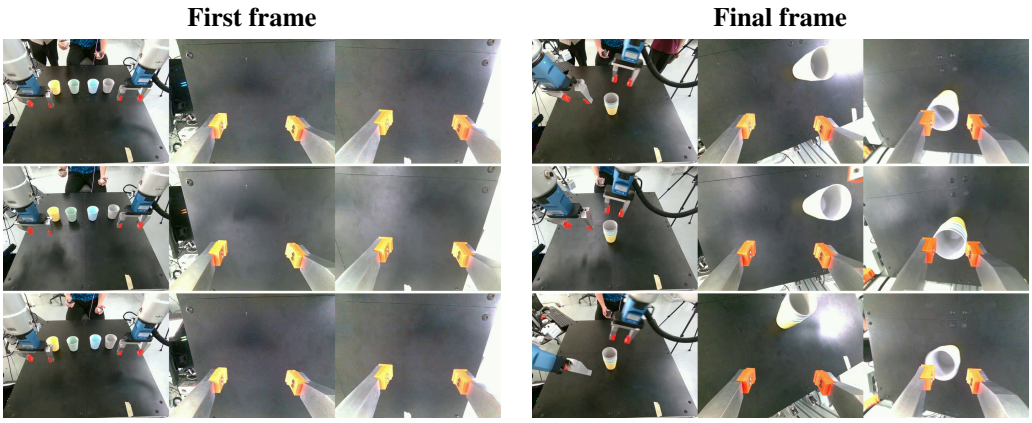


Figure 9: Three sample episodes of the **4cups** task (rows, top to bottom).