LEARNING TRANSFERABLE WORLD-ACTION MODELS FROM TASK-PAIRED HUMAN VIDEOS

Anonymous authors
Paper under double-blind review

ABSTRACT

World-action models inherit broad visual and physical priors from video pre-training. However, standard within-video next-frame prediction allows them to rely on appearance shared between the context and future, rather than learning interactions that transfer across scenes and embodiments. We introduce WATCH, a cross-video pre-training that learns from task-paired human demonstrations collected across diverse environments. Conditioned on the video and actions of one demonstration, the model jointly predicts another demonstration's actions and future frames from its current observation. Predicting one demonstration from another encourages the model to learn task-relevant interactions that transfer across scenes and demonstrators. Our pre-training shows that these capabilities can be learned from readily scalable human–human pairs alone, without requiring matched human–robot demonstrations. Analyses before robot post-training show improved motion alignment across embodiments and better action and video prediction on unseen robots, with representations closer to those learned from robot pairs. After robot post-training, WATCH improves MimicDroid in-context success from 40.9% to 68.3% with Cosmos Policy and from 59.8% to 77.7% with Cosmos3, with the largest gains on the unseen levels. Gains persist under context-free post-training across robot embodiments. WATCH outperforms within-video pre-training and co-training baselines, supporting cross-video prediction as a scalable approach to learning transferable world-action representations.

1 INTRODUCTION

Video models learn rich spatiotemporal representations by predicting how visual scenes evolve, which increasingly serve as world models (NVIDIA, 2025a; Wan et al., 2025) and backbones for robot foundation models (Kim et al., 2026; Yuan et al., 2026; Ye et al., 2026). World-action models extend this predictive capability to robot control by jointly modeling actions and their visual consequences, providing a learning signal not only for which action to take but also for how that action changes the world. These models typically build on video backbones pretrained with next-frame prediction over large-scale datasets spanning diverse environments and embodiments (NVIDIA, 2025a; Assran et al., 2025). Ideally, such pre-training should learn action representations that preserve the motion and interaction shared across different scenes and embodiments (Zheng et al., 2026; Ye et al., 2025), rather than the appearance and execution style specific to a particular environment or demonstrator.

However, standard video pre-training predicts future frames **within a single demonstration** (NVIDIA, 2025a; Assran et al., 2025), where the context and target share scene appearance, viewpoint, and collection source. This continuity allows prediction to exploit demonstration-specific visual cues without requiring transferable interactions to be separated from those factors. Such entanglement can make post-training for robot control more difficult, as the model must learn to predict robot actions under differences in environment, embodiment, and viewpoint from the pre-training data. Even with diverse training data, the within-video objective does not explicitly encourage the model to represent similar motions and interactions consistently across different scenes and embodiments.

A straightforward way to encourage such consistency is to condition prediction on a different demonstration of the same task. Under this **cross-video training**, the reference supplies the task and its motion, while the query's current observation provides the context for predicting its actions and future frames. Because the two demonstrations differ in scene, viewpoint, or demonstrator, relating

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.

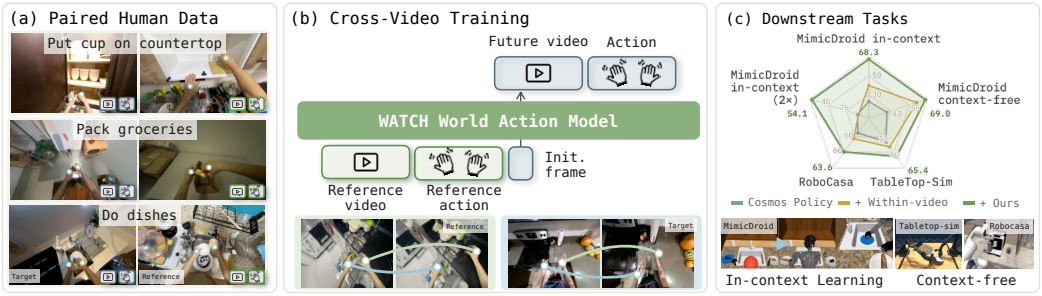


Figure 1: **WATCH overview. (a)** Human videos of the same task are paired across collection sources and scenes. **(b)** Conditioned on the reference video and actions, the model predicts the target's actions and future frames from its initial frame. The reference's appearance does not transfer, so what it can pass on is the interaction. **(c)** After robot post-training, WATCH substantially improves success rates over the base model in both in-context and context-free settings.

the reference to the query encourages the model to capture shared motion and interaction beyond demonstration-specific visual cues. Related cross-embodiment policies learn such reference-to-query correspondences for robot control using task-matched human–robot demonstrations (Jain et al., 2024; Shah et al., 2025; Zhou et al., 2026; Park et al., 2026). Using this pairing strategy for large-scale pre-training, however, would require corresponding robot demonstrations, whose collection incurs the costs of robot hardware and teleoperation.

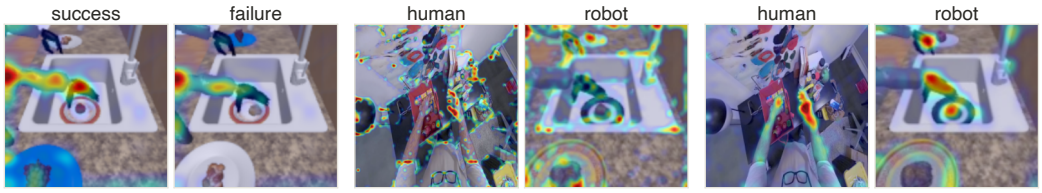
Human demonstrations can be collected across people and everyday environments without robot hardware or teleoperation (Yang et al., 2025; Hoque et al., 2026; Punamiya et al., 2026; Zheng et al., 2026), offering a potentially more scalable and cost-effective source of pre-training data. To learn from this diversity through cross-video pre-training without matched robot demonstrations, we introduce **WATCH**, **W**orld-**A**ction model **T**raining from **C**ross-**H**uman videos. Our key insight is that demonstrations of the same task from different collection sources (Punamiya et al., 2026; Hoque et al., 2026) already vary in demonstrator, viewpoint, and environment, and can therefore serve as cross-embodiment pairs without matched robot data. This formulation allows pre-training to scale with human demonstrations alone, reserving costly robot data collection for subsequent adaptation to the target embodiment. As shown in Fig. 1, the model conditions on one human demonstration and predicts the actions and future frames of another, improving downstream robot success in both in-context and context-free settings after robot post-training.

Our analyses (§3) suggest that cross-video pre-training learns more embodiment-invariant features through joint action–video supervision, aligning motions across collection sources while preserving motion information and focusing attention on hand–object interactions. These changes accompany improved robot post-training (§4): MimicDroid (Shah et al., 2025) success rises from 40.9% to 68.3% with Cosmos Policy, with gains also observed across backbones and embodiments under context-free post-training. Comparisons with within-video learning (Yang et al., 2025; Luo et al., 2025) and co-training (Punamiya et al., 2026; Zheng et al., 2026), together with improvements from more paired human data, support cross-demonstration prediction as a scalable route to transferable world-action representations.

Our main contributions are as follows: (1) We introduce WATCH, a cross-video world-action pre-training objective that conditions on one human demonstration while predicting another, together with task-matched human-to-human pairs that supply embodiment and environment variation without paired robot demonstrations. (2) We show that cross-video pre-training shifts representations from source-specific appearance toward task-relevant motion and interaction (§3). (3) We show consistent gains across video-model backbones, post-training protocols, and evaluation environments, largest on the unseen levels (§4).

2 WATCH: World-Action Model Training from Cross-Human Videos

We introduce WATCH, a human-data-only pre-training stage that aims to learn embodiment-invariant interaction representations before robot post-training. We first review within-video world-action



(a) Post-trained original model (b) Original video model (c) Cross-video pre-training (ours)

Figure 2: **Where a world-action model looks.** Attention to the current frame, grouped by model in the order the text reads them. (a) After robot post-training of the baseline video backbone (Cosmos-Predict2.5), attention focuses on the hand and object in a successful rollout and on the forearm in a failed one (`PnPSinkToCabinet`). (b) The original video model attends broadly to edges and background on both robot frames and human demonstrations, which offer a scalable source of interaction and motion cues for robot learning. (c) On the same frames, our cross-video pre-training shifts attention toward the acting hand and manipulated object.

modeling (§2.1) and what the current video prior may miss (§2.2), then describe how task-paired human demonstrations and cross-video action–video prediction encourage the model to capture interactions shared across scenes and embodiments, and how the resulting model supports both in-context and context-free robot post-training (§2.3).

2.1 Preliminary

World-action models and within-video training. World-action models turn the spatiotemporal priors of video models into policies by jointly predicting actions and their visual outcomes (Li et al., 2025; Zhu et al., 2025; Kim et al., 2026). Cosmos Policy (Kim et al., 2026) implements this principle by incorporating state and action representations into the latent token sequence of a pre-trained video model; we follow its recipe on Cosmos-Predict2.5 (NVIDIA, 2025a) (§D.2). The latent sequence is split into clean conditioning slots and target slots, the latter holding future frames and actions. Training follows the backbone's rectified-flow objective, $\mathcal{L}(\theta) = \mathcal{L}_{\text{visual}}(\theta) + \lambda\,\mathcal{L}_{\text{action}}(\theta)$, where each term applies velocity matching to the corresponding target slots, with no loss on conditioning slots. These models typically follow a **within-video** formulation: conditioning observations and prediction targets come from the same demonstration or episode. Writing I_t for the frame, s_t for the proprioceptive state, ℓ for language instruction, and a_t for the action at time t, the model learns $p_\theta(a_{t:t+C}, I_{t+1:t+C+1} \mid I_t, s_t, \ell)$ over a chunk size of C steps, where $x_{i:j}$ denotes $x_i, \ldots, x_{j-1}$, so that $a_{t:t+C}$ holds C actions. The condition and target therefore share the same scene, viewpoint, and demonstrator (or embodiment).

2.2 Motivation

What the video prior may miss. We examine Cosmos Policy (Kim et al., 2026) obtained by directly post-training Cosmos-Predict2.5 (NVIDIA, 2025a) on robot data. In the rollouts of Fig. 2a, attention focuses on the hand near the manipulated object in the successful one, but on the wrist and forearm in the failed one. To assess whether the original video backbone emphasizes these cues, we inspect its attention on human demonstrations as well as robot observations. In both cases, attention is broadly distributed over edges and background rather than concentrated on the interacting hand and object (Fig. 2b). We hypothesize that this partly reflects its pre-training objective, which models visual change throughout the scene without prioritizing the interaction relevant for control. This motivates **cross-video pre-training**, where another demonstration of the same task guides prediction in the current scene: predicting across demonstrations with different object placements and demonstrators encourages the model to capture the shared interaction while adapting the motion to the current hand–object configuration. Consistent with this, cross-video pre-training shifts attention toward the acting hand and manipulated object in both human and robot observations (Fig. 2c).

2.3 Cross-Video World-Action Modeling

Human data and cross-video pairs. Human demonstrations collected from different people, with different recording rigs and in different environments, vary along the axes on which a robot differs from a human, and task-matched pairs across these sources supply this variation without requiring matched



Figure 3: **Overview of WATCH.** The reference video and its action trajectory, from a different demonstration of the same task, enter as clean conditioning tokens together with the query's current frame and proprioceptive state; the query's action chunk and future frames are the noised tokens the model denoises. Every token is tagged with its slot type and physical time, so the same sequence layout carries over to post-training with a human demonstration as the reference or with the reference slots removed. The loss is the joint velocity-matching objective of §2.1, applied to the generated tokens.

robot data. We construct query–reference pairs from different episodes of EgoVerse (Punamiya et al., 2026). Human actions are derived from subsequent body configurations and expressed with states in a shared human–robot task space, using a canonical frame fixed at each episode's start. A semantic task taxonomy restricts candidates to compatible activities, which we rank by dynamic-time-warping distance between motion trajectories. Motion descriptors are standardized within each window, which reduces their sensitivity to motion magnitude so that interactions can match despite differences in reach. Each query's reference set $\mathcal{R}(q)$ holds up to 8 nearest matches from its own source and up to 7 from other sources, the latter admitted within a fixed factor of the best score. Training samples one reference uniformly from this set at each step, from any source (§C.2).

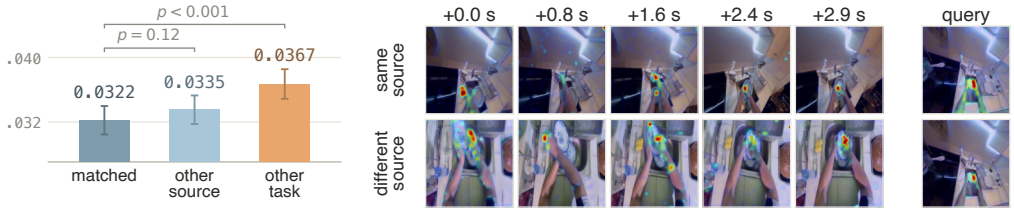
This procedure yields 10.99M query–reference pairs, 10.0K hours of paired 3-second windows counted over all pairs. Each query has 10.52 references on average, including 6.35 cross-source matches; 90.7% of queries have at least one cross-source reference. Compared with same-task random cross-source pairs, the selected matches show higher trajectory-shape cosine similarity by 0.36, indicating agreement in motion beyond task semantics (§C.3).

Reference-conditioned joint prediction. As illustrated in Fig. 3, the model receives the current query frame and state together with the reference video and actions. For a query window q and a reference window r drawn from its reference set, we model

$$p_\theta\left(a^q_{t:t+C},\, I^q_{t+1:t+C+1} \mid I^q_t,\, s^q_t,\, \ell,\, I^r_{u:u+C+1},\, a^r_{u:u+C}\right), \qquad r \sim \mathcal{R}(q), \tag{1}$$

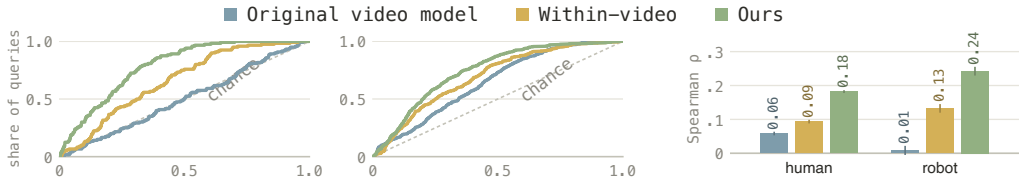
over a prediction horizon C, where u is the start of the reference window and the task description ℓ enters as text on both backbones. The reference supplies an example of task progression while the query observation specifies the target scene and source, so predicting the query future requires transferring the demonstrated interaction into the query's visual and geometric context. Inputs and outputs are encoded as six slots: current query frame (I^q_t), query state (s^q_t), reference video ($I^r_{u:u+C+1}$), reference actions ($a^r_{u:u+C}$), predicted future video ($I^q_{t+1:t+C+1}$) and actions ($a^q_{t:t+C}$). Each carries its own positional encoding over slot type and physical time, so where the reference begins and where the action begins are marked by slot rather than by absolute offset, and the encoding carries over to post-training as slots are repopulated or dropped. Following the joint video-action prediction recipe of Cosmos Policy, we implement WATCH on Cosmos-Predict2.5 and Cosmos3 (NVIDIA, 2026) using the joint objective of §2.1, applied only to the generated slots, which ties the predicted motion to its expected visual effect while retaining the backbone's dynamics prior. A query and its reference are 3-second windows, the action term is weighted by $\lambda = 48$, and post-training on the target robot keeps the same objective and slot layout (§D.1).

Post-training with and without context. The model trained by WATCH supports both in-context and context-free post-training, with predicted actions expressed in the target robot's native control space. In-context post-training conditions robot action and future-video prediction on a human reference demonstration. Context-free post-training removes the reference slots and predicts robot actions and future frames from the robot's current observation, state, and task instruction. Both protocols retain the same backbone and joint prediction objective.



(a) Sensitivity to the reference (b) Action-token attention

Figure 4: **Sensitivity to the reference demonstration. (a)** Action MSE (lower is better) over 137 held-out episodes using a matched reference, a similar-motion reference from a different collection source, or a different-task reference from the same source. Brackets: Holm-corrected p-values, episode-paired Wilcoxon test. Our model is less sensitive to a change of source than of task, suggesting that our training encourages a focus on task-relevant actions. **(b)** Action-token attention for the same query (do dishes) with references from the same collection source (top) and a different one (bottom). Even with a reference from a different source, attention concentrates on interactions central to task execution.



(a) Retrieval, human (b) Retrieval, robot (c) Held-out embodiment regression

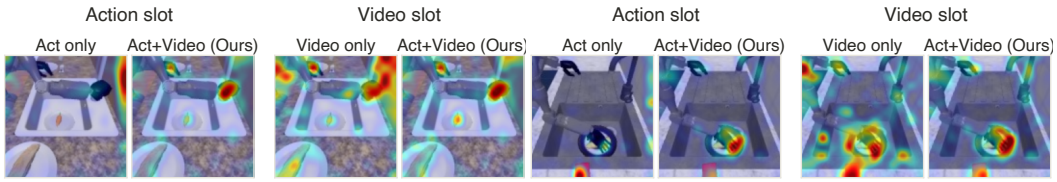
Figure 5: **Representation alignment and motion transfer across embodiments.** Action-slot features from layers 10–27, with retrieval ranks and regression scores averaged over these layers, on 359 held-out EgoVerse windows (human) and 396 RoboMIND windows from three robots (robot). **(a, b)** Motion retrieval: share of queries whose matched window from a different collection source (a) or robot (b) falls within a normalized rank; a higher curve is better, the diagonal is chance. **(c)** Held-out embodiment regression: a regressor fit on two embodiments predicts nine future-motion quantities on the third; bars are the mean Spearman correlation over quantities and folds, error bars the standard error over layers.

3 What Cross-Video Pre-Training Learns

Motivated by the hypothesis that cross-video prediction encourages the model to capture interactions shared across demonstrations, we examine what WATCH learns before robot post-training: how it uses task information from a reference to predict interactions in another demonstration, through reference sensitivity, motion alignment, and hand–object attention. We then assess how this learning extends to unseen human and robot demonstrations, including whether human pairs yield representations similar to those learned from robot pairs. We use held-out human pairs from EgoVerse (Punamiya et al., 2026), paired robot data from RoboMIND (Wu et al., 2025), and robot observations from MimicDroid (Shah et al., 2025), the post-training benchmark of §4.1. The results suggest that cross-video pre-training captures task-relevant motion and interactions across embodiments.

Conditioning is more sensitive to task changes. We compare matched references with similar-motion references from a different collection source and different-task references from the same source in Fig. 4a. Action MSE increases from 0.0322 to 0.0335 under the source change and to 0.0367 under the task change, only the latter significant after Holm correction. In the visualized examples, attention remains concentrated on the acting hands and manipulated object when the reference source changes (Fig. 4b; §E.3). Together, these observations suggest that reference conditioning tracks task-relevant interactions across collection sources.

Embodiment-invariant motion representations from human demonstrations. Cross-embodiment motion transfer requires both alignment of corresponding movements and preservation of information about how those movements unfold. We examine both properties using action-slot features of the original video model, the within-video baseline, and WATCH on 359 held-out EgoVerse windows and 396 RoboMIND windows from three robots (Fig. 5). In motion retrieval (Fig. 5a,b), WATCH ranks



(a) `TurnOnFaucet` (b) `PnPSinkToRightCounterPlate`

Figure 6: **Attention visualizations under different pre-training supervision.** All models use cross-video pairing and differ in which reference modalities they receive and predict: actions only (Act only), future frames only (Video only), or both (Act+Video, ours). Joint supervision concentrates attention on the acting hand and the manipulated object on both the action and the video slot.

matching motions from different human sources and different robots earlier than both baselines. To test whether this alignment also preserves transferable motion information, we fit a regressor on two embodiments and evaluate it on the third (Fig. 5c). It predicts future motion from `WATCH`'s features at a Spearman correlation of 0.18 on human and 0.24 on robot windows, against 0.06 and 0.01 for the original video model. These complementary results suggest that human cross-video pre-training supports both cross-embodiment correspondence and a transferable encoding of motion, extending to unseen robots.

Joint supervision focuses attention on interactions. To understand the role of predicting both motion and its visual consequences, we compare attention under action-only, video-only, and joint training with the same cross-video pairing (§2.3), using the respective modalities as reference inputs and prediction targets. Fig. 6 shows attention from the corresponding output slots on identical query frames. Action-only attention peaks near image boundaries, away from the manipulated object, while video-only attention spreads across the interaction region and surrounding scene. Joint training concentrates both action-slot and video-slot attention on the acting hand and object, around the faucet in `TurnOnFaucet` and the grasp in `PnPSinkToRightCounterPlate`. This shared focus suggests that jointly predicting motion and its visual consequences encourages both outputs to attend to the interaction connecting them.

The prior generalizes to unseen robot data. We evaluate whether a model pre-trained on human demonstrations generalizes to robot data unseen during pre-training, using action and video prediction on RoboMIND (Wu et al., 2025) (Table 1). Compared with the within-video baseline using query context alone, reference-conditioned `WATCH` lowers action MSE from 0.046 to 0.039 and raises accuracy from 55.6% to 68.5%. Video FVD and FID also improve over the original video model, whereas within-video pre-training worsens both. These results support generalization from human demonstrations to unseen robots in both action and video prediction.

Table 1: **Evaluation on held-out robot data.** End-effector action and video prediction on RoboMIND, unseen by every model; *original model* is the video model before any action training. Other sets are in §E.2.

	Action		Generated video	
	MSE ↓	acc@0.1 ↑	FVD ↓	FID ↓
Original model	–	–	290.7	12.48
Within-video	0.046	55.6	1485.1	53.53
`WATCH` (Ours)	**0.039**	**68.5**	**154.9**	**8.33**

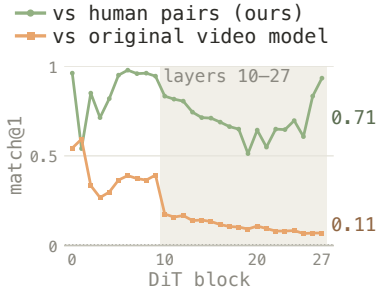


Figure 7: **Alignment with a robot-trained model.** Match@1 for human-trained `WATCH` (green) and the original video model (orange), measured against the robot-trained model on 359 EgoVerse windows (robot windows: Fig. 19). Shaded: layers 10–27, with their mean.

Human and robot pairs induce similar representations. We examine whether training on human demonstrations yields robot-motion representations similar to those learned directly from robot demonstrations. We compare `WATCH` trained on 1K hours of human pairs with a counterpart trained on 1K hours of RoboMIND (Wu et al., 2025) robot pairs using the same cross-video objective (§F.1). Both are evaluated on the same 359 EgoVerse and 300 RoboMIND windows, the latter seen during pre-training only by the robot-trained model. We align the human-trained and original video models' action-slot features to the robot-trained model's features on half of the windows, then measure cross-model window matching on the rest (§F). Human-trained `WATCH` achieves a mean match@1 of 0.71 on human and 0.50 on robot windows over layers 10–27, compared with 0.11 and 0.13 for

the original video model (Fig. 7). This correspondence supports human demonstrations as a scalable source for learning robot-relevant motion representations.

4 POST-TRAINING EXPERIMENTS

In this section, we evaluate whether the representation properties identified in our analyses translate into improved robot control after post-training on robot demonstrations. §4.2 compares WATCH with prior policies under in-context and context-free settings. §4.3 examines how the pre-training objective, reference set, supervised modalities, and amount of human video affect downstream performance on a fixed backbone.

4.1 EXPERIMENTAL SETUP

In-context benchmarks. We evaluate in-context control on MimicDroid (Shah et al., 2025) with the GR1 humanoid. The benchmark ships human demonstrations but no robot-executed trajectories or evaluation scenes: MimicGen (Mandlekar et al., 2023) scene draws and retargeting turn its three demonstrations per task into our 253 GR1 training episodes, the 100 reference demonstrations per task and the evaluation scenes (§C.5). We follow the MimicDroid L1–L3 split: L1 is the training tasks in the training scenes, L2 changes appearance and objects within the layout, and L3 holds out the layout; each episode is conditioned on a demonstration from its test setting. To probe how the policy uses the reference, we add an infeasible-demonstration setting: since human demonstrations are often much faster than the robot can execute, we replay the reference demonstrations at $2\times$ speed during both post-training and evaluation, faster than the robot can follow.

Table 2: **In-context post-training on MimicDroid.** Success rate (%); *Avg.* over L1–L3. †: reported by prior work under its own evaluation data.

Method	L1	L2	L3	Avg.
ICL baselines				
Vid2Robot (2024)	<u>90.8</u>	51.0	0.8	47.6
HAND† (2025)	29.0	21.0	1.0	17.0
MimicDroid† (2025)	59.0	44.0	26.0	43.0
VLA baselines				
$\pi_{0.5}$ (2025)	79.2	38.3	8.3	41.9
GR00T-N1.7 (2025b)	78.3	59.2	10.0	49.2
World-action models				
Cosmos Policy (2026)	75.0	29.4	18.3	40.9
+ WATCH (Ours)	**91.7**	72.5	<u>40.8</u>	<u>68.3</u>
Cosmos3 (2026)	69.2	<u>79.4</u>	30.8	59.8
+ WATCH (Ours)	**91.7**	**86.5**	**55.0**	**77.7**

Context-free benchmarks. To test whether the representations remain useful without a reference and transfer across robots, we evaluate three settings with no reference during post-training or evaluation: context-free MimicDroid on the GR1 humanoid, with L1 and L2 reported as *Easy* and *Hard*; TableTop-Sim (Im et al., 2026) with bimanual ALOHA; and RoboCasa (Nasiriany et al., 2024) Kitchen-H50 with a single-arm Franka and 50 human demonstrations per task. For RoboCasa only, Cosmos3 pre-training additionally includes RoboMIND data (§D.2).

Backbones and baselines. We use Cosmos Policy (Kim et al., 2026) on the 2B Cosmos-Predict2.5 (NVIDIA, 2025a) backbone, with the post-training objective reduced to future-video and action prediction (§D.2). We also apply WATCH to Cosmos3 Edge-2B (NVIDIA, 2026), a backbone already trained on human and robot action data under a within-video objective. In-context baselines are Vid2Robot (Jain et al., 2024), HAND (Hong et al., 2025), MimicDroid (Shah et al., 2025), $\pi_{0.5}$ (Physical Intelligence, 2025), and GR00T-N1.7 (NVIDIA, 2025b); the context-free comparison adds MolmoAct2 (Fang et al., 2026). All methods we train use the same robot demonstrations as WATCH; † marks numbers reported by prior work under its own evaluation data.

4.2 COMPARISON WITH PRIOR METHODS

In-context post-training. Table 2 compares WATCH with prior demonstration-conditioned policies (Jain et al., 2024; Hong et al., 2025; Shah et al., 2025) and vision-language-action policies (Physical Intelligence, 2025; NVIDIA, 2025b) under the MimicDroid L1–L3 split. With Cosmos Policy (Kim et al., 2026), WATCH raises the average success rate from 40.9% to 68.3%, with gains at all three levels and the largest under environment shift (L2 29.4% to 72.5%, L3 18.3% to 40.8%). The same trend holds with Cosmos3 (NVIDIA, 2026), where the average increases from 59.8% to 77.7% and L3 from 30.8% to 55.0%. Some baselines match or exceed Cosmos Policy on L1 but fall to 10% or below on L3, the unseen layouts, where WATCH gains substantially.

Table 3: **Context-free post-training.** Success rate (%) without a reference demonstration; *Avg.* over the two levels of each benchmark.

Method	MimicDroid			TableTop-Sim			RoboCasa
	Easy	Hard	Avg.	Easy	Hard	Avg.	Kitchen-H50
VLA baselines							
$\pi_{0.5}$ (2025)	78.3	29.8	54.1	70.4	34.0	52.2	54.5
GR00T-N1.7 (2025b)	79.2	45.8	62.5	66.8	50.8	58.8	46.4
MolmoAct2 (2026)	76.7	28.3	52.5	66.4	39.6	53.0	54.3
World-action models							
Cosmos Policy (2026)	71.7	26.9	49.3	69.6	49.6	59.6	56.0
+ WATCH (Ours)	**90.8**	**47.1**	**69.0**	76.8	54.0	65.4	**63.6**
Cosmos3 (2026)	78.3	21.0	49.7	76.4	55.2	65.8	55.0
+ WATCH (Ours)	**90.8**	36.0	63.4	**85.6**	**58.4**	**72.0**	55.8

Table 4: **Pre-training ablations.** Success rate (%) on a fixed Cosmos Policy backbone; rows vary the pre-training objective, data mixture, and reference set (defined in the text): in-context MimicDroid with references at recorded and at 2× speed, and context-free MimicDroid, TableTop-Sim, and RoboCasa Kitchen-H50. *None* is the backbone post-trained directly (Kim et al., 2026).

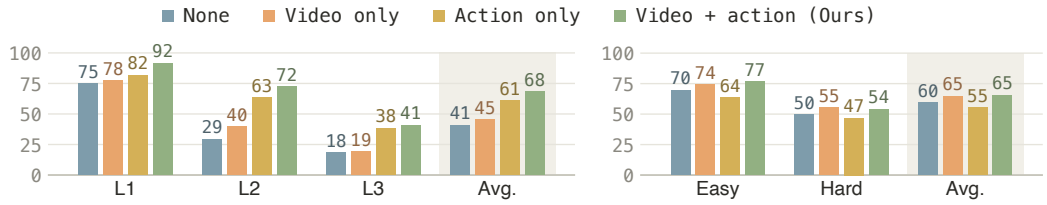
Pre-training	In-context								Context-free						
	MimicDroid				MimicDroid, 2×				MimicDroid			TableTop-Sim			Robo-Casa
	L1	L2	L3	Avg.	L1	L2	L3	Avg.	Easy	Hard	Avg.	Easy	Hard	Avg.	
None (2026)	75.0	29.4	18.3	40.9	73.3	18.8	0.0	30.7	71.7	26.9	49.3	69.6	49.6	59.6	56.0
Within-video (2025; 2025)	85.8	44.4	23.3	51.2	75.0	7.5	8.3	30.3	88.3	40.8	64.6	71.6	51.6	61.6	58.3
Co-training (2026; 2026)	87.5	58.3	34.2	60.0	**88.3**	25.2	5.0	39.5	90.0	42.3	66.1	69.2	52.8	61.0	61.7
Same-source top-1	84.2	58.3	33.3	58.6	85.8	26.7	23.3	45.3	90.0	36.7	63.3	69.6	51.2	60.4	56.8
WATCH (Ours)	**91.7**	**72.5**	**40.8**	**68.3**	84.2	**36.5**	**41.7**	**54.1**	**90.8**	**47.1**	**69.0**	**76.8**	**54.0**	**65.4**	**63.6**

Context-free post-training. Table 3 evaluates standard behavior-cloning policies without reference demonstrations during either post-training or evaluation. On MimicDroid, WATCH improves Cosmos Policy from 49.3% to 69.0% and Cosmos3 from 49.7% to 63.4%. The benefit extends to TableTop-Sim (bimanual ALOHA), where both backbones improve, and to RoboCasa (single-arm Franka), where Cosmos Policy gains 7.6 points while Cosmos3, which already saw robot data and here RoboMIND, gains 0.8. Gains with no reference during robot post-training and evaluation show that the benefit of pre-training persists without a test-time reference.

4.3 Ablation Studies

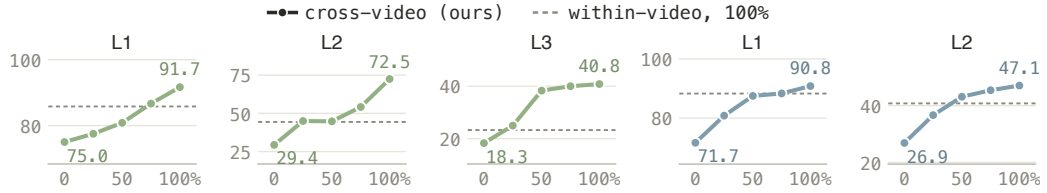
All variants keep the Cosmos Policy (Kim et al., 2026) backbone, the robot post-training, and the human-video data fixed. *Within-video* (Yang et al., 2025; Luo et al., 2025) predicts the future actions and frames of each demonstration from its own observation, without a reference. *Co-training* (Punamiya et al., 2026; Zheng et al., 2026) keeps that formulation and adds robot action data from the robot subset of EgoVerse and Humanoid Everyday (Zhao et al., 2025). Finally, *same-source top-1* conditions every query on its single best-ranked reference from its own collection source, whereas WATCH (Ours) samples from the full reference set of §2.3.

Ablation on pre-training objective. Table 4 shows that within-video pre-training improves over no pre-training and co-training improves further on MimicDroid, but both leave the representation tied to cues within individual demonstrations. By predicting across demonstrations, WATCH encourages features that capture shared interactions, consistent with its larger gains on L2 and L3 and its advantage on all three context-free benchmarks. Under 2×-speed references, within-video pre-training loses its whole gain (30.3% against 30.7% with none) whereas WATCH keeps most of its own, so the cross-video prior still supports learning from references the robot cannot follow. Together, these results suggest that cross-demonstration prediction, not data diversity alone, turns diverse human experience into more embodiment-invariant representations.



(a) MimicDroid, in-context (b) TableTop-Sim, context-free

Figure 8: **Pre-training supervision.** Success rate (%) with post-training fixed at video + action and pre-training supervising nothing (*None*), the video only, the actions only, or both, on (a) MimicDroid in-context, averaged over the four tasks per level with *Avg.* over L1–L3, and (b) TableTop-Sim context-free.



(a) In-context (b) Context-free

Figure 9: **Scaling the pre-training data.** MimicDroid success rate (%) against the portion of the paired human video used for pre-training on a fixed Cosmos Policy backbone, from no pre-training (0) to all EgoVerse pairs (100%). (a) In-context evaluation. (b) Context-free evaluation. Dashed: within-video pre-training on the 100% EgoVerse training data.

Ablation on pre-training modalities. We evaluate the modality variants in §3, fixing post-training to joint action–video prediction (Fig. 8 and §G.4). Action-only pre-training outperforms video-only on in-context MimicDroid, particularly on L2 and L3, while the reverse holds on context-free TableTop-Sim. MimicDroid preserves the pre-training action space and reference conditioning, whereas TableTop-Sim uses a different action space and no reference input. Joint pre-training achieves the highest average success in both settings, with a small advantage over video-only on TableTop-Sim (65.4 versus 64.6). This pattern suggests that action supervision provides explicit guidance on the motion and temporal progression of interactions, yielding a prior that is readily reused when the action representation and conditioning are preserved. Video prediction complements this signal by supervising the visual consequences of motion, potentially supporting transfer across larger differences from the pre-training setup.

Scaling the pre-training data. We vary the amount of training data and its matched references while keeping the backbone and post-training setup fixed. The number of pre-training steps increases with the amount of data, so larger datasets also use more compute (§G.3). In in-context and context-free benchmarks, performance improves with more data (Fig. 9). With 25% of the cross-video data, WATCH matches within-video pre-training using all the data on the unseen in-context levels (L2 and L3). It surpasses this baseline at 50% on L3 and 75% on L2. Under context-free post-training, it exceeds the baseline on L2 at around 50%. These trends indicate that WATCH benefits from more paired human videos. Our comparison with robot-pair pre-training further suggests that collecting human videos can be a more cost-effective way to improve performance than collecting robot videos (§F.3).

5 Conclusion

We presented WATCH, a cross-video pre-training stage that turns task-paired human demonstrations into a prior over motion and interaction for world-action models, without matched human–robot data. Because the model must predict one demonstration from another, it is trained toward how a task is executed rather than how a demonstration looks: the learned features align the same motion across people, rigs, and robots, attend to the hand and the object, and place robot frames where a model trained on robot data places them. This prior carries over to control: across two backbones and three benchmarks it improves both in-context and context-free post-training, most where the scene departs from the demonstration, and it grows with paired human video, which is far cheaper to collect than robot data; extending the pairing beyond one human dataset to less structured sources is future work.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish writing, to draft sections of the paper, for retrieval and discovery of related work, for research execution, and for checking mathematical claims. Drafted passages were rewritten by the authors and checked against the experimental records; retrieved references were read and verified before being cited; AI-assisted analysis and figure code was inspected, and its outputs were cross-checked against the logged results by at least one author; and derivations and statistical tests, including the pairing distance of §C.2 and the tests of §3, were re-verified by hand. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We will release the code for pair construction, pre-training, post-training, and evaluation, together with the pairing manifests, the generated MimicDroid GR1 training episodes and evaluation scene manifests, and the pre-trained checkpoints. The appendix documents what the release implements: the action and state representation and the pair construction with its distance, descriptor, and reference-set rules (§C.1 and §C.2), the corpus and held-out splits (§C.3 and §C.4), the MimicDroid robot data, evaluation tiers, and episode counts (§C.5), the architecture, hyperparameters, schedules, and inference settings (§D.1 and §D.2), the pre-training variants used as baselines (§D.3), the protocol of the representation analyses (§E.5), and the baseline recipes and per-task results behind every table (§G.1 and §G.2).

REFERENCES

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Arpit Bahety, Priyanka Mandikal, Ben Abbatematteo, and Roberto Martín-Martín. ScrewMimic: Bimanual imitation from human videos with screw space projection. In *Proc. of the Robotics: Science and Systems (RSS)*, 2024.

Andreas Blattmann, Robin Rombach, Kaan Oktay, Jonas Müller, and Björn Ommer. Semi-parametric neural image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, et al. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning (ICML)*, 2022.

Haoquan Fang, Jiafei Duan, Donovan Clay, Sam Wang, Shuo Liu, Weikai Huang, Xiang Fan, Wei-Chuan Tsai, Shirui Chen, Yi Ru Wang, Shanli Xing, Jaemin Cho, Jae Sung Park, Ainaz Eftekhar, Peter Sushko, Karen Farley, Angad Wadhwa, Cole Harrison, Winson Han, Ying-Chun Lee, Eli VanderBilt, Rose Hendrix, Suveen Ellawela, Lucas Ngoo, Joyce Chai, Zhongzheng Ren, Ali Farhadi, Dieter Fox, and Ranjay Krishna. MolmoAct2: Action reasoning models for real-world deployment, 2026. URL https://arxiv.org/abs/2605.02881.

Letian Fu, Huang Huang, Gaurav Datta, Lawrence Yunliang Chen, William Chung-Ho Panitch, Fangchen Liu, Hui Li, and Ken Goldberg. In-context imitation learning via next-token prediction. *arXiv preprint arXiv:2408.15980*, 2024.

Quentin Garrido, Tushar Nagarajan, Basile Terver, Nicolas Ballas, Yann LeCun, and Michael Rabbat. Learning latent action world models in the wild. In *Proc. of the Forty-third International Conference on Machine Learning (ICML)*, 2026. URL https://openreview.net/forum?id=HDf5semiaB.

Matthew Hong, Anthony Liang, Kevin Kim, Harshitha Rajaprakash, Jesse Thomason, Erdem Bıyık, and Jesse Zhang. Hand me the data: Fast robot adaptation via hand path retrieval. *arXiv preprint arXiv:2505.20455*, 2025.

Ryan Hoque, Peide Huang, David Yoon, Jian Zhang, et al. EgoDex: Learning dexterous manipulation from large-scale egocentric video. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2026.

Hokyun Im, Euijin Jeong, Andrey Kolobov, Jianlong Fu, and Youngwoon Lee. TwinVLA: Data-efficient bimanual manipulation with twin single-arm vision-language-action models. In *Proc. of the Fourteenth International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=jG9W6nAwVz.

Vidhi Jain, Maria Attarian, Nikhil J Joshi, Ayzaan Wahid, Danny Driess, Quan Vuong, et al. Vid2Robot: End-to-end video-conditioned policy learning with cross-attention transformers. In *Robotics: Science and Systems (RSS)*, 2024.

Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. EgoMimic: Scaling imitation learning via egocentric video. In *Proc. of the 2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13226–13233. IEEE, 2025a.

Simar Kareer, Karl Pertsch, James Darpinian, Judy Hoffman, Danfei Xu, Sergey Levine, Chelsea Finn, and Suraj Nair. Emergence of human to robot transfer in vision-language-action models. *arXiv preprint arXiv:2512.22414*, 2025b.

Hanjung Kim, Jaehyun Kang, Hyolim Kang, Meedeum Cho, Seon Joo Kim, and Youngwoon Lee. UniSkill: Imitating human videos via cross-embodiment skill representations. In *Proc. of the 9th Annual Conference on Robot Learning (CoRL)*, 2025.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, et al. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, 2024.

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. In *Proc. of the Fourteenth International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=wPEIStHxYH.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. In *Proc. of the Robotics: Science and Systems (RSS)*, 2025.

Jingwei Liu, Ling Yang, Hongyan Li, and Shenda Hong. Retrieval-augmented diffusion models for time series forecasting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Hao Luo, Yicheng Feng, Wanpeng Zhang, Sipeng Zheng, Ye Wang, Haoqi Yuan, Jiazheng Liu, Chaoyi Xu, Qin Jin, and Zongqing Lu. Being-H0: vision-language-action pretraining from large-scale human videos. *arXiv preprint arXiv:2507.15597*, 2025.

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. MimicGen: A data generation system for scalable robot learning using human demonstrations. In *Proc. of the 7th Annual Conference on Robot Learning (CoRL)*, 2023.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. RoboCasa: Large-scale simulation of everyday tasks for generalist robots. In *Proc. of the Robotics: Science and Systems (RSS)*, 2024.

NVIDIA. Cosmos world foundation model platform for physical AI. *arXiv preprint arXiv:2501.03575*, 2025a.

NVIDIA. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025b.

NVIDIA. Cosmos 3: Omnimodal world models for physical AI. *arXiv preprint arXiv:2606.02800*, 2026.

Georgios Papagiannis, Norman Di Palo, Pietro Vitiello, and Edward Johns. R+X: Retrieval and execution from everyday human videos. In *Proc. of the 2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8284–8290. IEEE, 2025.

Jeongeun Park, Juhan Park, Taekyung Kim, Sungjoon Choi, Dongyoon Han, and Sangdoo Yun. Retrieve, don't retrain: Extending vision language action models to new tasks at test time. *arXiv preprint arXiv:2606.15631*, 2026.

Physical Intelligence. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Ryan Punamiya, Dhruv Patel, Patcharapong Aphiwetsa, Pranav Kuppili, Lawrence Y. Zhu, Simar Kareer, Judy Hoffman, and Danfei Xu. EgoBridge: Domain adaptation for generalizable imitation from egocentric human data. In *Proc. of the Thirty-ninth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2025. URL https://openreview.net/forum?id=FGMBxzpgis.

Ryan Punamiya, Simar Kareer, Zeyi Liu, Josh Citron, Ri-Zhao Qiu, Xiongyi Cai, et al. EgoVerse: An egocentric human dataset for robot learning from around the world. *arXiv preprint arXiv:2604.07607*, 2026.

Pierre Sermanet, Tianli Ding, Jeffrey Zhao, Fei Xia, Debidatta Dwibedi, Keerthana Gopalakrishnan, Christine Chan, Gabriel Dulac-Arnold, Sharath Maddineni, Nikhil J Joshi, et al. RoboVQA: Multimodal long-horizon reasoning for robotics. In *Proc. of the 2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 645–652. IEEE, 2024.

Rutav Shah, Shuijing Liu, Qi Wang, Zhenyu Jiang, Sateesh Kumar, Mingyo Seo, Roberto Martín-Martín, and Yuke Zhu. MimicDroid: In-context learning for humanoid robot manipulation from human play videos. *arXiv preprint arXiv:2509.09769*, 2025.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, et al. RoboMIND: Benchmark on multi-embodiment intelligence normative data for robot manipulation. In *Proc. of the Robotics: Science and Systems (RSS) 2025*, 2025. URL https://www.roboticsproceedings.org/rss21/p152.pdf.

Xionghao Wu, Yijun Yang, Shiyang Zhou, Haoze Sun, Jianhui Liu, Songsong Yu, Jiyao Zhang, Wenbo Li, Bo Wang, Guoqing Ma, et al. ZimaBlue: Evolving generalizable world action models through scalable video pre-training. *arXiv preprint arXiv:2609.00188*, 2026.

Sicheng Xie, Haidong Cao, Zejia Weng, Zhen Xing, Haoran Chen, Shiwei Shen, et al. Human2Robot: Learning robot actions from paired human-robot videos. *arXiv preprint arXiv:2502.16587*, 2025.

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

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, et al. Latent action pretraining from videos. In *International Conference on Learning Representations (ICLR)*, 2025.

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

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

Zhenyu Zhao, Hongyi Jing, Xiawei Liu, Jiageng Mao, Abha Jha, Hanwen Yang, Rong Xue, Sergey Zakharov, Vitor Guizilini, and Yue Wang. Humanoid Everyday: A comprehensive robotic dataset for open-world humanoid manipulation. *arXiv preprint arXiv:2510.08807*, 2025.

Ruijie Zheng, Dantong Niu, Yuqi Xie, Jing Wang, Mengda Xu, Yunfan Jiang, et al. EgoScale: Scaling dexterous manipulation with diverse egocentric human data. *arXiv preprint arXiv:2602.16710*, 2026.

Jiaming Zhou, Qihang Zhang, Gangwei Xu, Cunxin Fan, Yujie Zhao, Ruilin Wang, Yiming Luo, Shuai Yang, Xing Zhu, Yujun Shen, et al. Zero-WAM: In-context world-action modeling from human videos for open-ended task generalization. *arXiv preprint arXiv:2608.26103*, 2026.

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 *Proc. of the Robotics: Science and Systems (RSS)*, 2025.

APPENDIX

The appendix is organized as follows.

- §A reviews related work; §B states the limitations of this study.
- §C describes the data: the action and state representation (§C.1), pair construction (§C.2), pairing statistics (§C.3), the held-out sets (§C.4), the MimicDroid robot data and evaluation protocol (§C.5), and pair examples (§C.6).
- §D gives implementation details: the architecture (§D.1), training and inference settings (§D.2), and the pre-training variants used as baselines (§D.3).
- §E extends the analyses of §3: more frames for the motivation (§E.1), prediction on more held-out data (§E.2), the swapped-reference attention on more queries (§E.3), attention on robot and human frames (§E.4), the protocol of the representation analyses (§E.5), and additional probing results (§E.6).
- §F compares human video with robot data as pre-training data: the setup (§F.1), the representation and attention analyses (§F.2), and downstream transfer from the two arms (§F.3).
- §G collects the baseline recipes (§G.1), per-task results (§G.2), the data scaling ladder (§G.3), and the supervision ablation with the post-training target varied as well (§G.4).
- §H shows qualitative rollouts.

A RELATED WORK

Pre-training for robot policies. Pre-training on large-scale, cross-embodiment robot datasets enables generalist policies to acquire diverse skills and transfer across embodiments (Kim et al., 2024; NVIDIA, 2025b; Physical Intelligence, 2025), but scaling this experience remains constrained by robot data collection. Human videos offer additional experience through inferred latent actions (Ye et al., 2025; Garrido et al., 2026) or explicit motion supervision (Punamiya et al., 2026; Kareer et al., 2025a). Recent approaches co-train on human and robot data to ground this experience in executable actions (Punamiya et al., 2025; Kareer et al., 2025b). Video-based models also incorporate action learning before target-robot adaptation: Cosmos3 introduces dynamics and policy mid-training (NVIDIA, 2026), while ZimaBlue aligns video and action on heterogeneous robot trajectories after robot and human video pre-training (Wu et al., 2026). These co-training and alignment stages require robot data before target-domain adaptation. `WATCH` instead uses task-matched human demonstrations alone during its proposed pre-training stage, followed by post-training on a small amount of target-domain data.

Demonstration-conditioned robot learning. One- and few-shot imitation learning enables task adaptation from a small number of robot demonstrations (Fu et al., 2024) or human videos (Kim et al., 2025; Jain et al., 2024; Xie et al., 2025; Shah et al., 2025; Bahety et al., 2024; Papagiannis et al., 2025; Park et al., 2026). Most closely related, Zero-WAM learns from automatically constructed human-to-robot pairs to execute unseen tasks conditioned on human video prompts (Zhou et al., 2026). These approaches establish demonstrations as an effective interface for task adaptation; `WATCH` explores their role in representation learning through human-to-human pre-training. The learned representations support both in-context control and reference-free behavior cloning after robot post-training. This perspective draws on reference-conditioned learning in language modeling, image synthesis, and time-series forecasting (Borgeaud et al., 2022; Blattmann et al., 2022; Liu et al., 2024), and examines its implications for transferable representations in robot foundation models.

B LIMITATIONS

A shared Cartesian action space. Human and robot actions are written in one Cartesian task space, the end-effector pose and fingertip positions of §C.1, expressed in a canonical frame fixed at the start of each episode. Three parts of the method rely on this choice. The pairing distance (§C.2) compares wrist trajectories and hand terms across demonstrations, which is meaningful only because a human wrist and a robot wrist are described by the same quantities. The reference slots condition the model on the reference's actions in that space, so the action prior is a prior over end-effector motion. And

post-training reuses the same action representation, which is why a human-trained action head can be warm-started on a robot at all. A joint-space action target breaks all three at once: a human demonstration has no counterpart to a robot's joints, so neither the pairing nor the action prior carries over, and the prior would have to be re-expressed, through a Cartesian intermediate target that a low-level controller tracks, or through a learned map between joint and end-effector motion. All three post-training targets in this paper, including the 20-d bimanual ALOHA space, are Cartesian, so the method covers end-effector controllers as they are commonly deployed but has not been shown for joint-space policies. The canonical frame carries a second assumption: the frame is anchored once per episode, so a mobile base or a moving camera would need the frame to be re-anchored or the motion to be expressed relative to the current view.

Pairing from a task taxonomy and tracked hands. Candidates are gated by a task class and ranked by a descriptor built from the hand representation (§C.2), so pairing needs task labels and hand tracking; unlabeled video, or video without hand pose, needs a label-free gate and a pose estimator before it can be paired. The gate also fixes the granularity of a pair: a class that mixes distinct motions admits pairs that agree in task but not in motion, and the ranking has to separate them, which it does well beyond same-class random pairs (§C.3) but not perfectly. The ranking compares motion shape rather than magnitude, so a reference tells the model what to do but not how far to move, and the extent has to come from the query's own frame and state.

Room to scale. Pre-training uses paired windows from a single human dataset, and the scaling curve (§4.3) has not saturated: a quarter of the corpus already matches within-video pre-training on the full corpus on the unseen levels, and every level is still rising at the full corpus. More paired human video from more collection sources is therefore the first lever, and the pairing places no limit on how many sources it can span. Robot data is the second: cross-video prediction applies to robot pairs as it does to human pairs (Fig. 7), so adding robot demonstrations where they exist should add to the prior rather than replace it; whether robot data helps more when it enters through the same cross-video objective than alongside it, as in co-training, is a direct comparison we leave to future work.

C Dataset Details

C.1 Action and State Representation

Layout. The 57-dimensional vector is laid out as

$$\underbrace{\text{head}_{9}}_{0:9} \;\big|\; \underbrace{\text{eef}^{L}_{9}}_{9:18} \;\big|\; \underbrace{\text{hand}^{L}_{15}}_{18:33} \;\big|\; \underbrace{\text{eef}^{R}_{9}}_{33:42} \;\big|\; \underbrace{\text{hand}^{R}_{15}}_{42:57}.$$

A pose term is $[\,p,\ R_{:,1},\ R_{:,2}\,]$, a translation followed by the first two columns of the rotation matrix. A hand term is five fingertip positions expressed in the wrist frame, in meters, which describes a grasp without committing to a hand model. For a parallel gripper the same term is synthesized by interpolating between a fully open and a fully closed fingertip template with the normalized gripper command, so a gripper and a hand occupy the same coordinates.

Frames. For human demonstrations the canonical frame is placed at the waist at the first frame of the episode, with its heading taken from the mean head heading over the opening frames and its vertical axis aligned to gravity, and is held fixed for the whole episode. Expressing the absolute form in the moving head frame would instead make a stationary object appear to move whenever the wearer turns. For robot demonstrations the pose terms are base-relative and the head term is constant, since the cameras do not move with the arms. Each source's channels are normalized by robust per-channel statistics estimated once over its own windows.

C.2 Pair Construction

Distance. Within a task-taxonomy group, candidates are ranked by

$$d(q,r) = \min_{\pi \in \Pi_b} \frac{1}{|\pi|} \sum_{(i,j)\in\pi} \left\| \frac{\phi_i(q) - \mu(q)}{\sigma(q)} - \frac{\phi_j(r) - \mu(r)}{\sigma(r)} \right\|_W + \lambda_h \max(0, |h(q) - h(r)| - \tau), \tag{2}$$

where $\phi_{1:T_d}(w)$ is the per-frame motion descriptor of window w and $\mu(w)$ and $\sigma(w)$ are its channel-wise mean and standard deviation over the window's T_d frames, the division taken per channel. This standardization removes the window's offset and reduces the sensitivity of the comparison to motion magnitude, so that executions of different reach are compared by the shape of their motion; it is not a strict invariance, since the position, rotation and hand channels do not all scale with reach. The first term warps the two motions against each other within a band $|i - j| \leq b$, and $h(w)$, the left hand's share of the total reach, separates a left-handed execution from a right-handed one, with a hinge at τ so that small differences in hand use go unpenalized.

Descriptor. Per frame and per hand, the descriptor concatenates the start-relative position (3), the 6D rotation (6), the action increment of the end effector (9) and the hand term (15), giving 33 channels per hand and 66 per frame. The sequence is subsampled to $T_d = 40$ steps before matching, and the weights W of Eq. 2 down-weight the fingertip channels relative to the wrist so that the trajectory rather than the finger detail drives the alignment: the fingertip channels carry a weight of 0.25 relative to the wrist. The band is $b = 6$ steps, and the handedness term uses $\lambda_h = 3.0$ with threshold $\tau = 0.25$; it separates a left-handed from a right-handed execution of the same task, and path length is not used for this, since it does not distinguish the two.

Reference set. Candidates are first gated to the query's task class in a canonical taxonomy, falling back to the class's super-group only when no other source recorded that class. Within the gate, a query keeps its 8 nearest references from the same collection source and up to 7 from other sources, a cross-source candidate being admitted whenever its distance is within 1.6 times the best score, for at most 15 references per query (the manifest reserves 20 slots). The full build gives $1{,}092{,}145$ windows and $1{,}045{,}007$ query windows with an average of 10.5 references each (median 11, 4 to 15 between the 5th and 95th percentiles), 4.2 from the same source and 6.4 from others; no query is without a reference, and 9.3% have no cross-source one: queries the super-group fallback does not reach, either because no other source recorded their super-group or because no cross-source candidate fell inside the distance gate. They are all MECKA queries, whose bridging rate is 82.4% against 100% for Scale, Aria and the play set. At each training step one reference is drawn uniformly from the query's whole set, from any source, so a query with cross-source references is conditioned on one of them in proportion to their share of its set.

C.3 Data Statistics

Table 5: **Within a task class, the retrieved reference agrees with the query's motion.** Agreement between a query and its top-ranked cross-source reference on motion-agreement metrics that the pairing score does not directly optimize (the descriptor sees positions, rotations and action increments, but is scored by warped descriptor distance, not by these metrics), against a random partner of the same task class and a random partner. Means over $12{,}000$ queries.

	Retrieved	Same-class null	Random null
speed-profile correlation ↑	**0.23**	−0.01	−0.01
trajectory-shape cosine ↑	**0.36**	0.00	0.00
| log | path-length ratio ↓	**0.69**	0.93	0.92
speed off by more than 3× ↓	**22.9%**	28.7%	39.4%

Corpus. Pre-training queries come from the three human collection sources of EgoVerse, MECKA, Scale and Aria, all at 30 Hz in the 57-d convention of §C.1, together with the MimicDroid play at 20 Hz, which is human as well: hand-tracked play by people in the sink environment, with no robot

in the loop (§C.5). The manifest holds 14,137 episodes, 13,315 of which carry query windows (the rest hold reference windows only), and 1,045,007 query windows, of which 551,337 are MECKA, 367,823 Scale, 109,652 Aria and 16,195 play, each query carrying up to 20 reference slots, so every query is a human demonstration. The robot subset of EgoVerse and Humanoid Everyday, both robot data, are listed as windows but serve as neither query nor reference here; EgoDex is in no training manifest at all.

Windows and totals. Pre-training windows span about 3 seconds: 92 frames at 30 Hz for the three EgoVerse sources and 60 frames at 20 Hz for the 20 Hz sources, the MimicDroid play and the robot windows, the lengths that pack with the current frame into the video autoencoder's $1 + 4(T - 1)$ frames (§D.1) and that the action chunk of §D.2 follows. Windows are drawn at a stride of 30 frames, with at most two windows per episode among a query's references, and the query's own episode is excluded from its references except in the MimicDroid play, whose twenty episodes are its only source; its 19,802 links are the only same-episode links in the build. The snapshot we pair holds 14,137 episodes of the 1.36K-hour EgoVerse release, with the release's robot subset left out (it enters only under co-training, §D.3). At the 1-second stride the 1,045,007 query windows amount to 871 hours of 3-second windows; with the one reference drawn per query at each step, an epoch covers 1,742 hours, and the full set of 10.99M query–reference links 10.0K hours. These are window-hours: each query window and each reference link counts as one 3-second window, and the hours of §F are counted the same way.

Pairing statistics. The build holds 10.99M query-reference links. They cover the corpus rather than a corner of it: 90.6% of the episodes (68.0% of the windows) serve as a reference at least once, a query's 20 slots span 2.0 sources and 10.4 episodes on average, and the reference stream the model actually sees, weighted by how often each window is drawn, retains 95% of the pool's motion diversity (247.8 against 260.5 effective motions under an RBF Vendi score on the descriptor). The DTW rerank adds motion fidelity on top of the taxonomy gate, which Table 5 scores on additional motion-agreement metrics that the pairing score does not directly optimize, against a null that draws a random partner of the same class. There are no self-matches, no overlapping windows between a query and its references, and no episode shared between training queries and the held-out set.

C.4 Held-Out Data

Split. The held-out set is 150 episodes, stratified over task and source and taken at the level of whole episodes, so no window of an evaluation episode is used as a training query.

Held-out sets. Two sets are kept out of training. (i) 150 whole episodes of the training sources (75 MECKA, 67 Scale, 8 Aria), stratified over task and source, are removed from the manifest and become the held-out EgoVerse queries (13,369 windows) behind the analyses of §3; their references are retrieved from the training pool. (ii) EgoDex (Hoque et al., 2026), recorded on a different rig with tracked fingers and converted to the same 57-d convention at 30 Hz, is the held-out human source of Table 6: it enters no training manifest, is paired only against itself by window-level top-3 retrieval, and carries its own normalization. Table 1 and Table 6 use 2,048 clips per set.

Analysis sets. The analyses of §3 are drawn from the 150 held-out episodes. The conditioning test (Fig. 4) needs, for each query, a matched demonstration from a different source and a different-task demonstration from the same source, and keeps the 137 episodes for which both exist. The representation analyses (Fig. 5) need pairs of queries that perform the same motion, and define motion similarity through the training retrieval itself: two held-out queries are a pair when their top-20 references share a demonstration, which is the retrieval descriptor's own notion of the same motion. Each query is then read with its own references, so neither member of a pair enters the other's input and nothing leaks. Restricting to tasks that at least two sources recorded, since a task recorded by a single source confounds task with identity, leaves 122 episodes, 359 action windows and 120 cross-source pairs, on which the human-set retrieval and probing results are computed; the t-SNE of §E.5 draws 95 of these windows, from three of the tasks that two of the three sources recorded. The representation analyses are also run on a robot window set that is not drawn from EgoVerse: 396 RoboMIND windows from three robots and six tasks, with windows of the same length and pairs matched by trajectory distance within a task, described in §E.5 and illustrated in Fig. 11.

C.5 MIMICDROID ROBOT DATA

What the release provides. MimicDroid ships the human side of the benchmark: hand-tracked human play in the sink environment (20 episodes at 20 Hz), which is the play in the pre-training manifest of §C.3, and three canonical demonstrations per task, twelve in all. None of it is robot-executed: the release ships no robot training episodes, so the GR1 training set, the retrieval pool and the evaluation scenes below are ours.

From human demonstrations to robot episodes. Every GR1 training episode comes from one of those twelve human demonstrations, augmented in the MimicGen manner and then retargeted. A scene is drawn by a seeded environment reset in a (layout, style) cell; one of the task's three human demonstrations is retargeted into that scene, and the retargeted trajectory is replayed on the GR1 humanoid, the episode being kept only if the replay succeeds. So the three demonstrations per task are the only human motions in the set, and the scene draws are what multiply them into 253 robot episodes. Retargeting carries the demonstrated two-hand end-effector path into the drawn scene and leaves the arm configuration to the controller: the replay commands absolute base-frame poses of the two end effectors together with their gripper signals, which an operational-space controller resolves on the humanoid. Each episode is an open-loop approach to the demonstration's start at 1 cm per frame, a hand-off blend, and then the demonstration replayed unaltered, with success judged once at the end; approach and blend are 30 and 10 frames for the door task, 20 and 10 for the plate task, and 0 and 16 for the cabinet and faucet tasks.

Timing around the grasp. The humanoid's gripper cannot close as fast as the demonstrated hand and still hold the object, so the retargeted trajectory is given more time there: for the two pick-and-place tasks it holds for 12 frames before the gripper closes and stretches the closing itself over twice its recorded duration. Without this the plate task replayed successfully in 1 scene draw out of 6; with it, in 5 out of 6.

Scenes and objects. Objects spawn within 4 cm of their position in the source demonstration and are restricted to one species per task (a tomato for the cabinet task, corn for the plate task), since the recorded grip is tied to the object's size. The same feasibility filter is applied to training and evaluation scenes, so it defines the distribution rather than biasing it. One scene draw records an episode of *every* task feasible in that layout, so the scene and the proprioception alone do not determine what is to be done and only the demonstration does.

Sizes. The set holds 253 episodes over the four L1 tasks (50 TURNONFAUCET, 50 CLOSELEFTCABINETDOOR, 81 PNPSINKTOCABINET, 72 PNPSINKTORIGHTCOUNTERPLATE), each 256 to 302 frames at 20 Hz with three cameras, and a retrieval pool of 100 demonstrations per task drawn from further scene draws. A second copy of the pool replayed at double speed serves the infeasibility test of §4.1, for post-training and evaluation alike; the query side is unchanged, and the matching descriptor drops its velocity channels, which the faster replay would otherwise double. Windows span 36 frames at 20 Hz, the 32-step action chunk of §D.2 plus four frames of headroom so that the reader can round the span down to the video autoencoder's $1 + 4(T - 1)$ lengths (§D.1) as the pre-training windows do (§C.3); they are taken at every time step, giving 61,686 query windows.

Pairing. Each query window carries six demonstration slots. Slot 0 is the episode's own canonical demonstration, aligned to the query by a global DTW path over the whole episode, which is the conditioning the one-shot evaluation supplies at every chunk; slots 1 to 5 are the five nearest pool windows under banded DTW on a 26-d two-hand pose descriptor of the preceding 32 frames, at most one per episode. Because robot and demonstration share units and a frame, this matching runs on absolute poses and needs none of the per-window normalization of Eq. 2. Data, targets and normalization are identical under in-context and context-free post-training, which only drop the reference slots; the action targets of the two readers agree to the bit.

Evaluation. L2 holds out the style and the objects within the training layout (PNPSINKTORIGHTCOUNTERPLATE, PNPSINKTOCABINET, CLOSELEFTCABINETDOOR, CLOSERIGHTCABINETDOOR) and L3 holds out the layout (TURNONFAUCET, PNPSINKTORIGHTCOUNTERPLATE, CLOSELEFTCABINETDOOR, PNPSINKTOMICROWAVETOP). Evaluation scenes

are frozen manifests whose seed stream is disjoint from training (evaluation seeds start at $1{,}000$ and training seeds at $1{,}000{,}000$, and the runtime refuses an overlap), and neither L2 nor L3 scenes are ever trained on; the context is one canonical demonstration recorded in the held-out setting. Each task is evaluated over 30 episodes, 10 per demonstration for each of its three canonical demonstrations, except L2 PNPSINKTORIGHTCOUNTERPLATE, which has four demonstrations and 40 episodes; the success rates of Tables 2, 3 and 4 are means over these episodes. The policy acts in 32-step chunks, generated with 35 denoising steps at guidance 1.5. Under in-context post-training the demonstration window is re-retrieved at every chunk from the policy's own recent proprioception, constrained to advance monotonically, within a band of 8 to 48 frames past the previous match, and locked to one demonstration episode, so that one demonstration conditions the whole episode; context-free post-training bypasses retrieval altogether.

Post-training target. The robot arm predicts the commanded action in the controller's frame rather than the measured state. Predicting the measured state and replaying it as a command would apply the tracking lag twice, once in the demonstration and once in the rollout, and the commanded form also needs no frame conversion at execution time.

C.6 PAIR EXAMPLES

Fig. 10 shows three query–reference pairs as the pairing produces them. Each query is a 3-second window; its reference set holds the nearest windows of the same task from the query's own collection source and, by design, from different sources (§C.2). The same-source reference shares the query's rig, viewpoint and environment; the different-source reference shares only the task and the motion, with a different person, camera and scene, which is the variation the cross-video objective trains against.

D IMPLEMENTATION DETAILS

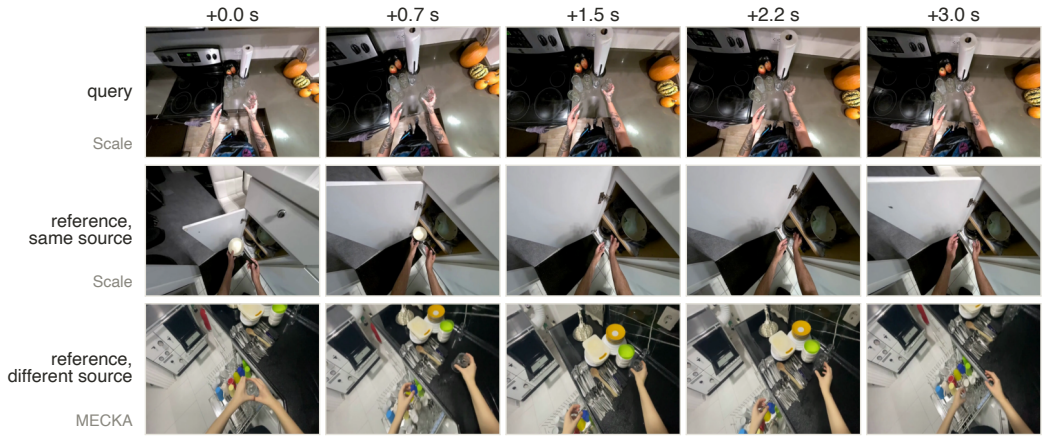
D.1 ARCHITECTURE

Slot layout. Pre-training packs six slots into the sequence, with k reference video slots and g future video slots:

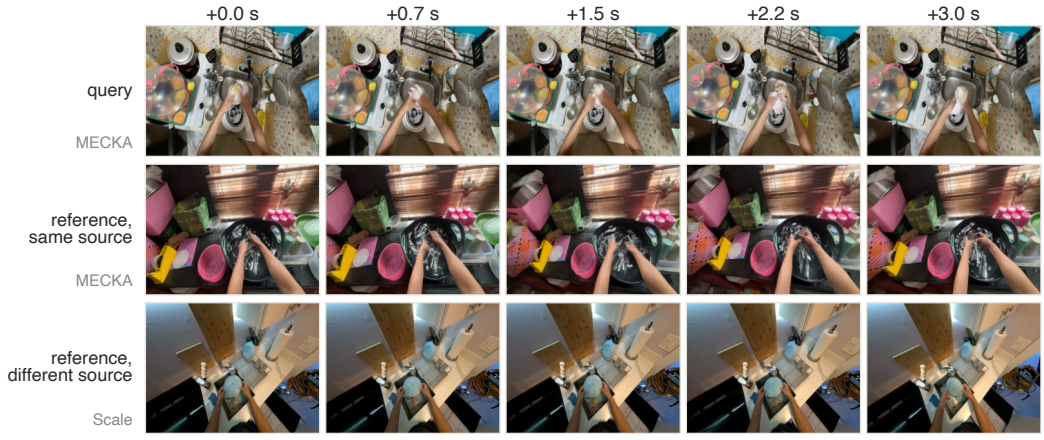
Slot	Contents	Time	Status
0	query state s^q_t	t	condition
1	reference actions $a^r_{u:u+C}$	t	condition
2	reference video $I^r_{u:u+C+1}$	$t-k\,\delta \dots t-\delta$	condition
3	query frame I^q_t	t	condition
4	query actions $a^q_{t:t+C}$	t	generated
5	query future video $I^q_{t+1:t+C+1}$	$t+\delta \dots t+g\,\delta$	generated

Query indices run over t and reference indices over u, the start of the reference window, as in Eq. 1; the Time column is the positional-time tag of each slot, which places the reference video before the query frame regardless of u. Pixels are packed as $1+4(T-1)$ frames to respect the causal video autoencoder, with black frames written into the non-pixel slots and overwritten in latent space by tiling. The query frame is placed immediately before the generation boundary so that the backbone's video continuation prior is preserved.

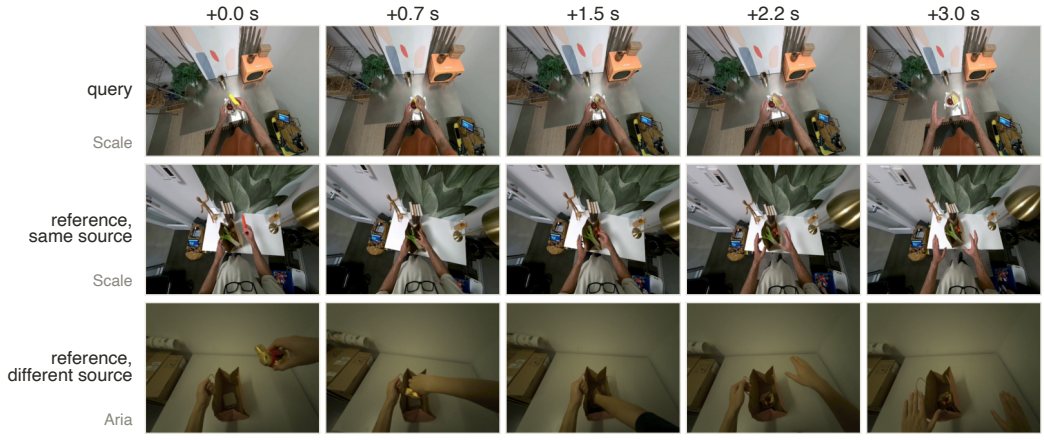
Backbone modifications. Two changes are made to the video backbone. Positional encoding is made continuous in time, taking a per-slot timestamp in seconds instead of an integer frame index, so that a slot's position no longer shifts when the number of reference or future frames changes. A slot-type embedding is added to each slot from a small table, initialized at zero so that the modified network reproduces the original one exactly at initialization; the attention is bidirectional, so a condition placed after the generated slots is still visible to them. Variable sequence lengths, which arise because human and robot windows differ in frame count, are handled by grouping samples of equal shape into a batch.



(a) Unload dishwasher



(b) Do dishes



(c) Bagging groceries

Figure 10: **Query–reference pairs.** Three queries from the training pairs, each with its reference from the same collection source and one from a different source, as five frames spanning the 3-second window. The source of every clip is named under its row label. (a) Unload dishwasher, a Scale query with a MECKA reference; (b) do dishes, a MECKA query with a Scale reference; (c) bagging groceries, a Scale query with an Aria reference.

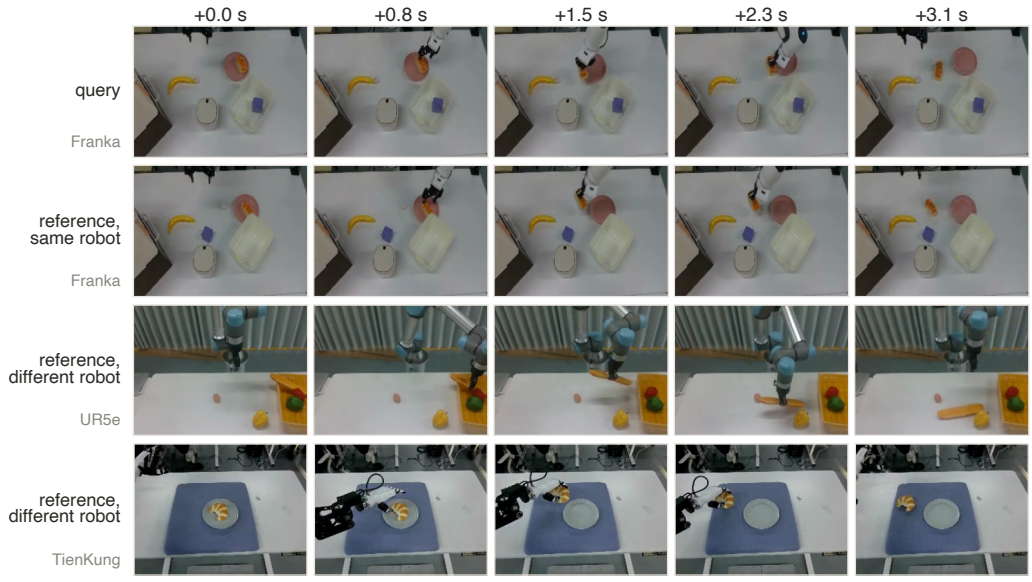


Figure 11: **A query of the robot analysis set with its nearest matches.** Bread in basket on RoboMIND: a Franka query window and the nearest same-task window, by dynamic-time-warping distance on the wrist trajectory, from the same robot and from each of the two other robots, as five frames over the 3-second window. Cross-robot pairs like the last two rows are the queries of the robot retrieval test (Fig. 5b).

Instantiation on each backbone. On the rectified-flow diffusion transformer, slots are latent frames and a condition is supplied as a clean latent excluded from the loss. On the mixture-of-transformers model there is no separate conditioning encoder: every modality is packed into one token sequence whose relations are carried by a shared 3D positional grid, the reference and the query are placed on the same temporal grid so that their frames correspond, and a condition is expressed by holding a token's noise level at zero. Actions enter through a domain-aware linear map in that case rather than by tiling. The task description ℓ of Eq. 1 is supplied as text on both backbones, through T5 cross-attention on Cosmos-Predict2.5 and as caption tokens on Cosmos3; the context-free variant drops the reference clause from the caption.

D.2 Training

Cosmos3 pre-training for RoboCasa. For the RoboCasa experiments, the Cosmos3 pre-training mixture adds two RoboMIND streams (Wu et al., 2025), Franka (80 tasks, 11,480 episodes, 1.61M frames, 10-d actions) and TienKung (32 tasks, 2,785 episodes, 4.6M frames, 20-d actions), each with three RGB views, beside the EgoVerse pairs (32.4M frames). The three sources are mixed by rank at 28:2:2, so RoboMIND is 12.5% of the training samples. Robot windows use an action chunk of 16 frames (0.53 s at 30 Hz), served without a reference, as RoboCasa is a context-free benchmark. The Cosmos3 runs for MimicDroid and TableTop-Sim use the human pairs alone.

Cosmos Policy recipe. Compared with the original Cosmos Policy recipe, our variant updates the video backbone from Predict2 to Predict2.5 and reduces the post-training objective to future-video and action prediction, without auxiliary future-proprioception prediction or rollout-data augmentation.

Hyperparameters. Pre-training uses a learning rate of 10^{-4} with a 500-step warmup and a cosine schedule (a 50K-step cycle with a floor at 0.3 of the peak) for 8K iterations, a global batch of 64 (2 per GPU on 32 GPUs); this is the full arm of the scaling ladder of §G.3, whose shorter arms scale the cycle with the data. The action weight is $\lambda = 48$. Post-training runs 8K iterations with the cycle matched to that length. Every configuration is a single training run. The action chunk is 92 frames at 30 Hz for human windows and 32 frames at 20 Hz for the MimicDroid and TableTop-Sim robot windows, which is why those windows span 36 frames (§C.5); the RoboCasa robot chunk is

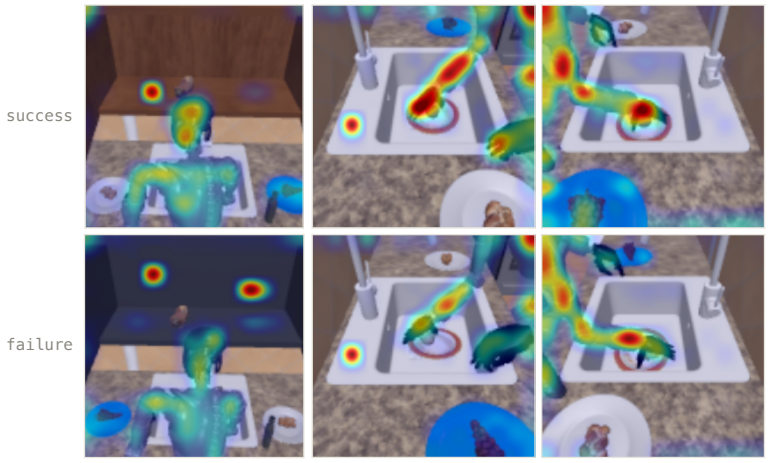


Figure 12: **The post-trained original video model, more frames.** Attention to the current frame of the two `PnPSinkToCabinet` rollouts of Fig. 2a, one frame later, from the three cameras. The success (top) attends to the hand at the object, the failure (bottom) to the forearm.

the 16 frames given above. On Cosmos-Predict2.5 the full transformer is updated; on Cosmos3 the generation experts, the time embedder, the latent and action projections, and the action modality embedding are updated at a learning rate of 2×10^{-4}, with a $5\times$ multiplier on the action heads, and the base model's action head is loaded rather than re-initialized.

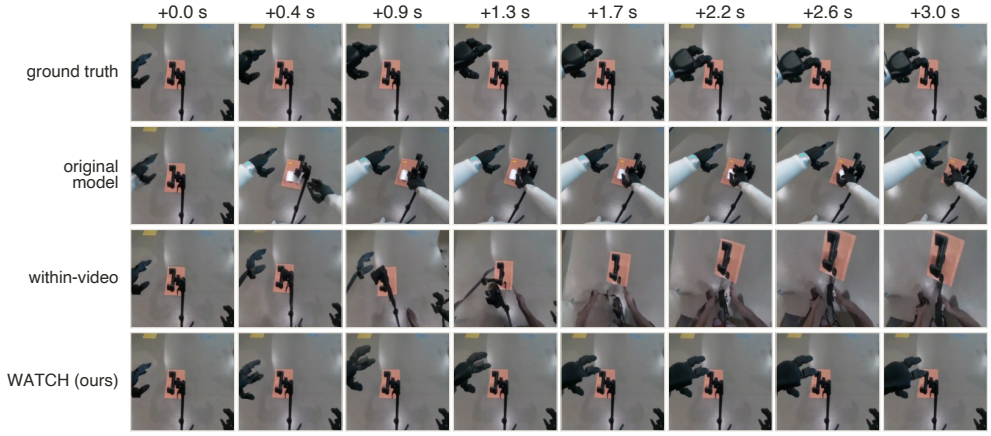
D.3 Pre-Training Variants

The within-video baseline keeps the architecture, the slot roles, the time axis and the objective of §2.3, blanks the reference video and action slots and shortens the reference video block to two slots, so the model is trained to predict each window's actions and future frames from its own observation alone; its sequence holds 29 latents against 50 for `WATCH`, so at the same 8K iterations it costs less per step. Co-training is the same blanked model with the two robot sources promoted to queries, 33,176 windows from the EgoVerse robot subset and 92,136 from Humanoid Everyday, for 1,170,319 queries in all; since the reference content is masked, adding a source needs no pairing.

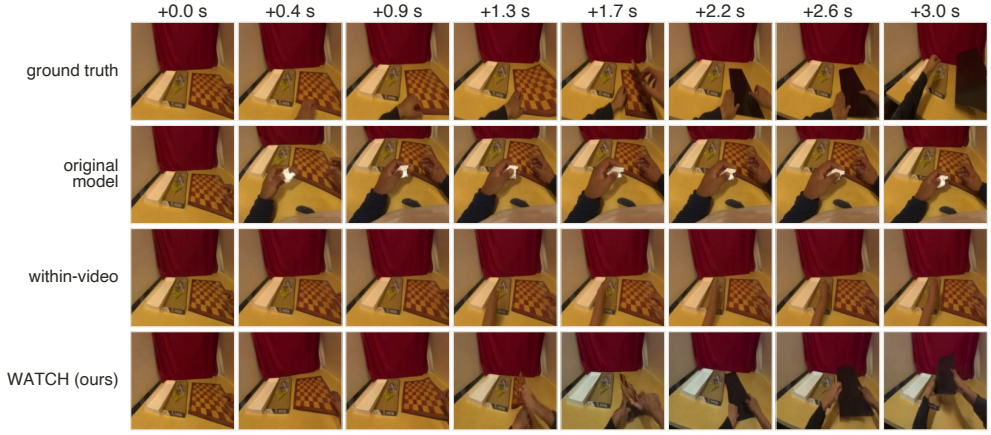
E Additional Analyses

E.1 More Frames for the Motivation

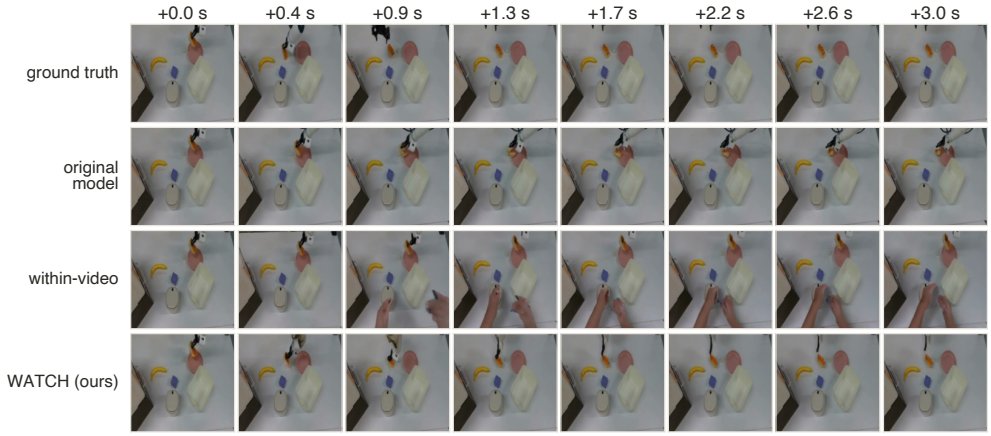
Fig. 2a shows the original video model post-trained on the robot at one frame of a successful and a failed rollout; Fig. 12 gives the next frame of the same two rollouts from all three cameras, with the same reading: the success attends to the hand at the object, the failure to the forearm. Fig. 2b and c show one human and one robot frame under the original video model and ours. Fig. 15 (§E.4) adds the within-video baseline on four more robot frames and on three human queries. The reading does not change with the frame: the backbone spreads over edges and background, within-video pre-training moves the mass onto the arm, and cross-video pre-training puts it on the hand and the object.



(a) Humanoid Everyday



(b) EgoDex



(c) RoboMIND

Figure 13: **Generated video on the held-out sets.** One clip per set, (a) Humanoid Everyday, (b) EgoDex, (c) RoboMIND, as eight frames over the 3-second generated window: the ground truth, the original video model, within-video pre-training and WATCH.

E.2 Prediction on More Held-Out Data

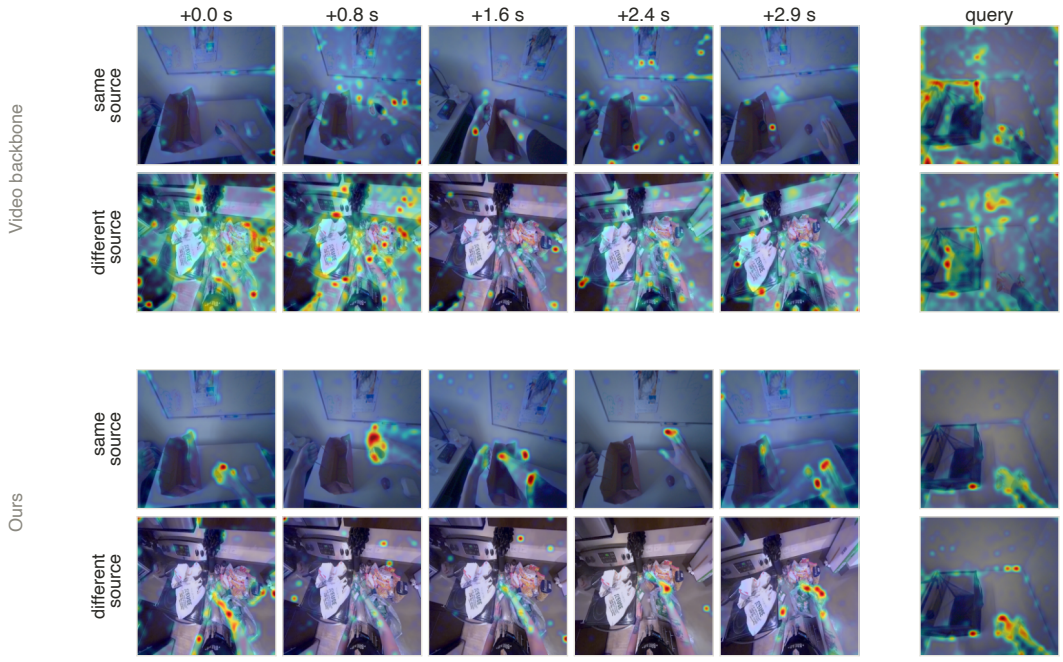
Table 6: **Prediction on three held-out sets.** Table 1 extended to Humanoid Everyday (Zhao et al., 2025), a humanoid teleoperation dataset, and EgoDex, a second human set, 2,048 clips per set, with the co-training variant of §D.3 added. Action metrics are on the end-effector subset of the action: MSE in normalization space and accuracy at tolerance 0.1. The original model generates video only. Best in bold, second underlined.

		Action (end effector)		Generated video		
Held-out set	Model	MSE ↓	acc@0.1 ↑	FVD ↓	FID ↓	LPIPS ↓
Humanoid Everyday	Original model	–	–	325.7	31.81	0.422
	Within-video	0.077	51.5	610.4	31.22	0.366
	Co-training	<u>0.045</u>	<u>66.2</u>	**105.1**	<u>12.70</u>	<u>0.267</u>
	WATCH (Ours)	**0.023**	**71.4**	<u>137.6</u>	**10.01**	**0.215**
EgoDex	Original model	–	–	200.5	**8.18**	0.243
	Within-video	<u>0.024</u>	<u>69.0</u>	<u>169.8</u>	9.82	<u>0.242</u>
	Co-training	0.033	65.8	220.1	14.17	0.299
	WATCH (Ours)	**0.019**	**73.4**	**142.7**	<u>8.53</u>	**0.211**
RoboMIND	Original model	–	–	<u>290.7</u>	<u>12.48</u>	<u>0.247</u>
	Within-video	<u>0.046</u>	55.6	1485.1	53.53	0.370
	Co-training	0.056	<u>56.7</u>	418.7	31.10	0.291
	WATCH (Ours)	**0.039**	**68.5**	**154.9**	**8.33**	**0.154**

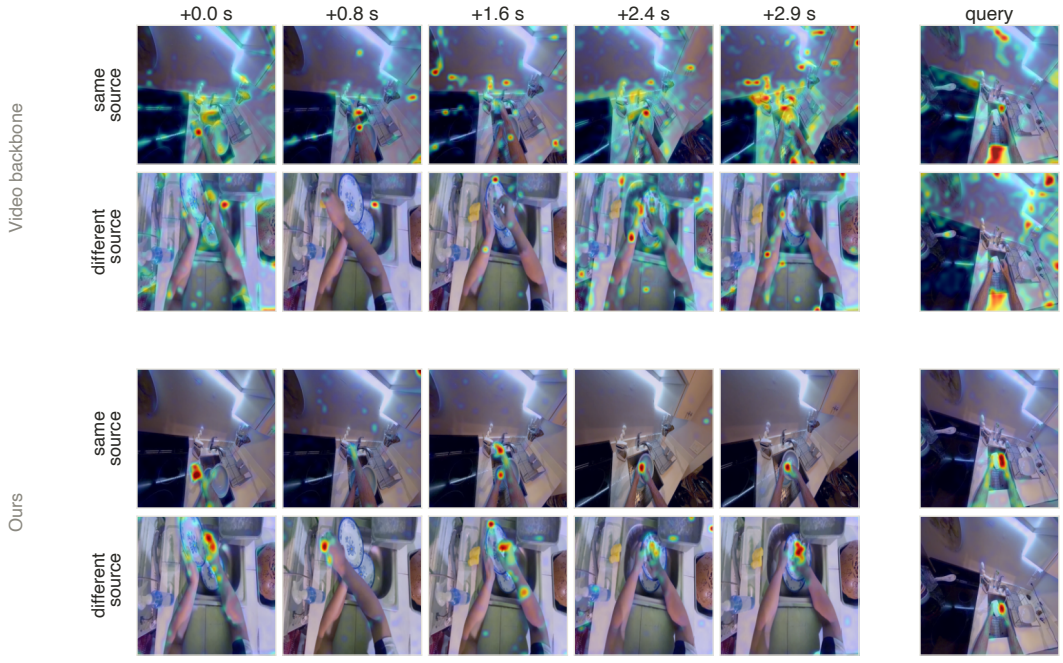
Table 1 evaluates prediction on RoboMIND; Table 6 adds two sets, Humanoid Everyday, a humanoid teleoperation dataset and thus robot data, and EgoDex, human demonstrations from a different egocentric rig, with the same end-effector action metrics and with the co-training variant added. Co-training is not a held-out comparison on Humanoid Everyday: it draws its robot queries from that dataset (§D.3), so it is tested on a source it was trained on, while WATCH, within-video and the original model have never seen it. Even so, WATCH matches or exceeds co-training on Humanoid Everyday on every metric but FVD, with half its end-effector error, a higher accuracy and sharper frames by FID and LPIPS: the cross-video objective reaches, on data it has never seen, the level that co-training reaches by training on it. On EgoDex, which none of the models has seen, WATCH leads on every action metric and on FVD and LPIPS, and co-training falls behind within-video throughout. On RoboMIND, robot data that none of the models has seen, WATCH again leads on every metric, with the lowest end-effector error and an accuracy twelve points above the next model, and the sharpest frames ahead of the original model, while within-video and co-training fall well behind on video. Fig. 13 shows one clip from each set: on Humanoid Everyday WATCH tracks the ground-truth gripper where the original model and within-video drift to a different arm and scene, and on EgoDex it is the closest to the ground truth, lifting the object as the hand does where the others leave it on the table.

E.3 Swapped Reference on More Queries

More queries and the backbone under a swapped reference. Fig. 4b shows the action-token attention of our model for one query under a reference from the same collection source and from a different one. Fig. 14 repeats the comparison on two more queries from different tasks and collection sources, and adds the original video model under the same two references. The layout follows Fig. 4b: the five columns on the left are the reference at successive points along its span, the column on the right is the query's current frame, and the two rows of a block differ only in the source of the reference. Under the backbone the mass is spread over edges, ceiling lights and background in both the reference and the query, and it changes little between the two rows. Under ours it sits on the hand and the manipulated object, in the reference and in the query alike, and it stays there when the reference comes from a different source in a different place.



(a) Bagging groceries



(b) Do dishes

Figure 14: **Swapped reference, two queries.** Action-token attention of the video backbone (upper block of each query) and ours (lower block) with a reference from the same collection source and from a different one. (a) Bagging groceries: an Aria query, the different-source reference a Scale demonstration. (b) Do dishes, the query of Fig. 4b: a Scale query, the different-source reference a MECKA demonstration.

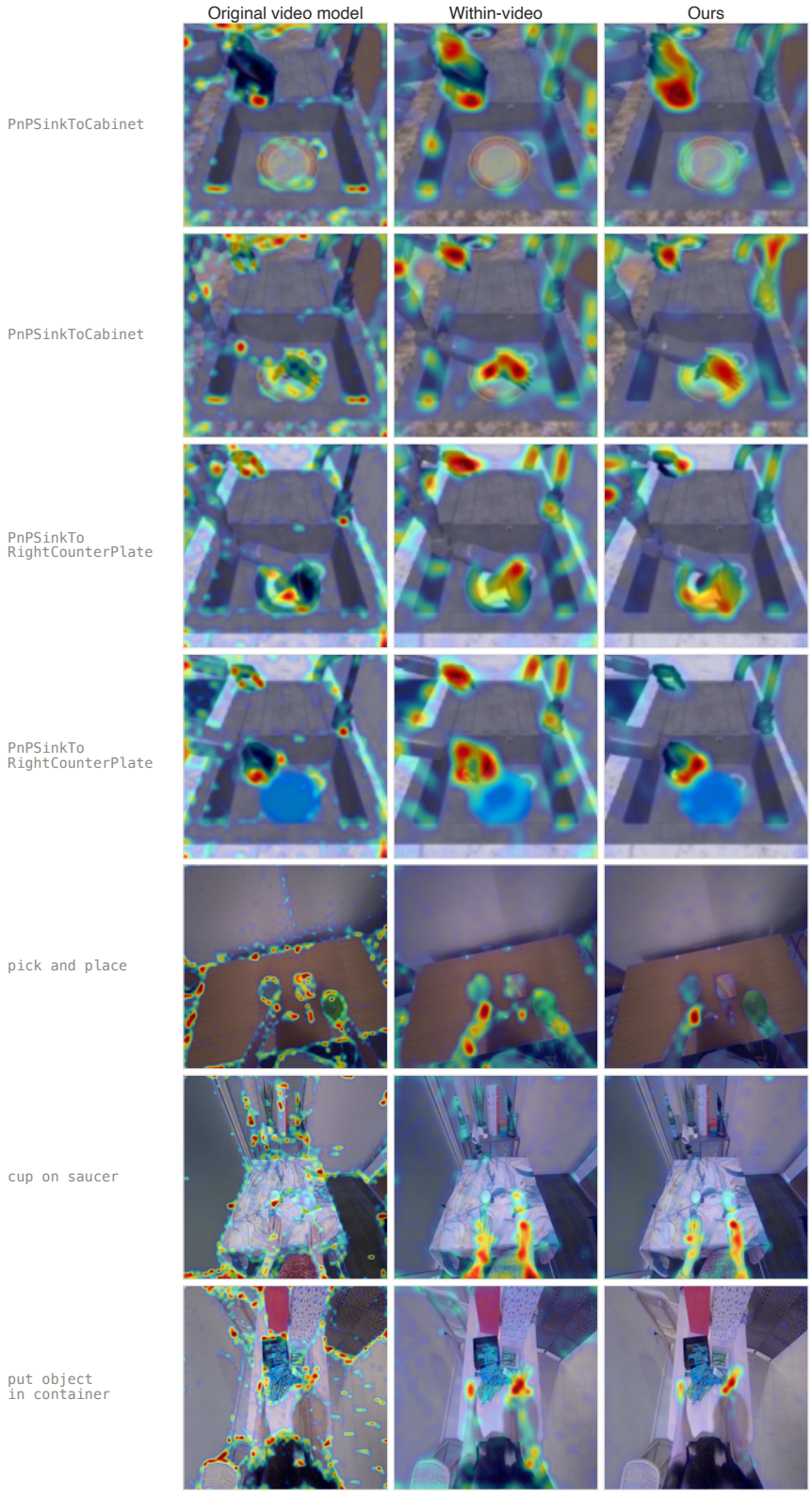


Figure 15: **Attention on robot and human frames, three models.** Attention to the current frame under the original video model (Cosmos-Predict2.5), the same model pre-trained within-video, and ours. Top four rows: side-right camera of GR1 MimicDroid rollouts of the task named on the left. Bottom three rows: three EgoVerse queries, on the query's current frame.

E.4 Attention on Robot and Human Frames, Three Models

Fig. 15 shows the current-frame attention of the original video model, the within-video baseline and ours on four robot frames (top four rows, side-right camera of GR1 MimicDroid rollouts) and three EgoVerse queries (bottom three rows). The original video model spreads over edges and background, within-video pre-training moves the mass onto the arm, and cross-video pre-training puts it on the hand and the object.

E.5 Analysis Protocol for the Representation Analyses

This subsection documents how the representation analyses of §3 (Fig. 5) and of §E.6 were run: the models, the two window sets, the features, and the retrieval and probing measures. It adds no new result.

Models. The *original video model* is the released Cosmos-Predict2.5-2B image-to-video checkpoint, with no EgoVerse training. The *within-video* baseline and *ours* are pre-trained on EgoVerse for 8K iterations with the same architecture and reference layout, within-video with blank reference slots and ours with the retrieved reference.

Window sets. The human set is the 359 windows of the analysis set (§C.4): MECKA 169, Scale 158, Aria 32. The robot set is 396 windows from RoboMIND (Wu et al., 2025) on three robots (Franka, UR5e, TienKung) and six tasks (bread in basket, open and close trash, open and close drawer, throw trash), every task on every robot, 22 episodes per cell and one window each (Fig. 11). The window takes the midpoint of the motion path as the present, the halves before and after as past state chunk and future, resampled to 92 frames as for the human windows; the camera is each robot's fixed top camera, and the wrist trajectory is the recorded end-effector pose (Franka) or forward kinematics of the recorded joints. No model has seen robot data.

Features. Every window is read through each model in its training layout: the current frame and the past state chunk as query, the reference video slots blank for all three models, the text prompt naming the task and the source or embodiment. Ours additionally receives the future actions of a matched reference from the same collection source, or the same robot, in its retrieved-action slots, as in training; the information in that reference can contribute to its retrieval and probing results alongside what pre-training changed in the features. We capture the first forward pass of generation (denoising step 0, guidance 1.5) and take the spatial mean of the action-slot hidden state at each transformer layer, one 2048-dimensional vector per window and layer. Fig. 5 reads layers 10–27, the later part of the 28-layer network, where the representation is closest to the action output, and averages over them rather than picking one layer: retrieval ranks are averaged over these layers per query, and probes are fit per layer and their scores averaged.

Retrieval. A pair is a query and its matched partner from a *different* collection source for the human set (120 pairs, from the training retrieval of §C.4), or from a *different* robot for the robot set (396 pairs, by dynamic-time-warping distance on size-normalized wrist trajectories within the same task). The partner is ranked by cosine distance, on window-wise z-normalized features, among all windows of the partner's source, or robot, and the rank is normalized to $[0, 1]$ with 0.5 chance; the curve is the cumulative distribution of the layer-averaged rank over the pairs. The area under the curve is 0.50, 0.66 and 0.78 on the human set and 0.65, 0.69 and 0.74 on the robot set for the original video model, within-video and ours.

Probes. Labels come from the wrist displacement over the window, in the waist frame for the human set and the robot base frame for the robot set. Each of the three direction axes is a three-way label, positive beyond $+3$ cm, still within ± 3 cm, negative beyond -3 cm (chance $1/3$); grasp is a four-way label, open, grasp, release or hold, from the hand closure at the first and last five frames of the window (pinch distance for the human set, gripper opening for the robot set; chance $1/4$). The classifier is fit per layer with standardization, PCA to 50 components and logistic regression under stratified five-fold cross-validation, scored by balanced accuracy; human windows are probed per hand and averaged. Bars in Fig. 17 are the mean over layers 10–27 and error bars the standard error

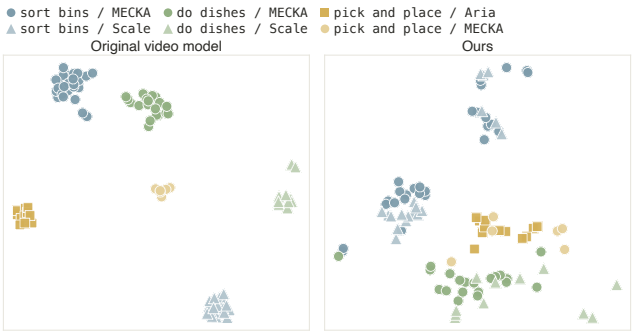
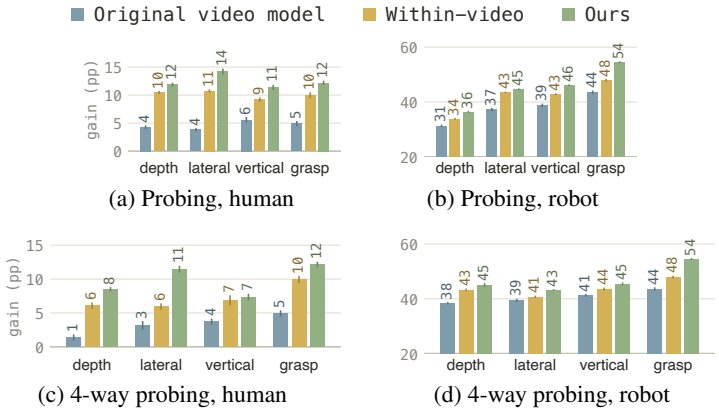


Figure 16: **Task layout across collection sources.** t-SNE of layer-0 action-slot features for 95 human windows of three tasks, each recorded by two of the three sources. Color is the task, in a full and a pale shade for its two sources, and shape is the source. Under the original video model each source forms its own island within a task; under ours the two sources of a task share an island.



(a) Probing, human (b) Probing, robot

(c) 4-way probing, human (d) 4-way probing, robot

Figure 17: **Linear probes of motion attributes.** Layers 10–27, as in Fig. 5; bars are linear-probe balanced accuracy above each attribute's chance level, in percentage points, error bars the standard error across layers. (a, b) Three-way direction probes and the four-way grasp probe on human (a) and robot (b) windows. (c, d) Four-way probes: each direction axis is split by sign and, within each sign, at the median magnitude, so the label carries the direction and a coarse size; grasp is the four-state label as before (chance $1/4$ throughout).

across layers, not a sampling interval. With one window per episode, the robot set's direction labels are tied to the task and robot, hence its high absolute accuracies.

t-SNE. Fig. 16 embeds layer-0 features into two dimensions with t-SNE, fitted separately per model on the same windows, so the panels share the points but not the axes. It draws the 95 human windows of three tasks each recorded by two sources, sort bins (MECKA 27, Scale 19), do dishes (MECKA 20, Scale 11) and pick and place (Aria 12, MECKA 6); single-source tasks are left out, since for them source and task cannot be told apart, and the container task because its 62 windows crowd the panel. The embedding uses perplexity 25 on 30 principal components, is initialized from the leading two components so it is deterministic, and runs 700 iterations with early exaggeration over the first 100.

E.6 Additional Probing Results

Fig. 17 probes the features of Fig. 5 for what they hold about the motion itself. The three-way probes (a, b) improve under ours on all four attributes on both sets, with the largest gains over the original video model in lateral direction for humans and grasp state for robots. The four-way probes (c, d) ask for the size of the motion as well as its direction; the ordering of the models is unchanged on every attribute, and the gains of ours are again largest on lateral direction and grasp (human) and on grasp (robot).

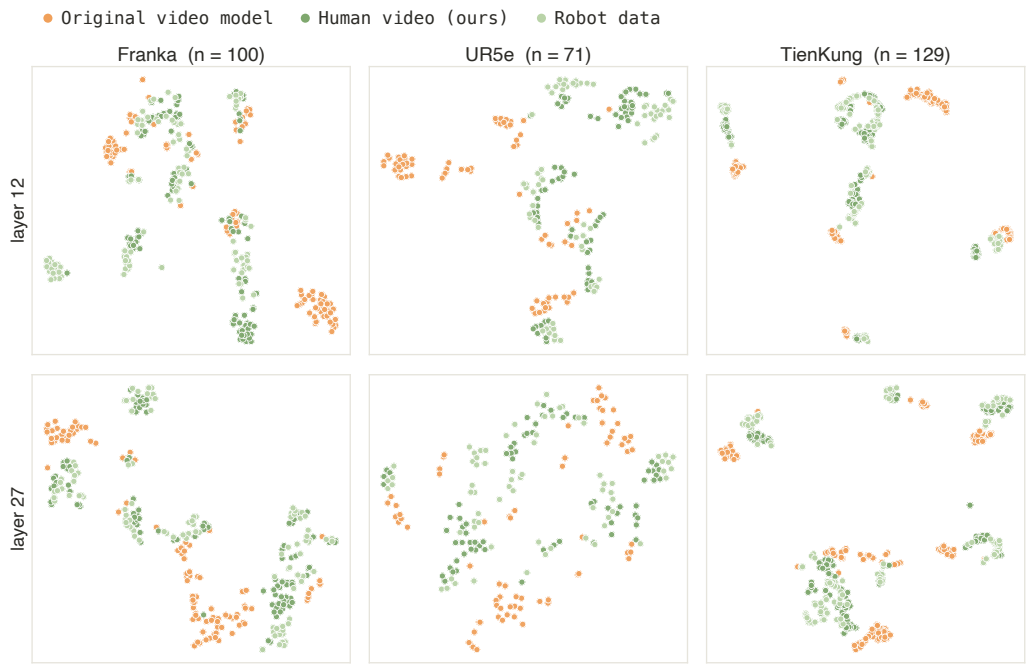


Figure 18: **Feature space of the human-trained and the robot-trained model on robot windows.** Action-slot features of 300 RoboMIND windows, per robot, at layers 12 and 27: each model is reduced to 10 principal components, the other two are rotated onto the human-video arm's components (two-fold cross-fit), and the three are embedded jointly with t-SNE, so every window appears three times.

The held-out-embodiment regression of Fig. 5c asks whether motion read from two embodiments transfers to a third. A kernel-ridge regressor is trained on the features of two embodiments and tested on the held-out one for nine quantities of the future motion (displacement, path length, speeds, timing and contact terms), each z-scored within its embodiment; the score is the Spearman correlation averaged over quantities and the three folds. Ours transfers best on both sets, at 0.18 on the human set and 0.24 on the robot set against 0.06 and 0.01 for the original video model, and the robot-set gain is the larger one even though no model has seen robot data. Two caveats: ours receives the reference's future actions as input (§E.5), which bears directly on a future-motion probe, and the Aria source has only 32 windows, so one of the three human folds is noisy.

F Human Data Against Robot Data as Pre-Training Data

F.1 Setup

This section asks how close pre-training on human video comes to pre-training on real robot data. We compare two arms with the same recipe and a comparable amount of paired video: `WATCH` on a 10% subset of the human pairs, about 1K hours, and the same objective on RoboMIND pairs alone, also about 1K hours, built with the same window and pairing definitions and counted as in §C.3. Only the robot arm sees RoboMIND during pre-training, and both are compared on RoboMIND windows, so agreement between them is the human arm matching a model trained on the test robots. The comparison set is a RoboMIND subset of 300 windows, 100 Franka, 71 UR5e and 129 TienKung, drawn so that the three robots share the same tasks.

F.2 Analysis

Feature space. Fig. 18 embeds the three models' features of the same robot windows jointly, after each model's feature space has been rotated onto the human arm's, so that a window's three copies would coincide if the models agreed. The human-video and the robot-data arm do agree: every cluster

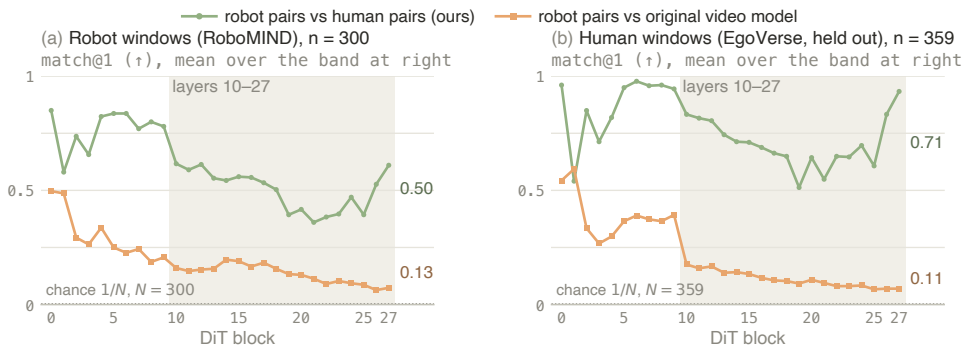


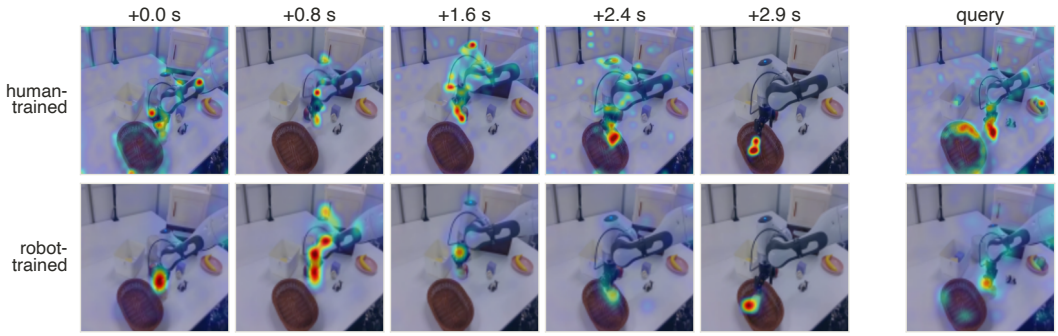
Figure 19: **Agreement with the robot-trained model, layer by layer.** Procrustes match@1 between the action-slot features of the robot-data arm and those of the human-video arm (green) or of the original video model (orange) at every DiT block, on (a) 300 RoboMIND windows and (b) 359 held-out EgoVerse windows. Each model's features are reduced to 50 principal components and scale-normalized; one model is rotated onto the other by an orthogonal Procrustes map fitted on half the windows and applied to the other half, the halves swapping roles so that every window is placed by a map it did not contribute to. The score is the fraction of windows whose nearest neighbor among all N reference points is the window itself, so chance is $1/N$: 0.0033 in (a) and 0.0028 in (b). Shaded: layers 10–27, the band of the summaries, with its mean at the right.

holds both greens, and a window's two copies sit at a mean distance of 0.03 to 0.06 after alignment, against 0.12 to 0.31 for the original video model, two to seven times farther on every robot and at both layers. The original model's copies form clusters of their own, so its disagreement is not noise around the same layout but a different organization of the same frames. The agreement holds across the three robots, on the three-camera Franka rig as on the single-camera UR5e and TienKung, and at both depths of the network, at layer 12 and at layer 27 next to the action head. A single objective applied to human pairs thus moves the representation of robot frames to nearly where the same objective applied to the robot data itself moves it, although the human arm has never seen a robot.

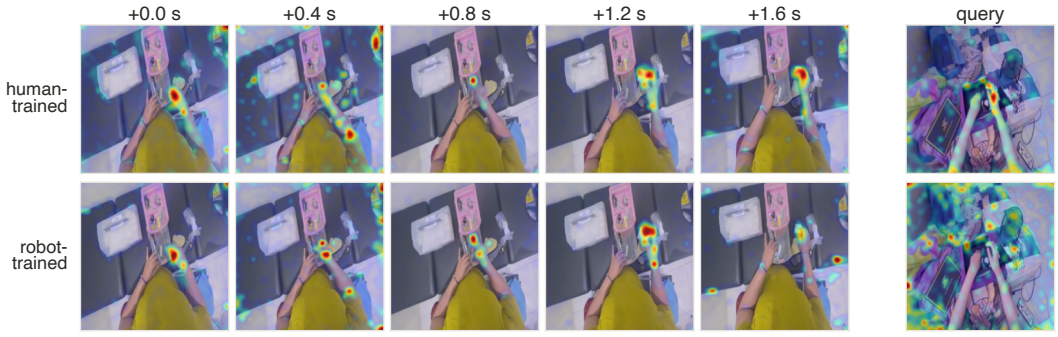
Agreement by layer. Fig. 19 applies a stricter test at every block: after a rotation fitted on other windows, a window's nearest neighbor under the other model should still be the window itself. The human-video arm keeps the robot-data arm's neighbors at every depth: on robot windows match@1 stays between 0.36 and 0.85 across the 28 blocks, with a mean of 0.50 over layers 10–27, and on human windows between 0.51 and 0.98, with a mean of 0.71, against a chance level of $1/N$, at most 0.0033 here. The original video model does not: against the robot arm it drops to 0.20 or less from layer 10 on, with band means of 0.13 and 0.11; the first ten blocks, where it still agrees at 0.2 to 0.6, carry the input-side features all three models share. From the tenth block on, then, a window keeps its neighbors across the human and the robot arm and loses them against the model both started from.

Attention. Fig. 20 asks the same question of where the two arms look, on one query from each domain. On the Franka query both arms follow the gripper and the bread through the reference, shifting their mass from the bread on the table to the gripper as it closes, and in the current frame both concentrate on the gripper and the basket, the human arm on a robot it has never seen. On the MECKA query both arms track the hand and the object it is sorting through the reference and land on the hands in the current frame, the robot arm now on a human scene it has never seen. Neither arm spreads its mass over the background the way the original video model does (Fig. 2b). In both directions, then, the arm trained on one embodiment attends to the other's frames the way the arm trained on that embodiment does.

Reading. Together, these analyses show a correspondence between the features learned from human pairs and from robot pairs, alongside similar attention patterns around hands or grippers and the manipulated object. These observations suggest that human-pair pre-training captures aspects of interaction that robot demonstrations also teach.



(a) Bread in basket, Franka query



(b) Sort bins, MECKA query

Figure 20: **Attention of the human-trained and the robot-trained model.** Action-token attention under a same-embodiment reference, five points of its span then the query frame. (a) A Franka query, bread in basket, from the left camera for both arms. (b) A MECKA query, sort bins, over the first half of the reference, where the sorting happens.

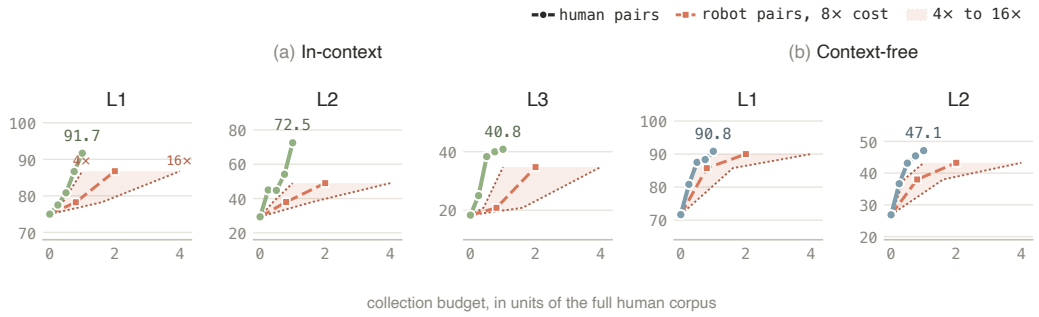


Figure 21: **Downstream transfer under a collection budget.** The human ladder of Figure 9 (green, blue) and the robot-pairs arm (red), under the same MimicDroid post-training. An arm's budget is its amount of paired video times the cost of collecting a unit of that video, in units of human video; the ratio between the two is not measured here, so the band sweeps $4\times$ to $16\times$ and the line draws $8\times$. Each robot line ends at the largest robot setting we ran. Both arms start at 0 from the same model without this pre-training stage. (a) In-context, L1 to L3; (b) context-free, L1 and L2. Each panel has its own y range.

F.3 Downstream Transfer from Human and Robot Pre-Training

Human demonstrations enable pre-training to scale without the robot hardware and teleoperation required for collecting robot data. We examine whether this advantage translates into more cost-efficient scaling than robot-pair pre-training, with the MimicDroid post-training protocol and target-robot data held fixed. Following the relative collection-cost analysis of RoboVQA (Sermanet et al., 2024), Figure 21 compares downstream success under assumed robot-to-human collection-cost ratios.

Table 7: **In-context MimicDroid, per task.** Success rate (%) of every model of Table 2 and Table 4 on the four tasks of each level; *Avg.* is the mean over the tasks of a level, and the last column the mean over L1–L3. Best in bold, second best underlined.

Method	L1					L2					L3					
	CloseLeftCabinetDoor	TurnOnFaucet	PnPSinkToCabinet	PnPSinkToRightCounterPlate	Avg.	CloseLeftCabinetDoor	CloseRightCabinetDoor	PnPSinkToCabinet	PnPSinkToRightCounterPlate	Avg.	CloseLeftCabinetDoor	TurnOnFaucet	PnPSinkToMicrowaveTop	PnPSinkToRightCounterPlate	Avg.	Avg.
ICL baselines																
Vid2Robot (2024)	**100.0**	**100.0**	63.3	**100.0**	90.8	53.3	0.0	63.3	**87.5**	51.0	0.0	0.0	0.0	3.3	0.8	47.6
VLA baselines																
$\pi_{0.5}$ (2025)	**100.0**	**100.0**	80.0	36.7	79.2	43.3	0.0	70.0	40.0	38.3	**33.3**	0.0	0.0	0.0	8.3	41.9
GR00T-N1.7 (2025b)	**100.0**	**100.0**	33.3	80.0	78.3	**100.0**	0.0	66.7	70.0	59.2	**33.3**	6.7	0.0	0.0	10.0	49.2
Cosmos-Predict2.5-2B																
Cosmos Policy (2026)	**100.0**	**100.0**	40.0	60.0	75.0	0.0	33.3	6.7	77.5	29.4	0.0	70.0	0.0	3.3	18.3	40.9
+ Within-video	**100.0**	**100.0**	63.3	80.0	85.8	**100.0**	0.0	40.0	37.5	44.4	**33.3**	56.7	0.0	3.3	23.3	51.2
+ Co-training	**100.0**	**100.0**	53.3	96.7	87.5	**100.0**	33.3	70.0	30.0	58.3	**33.3**	66.7	0.0	36.7	34.2	60.0
+ WATCH, same-source top-1	**100.0**	**100.0**	70.0	66.7	84.2	86.7	0.0	66.7	80.0	58.3	**33.3**	66.7	0.0	33.3	33.3	58.6
+ WATCH (Ours)	**100.0**	86.7	**100.0**	80.0	**91.7**	66.7	66.7	**86.7**	70.0	72.5	**33.3**	66.7	30.0	33.3	40.8	68.3
Cosmos3 Edge-2B																
Cosmos3 (2026)	**100.0**	**100.0**	46.7	30.0	69.2	**100.0**	**100.0**	60.0	57.5	79.4	**33.3**	0.0	6.7	**83.3**	30.8	59.8
+ WATCH (Ours)	86.7	**100.0**	90.0	90.0	**91.7**	**100.0**	**100.0**	73.3	72.5	**86.5**	**33.3**	**80.0**	**40.0**	66.7	**55.0**	**77.7**

We use an $8\times$ cost per hour of robot data as the central setting and show sensitivity to ratios from $4\times$ to $16\times$. Budgets cover pre-training data acquisition and are normalized by the estimated cost of the full human corpus.

Under these cost assumptions, human-pair pre-training scales more cost-efficiently than robot-pair pre-training. On in-context L2, increasing the human corpus from 25% to 100% raises success from 45.0% to 72.5%. The largest robot-pair setting reaches 49.0% at approximately twice the estimated collection budget under the $8\times$ assumption. A similar trend appears under context-free post-training, where human pairs reach comparable or higher success at lower estimated budgets. These results support human demonstrations as a cost-efficient source for scaling pre-training, allowing a given collection budget to support more data and stronger downstream control without increasing target-robot post-training data.

G Additional Experimental Results & Details

G.1 Baselines

All baselines we train are post-trained on the same robot demonstrations as WATCH (§4.1). MimicDroid is evaluated over 30 episodes per task (40 for the L2 plate task, §C.5), TableTop-Sim and RoboCasa over 50 episodes per task. The vision-language-action baselines $\pi_{0.5}$ (Physical Intelligence, 2025) and MolmoAct2 (Fang et al., 2026) cap the action dimension their heads accept, so on MimicDroid they use a 20-dimensional action vector rather than the 57-d representation of §C.1, which every other model uses there; on TableTop-Sim every model uses ALOHA's 20-d action target. GR00T-N1.7 (NVIDIA, 2025b) and Vid2Robot (Jain et al., 2024) follow their released recipes on the same data; HAND (Hong et al., 2025) and MimicDroid (Shah et al., 2025) are reported by prior work under its own evaluation data.

G.2 Per-Task Results

In-context MimicDroid. Table 7 breaks the in-context MimicDroid numbers of Tables 2 and 4 down by task. L1 is saturated on the door and faucet tasks for nearly every model, so the level is decided by the two pick-and-place tasks; on L2 and L3 cross-video pre-training improves multiple tasks at each level, and only the cross-video models reach double digits on L3 PnPSinkToMicrowaveTop.

Table 8: **Context-free MimicDroid, per task.** Success rate (%) of every model of Table 3 and Table 4 on the four L1 tasks and the four L2 tasks; *Avg.* is the mean over the tasks of a level, and the last column the mean over the two levels. Best in bold, second best underlined.

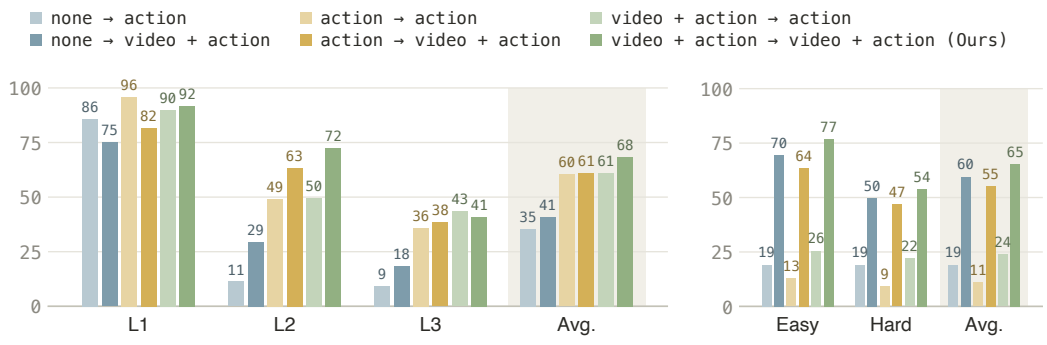
Method	L1					L2					
	CloseLeftCabinetDoor	TurnOnFaucet	PnPSinkToCabinet	PnPSinkToRightCounterPlate	Avg.	CloseLeftCabinetDoor	CloseRightCabinetDoor	PnPSinkToCabinet	PnPSinkToRightCounterPlate	Avg.	Avg.
VLA baselines											
$\pi_{0.5}$ (2025)	**100.0**	**100.0**	73.3	40.0	78.3	10.0	**0.0**	56.7	52.5	29.8	54.1
GR00T-N1.7 (2025b)	**100.0**	**100.0**	43.3	73.3	79.2	76.7	**0.0**	56.7	50.0	45.8	62.5
MolmoAct2 (2026)	**100.0**	**100.0**	26.7	80.0	76.7	70.0	**0.0**	3.3	40.0	28.3	52.5
Cosmos-Predict2.5-2B											
Cosmos Policy (2026)	**100.0**	43.3	53.3	90.0	71.7	0.0	**0.0**	30.0	77.5	26.9	49.3
+ Within-video pre-training	**100.0**	**100.0**	63.3	90.0	88.3	0.0	**0.0**	83.3	80.0	40.8	64.6
+ Co-training	**100.0**	93.3	73.3	**93.3**	90.0	0.0	**0.0**	**96.7**	72.5	42.3	66.1
+ WATCH, same-source top-1	**100.0**	**100.0**	**100.0**	60.0	90.0	0.0	**0.0**	66.7	80.0	36.7	63.3
+ WATCH (Ours)	**100.0**	**100.0**	76.7	86.7	**90.8**	0.0	**0.0**	93.3	**95.0**	**47.1**	**69.0**
Cosmos3 Edge-2B											
Cosmos3 (2026)	80.0	**100.0**	80.0	53.3	78.3	0.0	**0.0**	26.7	57.5	21.0	49.7
+ WATCH (Ours)	**100.0**	**100.0**	93.3	70.0	**90.8**	0.0	**0.0**	66.7	77.5	36.0	63.4

Table 9: **TableTop-Sim, per task.** Success rate (%) of every model of Table 3 and Table 4 on the five tasks of the easy (seen) and the hard (unseen) split under context-free post-training; *Avg.* is the mean over the tasks of a split, and the last column the mean over the two splits. Best in bold, second best underlined.

Method	Easy (seen)						Hard (unseen)						
	handover box	lift box	shoes table	dish drainer	box into pot	Avg.	handover box	lift box	shoes table	dish drainer	box into pot	Avg.	Avg.
VLA baselines													
$\pi_{0.5}$ (2025)	66.0	42.0	80.0	90.0	74.0	70.4	48.0	0.0	44.0	56.0	22.0	34.0	52.2
GR00T-N1.7 (2025b)	62.0	30.0	84.0	84.0	74.0	66.8	56.0	14.0	**82.0**	76.0	26.0	50.8	58.8
MolmoAct2 (2026)	70.0	32.0	70.0	92.0	68.0	66.4	20.0	**22.0**	42.0	86.0	28.0	39.6	53.0
Cosmos-Predict2.5-2B													
Cosmos Policy (2026)	82.0	34.0	72.0	78.0	82.0	69.6	76.0	14.0	60.0	74.0	24.0	49.6	59.6
+ Within-video	68.0	34.0	74.0	**100.0**	82.0	71.6	52.0	20.0	62.0	**100.0**	24.0	51.6	61.6
+ Co-training	74.0	34.0	76.0	92.0	70.0	69.2	76.0	14.0	66.0	86.0	22.0	52.8	61.0
+ WATCH, same-source top-1	62.0	44.0	76.0	94.0	72.0	69.6	62.0	10.0	62.0	92.0	30.0	51.2	60.4
+ WATCH (Ours)	82.0	40.0	76.0	96.0	**90.0**	76.8	54.0	**22.0**	**82.0**	86.0	26.0	54.0	65.4
Cosmos3 Edge-2B													
Cosmos3 (2026)	78.0	46.0	88.0	98.0	72.0	76.4	78.0	0.0	54.0	72.0	72.0	55.2	65.8
+ WATCH (Ours)	**96.0**	**58.0**	**92.0**	92.0	**90.0**	**85.6**	**92.0**	0.0	44.0	70.0	**86.0**	**58.4**	**72.0**

Context-free MimicDroid. Table 8 breaks the context-free MimicDroid numbers of Tables 3 and 4 down by task (the cabinet tasks close the doors). The two L2 cabinet doors score near zero for most models, so the L2 average is driven by the pick-and-place tasks, where cross-video pre-training gains most.

TableTop-Sim. Table 9 breaks the TableTop-Sim numbers of Tables 3 and 4 down by task. Lift box is the hard case on both splits; box into pot on the unseen split is where the backbones part, Cosmos3 keeping its seen performance while the Cosmos Policy variants fall sharply.



(a) MimicDroid, in-context

(b) TableTop-Sim, context-free

Figure 22: **Supervision ablation, all configurations.** Success rate (%). Hue: pre-training target (none, action, video + action); pale: action-only post-training, full: video + action. (a) MimicDroid, in-context, the reference given as actions only under action-only post-training; (b) TableTop-Sim, context-free.

G.3 Data Scaling Ladder

The sweep of §4.3 draws 25, 50 and 75% of the query episodes. The split is made within each collection source and rounded up, so the arms hold 3,330, 6,659 and 9,988 of the 13,315 episodes that carry query windows (§C.3), a few episodes above the exact quarters, and 25.6, 49.7 and 74.4% of the query windows. Training length scales with the data, 2K, 4K, 6K and 8K iterations at a global batch of 64, so every arm sees the same 0.49 epochs; data and compute therefore grow together, and the ladder is a fixed-epoch scaling of the corpus.

G.4 Supervision Ablation: Post-Training Target as Well

Fig. 8 fixes post-training at video + action and varies what pre-training supervises; Fig. 22 varies the post-training target as well, so every bar reads *pre-training target* → *post-training target*, and under action-only post-training the in-context reference is given as actions only. Video + action pre-training is the best pre-training target under either post-training target on both benchmarks. Action-only post-training is competitive on L1 and close on L3 but trails on L2 under every pre-training target, and collapses on TableTop-Sim, which has no reference. Action-only pre-training helps on MimicDroid, where the reference actions carry the task, but falls below no pre-training on TableTop-Sim under both targets (11 and 55 against 19 and 60).

H Qualitative Results: Robot Rollouts

H.1 MimicDroid, In-Context

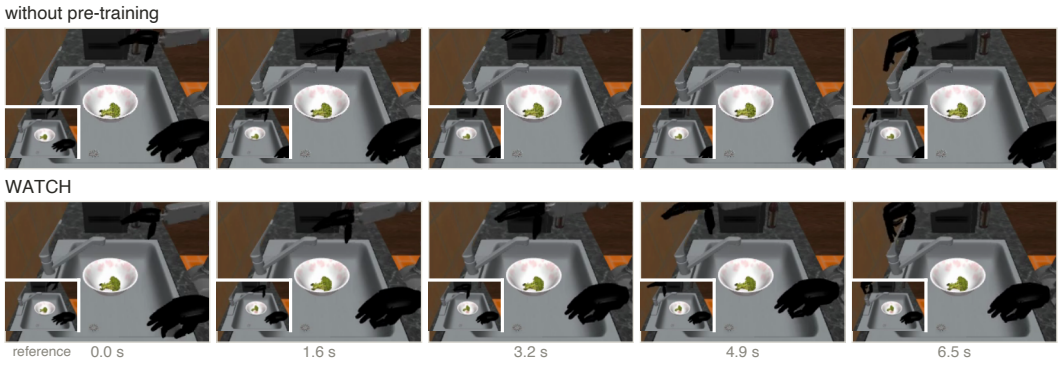
In Fig. 23 the inset advances with the robot under `WATCH`, as each chunk retrieves the reference window matching its progress, and the hand tracks the demonstrated path; without pre-training it hovers or stalls after the grasp.

H.2 MimicDroid, In-Context with 2× Speed Demonstrations

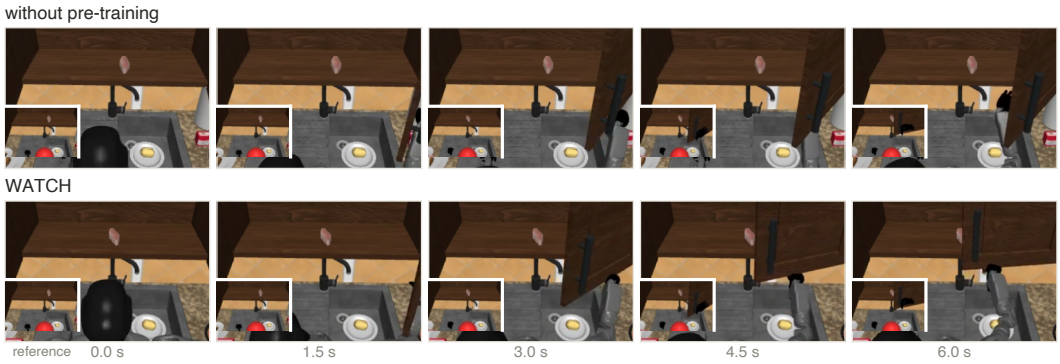
Figs. 24 and 25 show demonstrations the arm cannot copy: at 2× speed the hand closes the door, or reaches the cabinet, within one chunk. Without pre-training the generated frames stay at the reach and the rollout fails; under `WATCH` they follow the demonstration's progress, not its timing.

H.3 TableTop-Sim, Context-Free

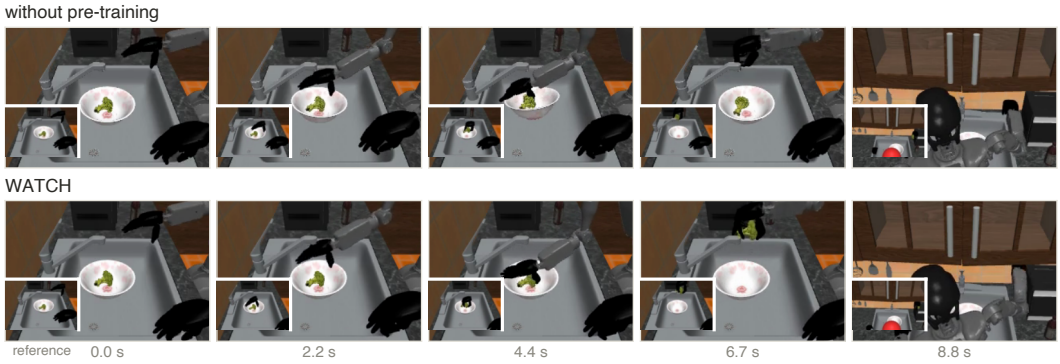
In Fig. 26, without pre-training the policy reaches or lifts the object but fails at the completing step; `WATCH` finishes within the same window on both splits.



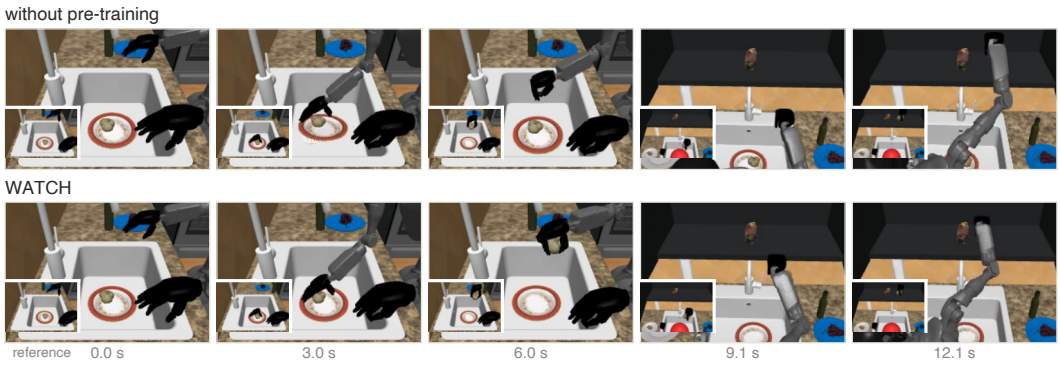
(a) `TurnOnFaucet` (L3)



(b) `CloseRightCabinetDoor` (L2)

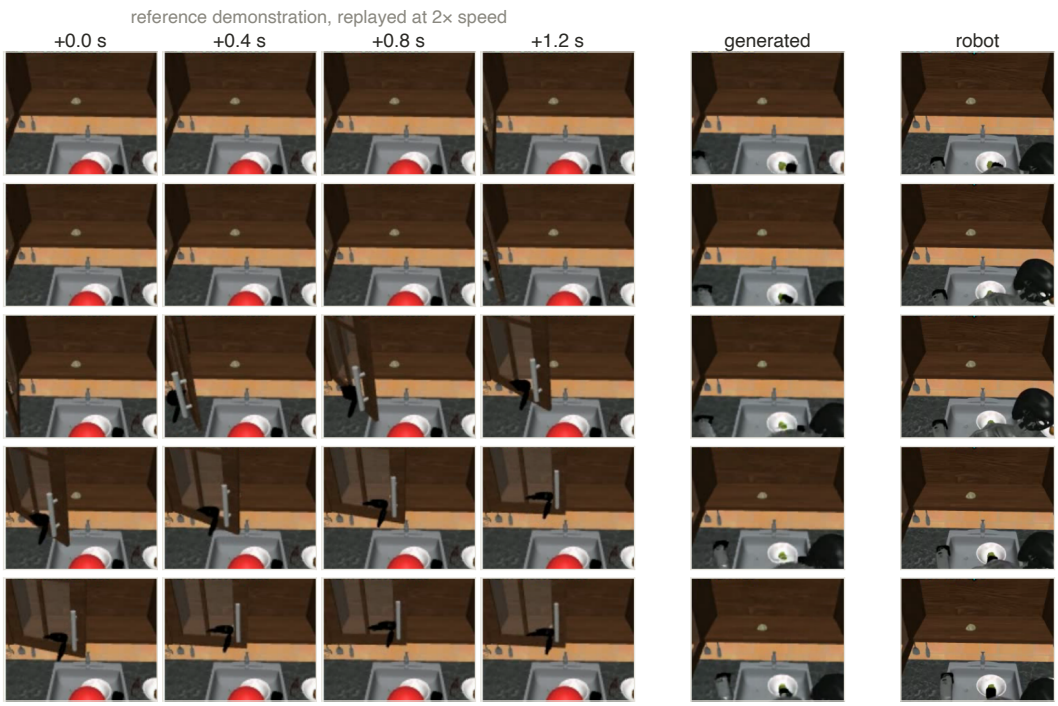


(c) `PnPSinkToMicrowaveTop` (L3)

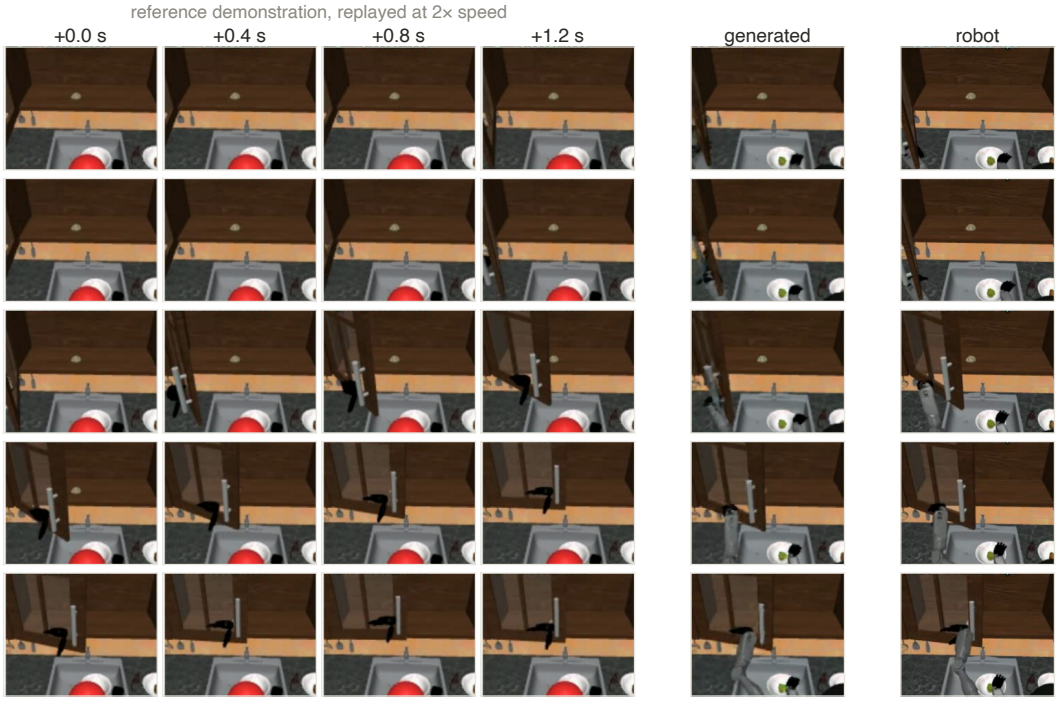


(d) `PnPSinkToCabinet` (L2)

Figure 23: **In-context rollouts on MimicDroid.** Per task, Cosmos Policy post-trained in context without pre-training (top) and `WATCH` (bottom) at the same five time steps over the `WATCH` episode; the failing episode runs on to the time limit. The inset is the retrieved reference frame the policy is conditioned on. Each frame uses whichever of the front, left and right cameras shows the motion best; in (b) the two rows start from different initial states.

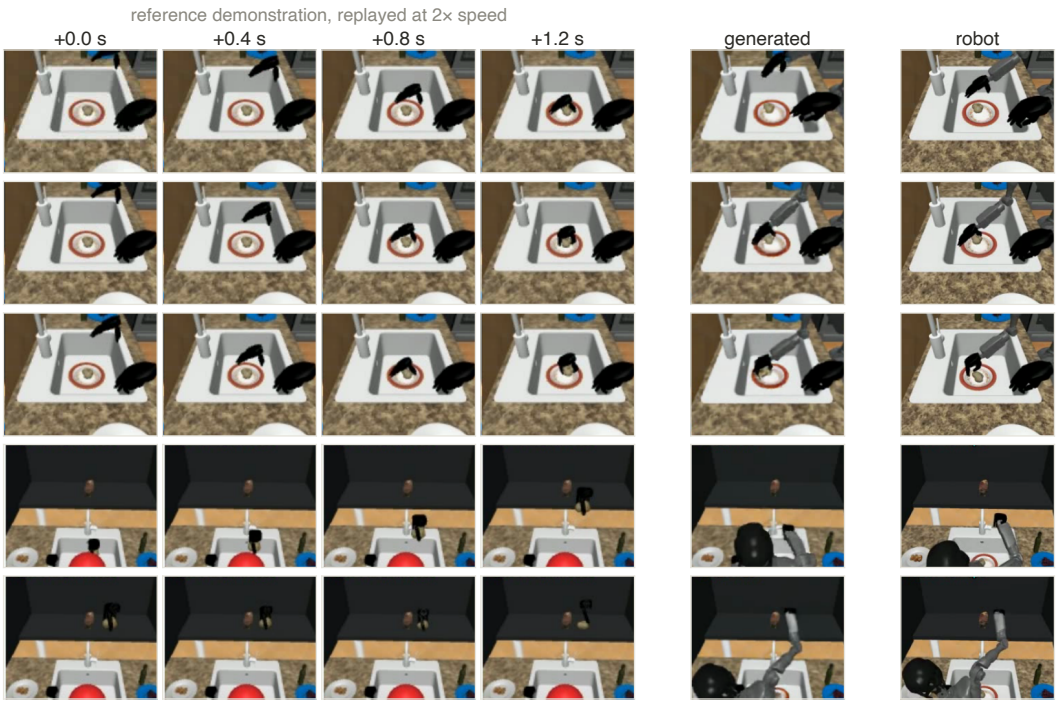


(a) Cosmos Policy without pre-training

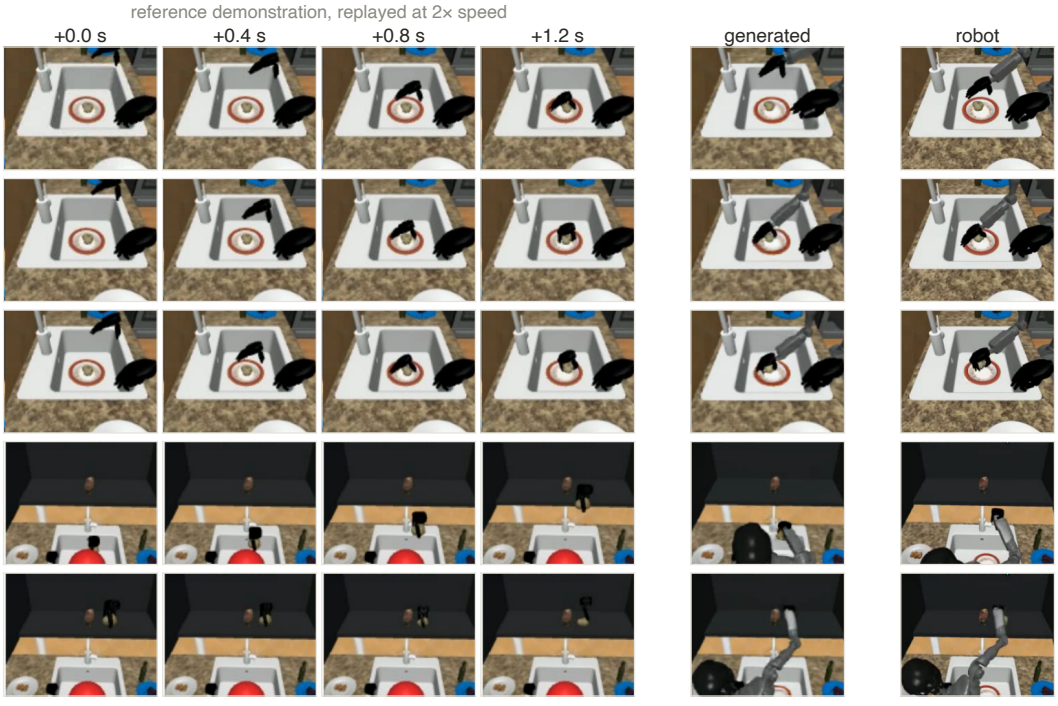


(b) WATCH

Figure 24: **Reading a 2×-speed demonstration, chunk by chunk.** The same CloseLeftCabinetDoor episode (L3, camera behind the robot) under (a) Cosmos Policy without pre-training, which fails, and (b) WATCH, which succeeds. Rows are 1.6-second action chunks from 1.6 to 9.6 s: four reference frames at 0.4-s steps, replayed at 2×; *generated*, the frame predicted for the chunk; *robot*, the frame reached at its end. The demonstrated hand closes the door within a chunk; without pre-training the generated frames never show it closing and the arm stalls, WATCH spreads the motion over its own chunks and finishes.

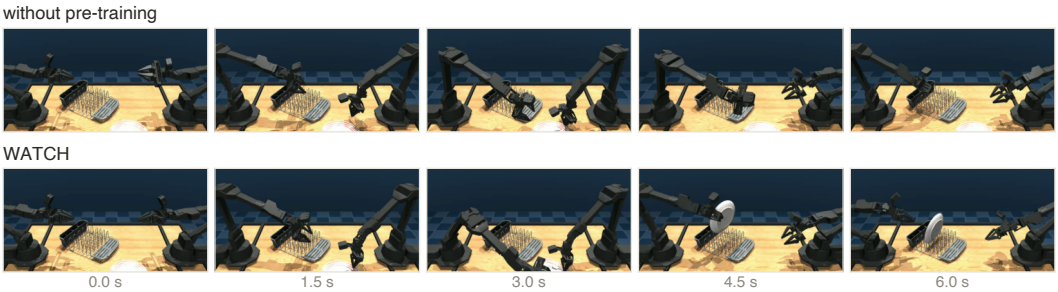


(a) Cosmos Policy without pre-training



(b) WATCH

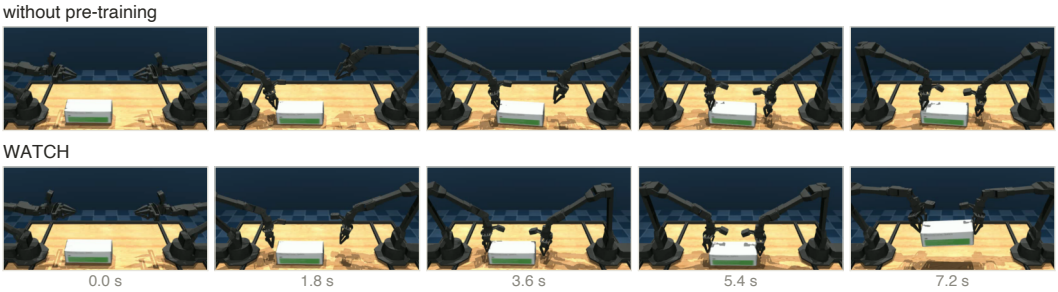
Figure 25: **A two-stage task under a 2×-speed demonstration.** The same PnPSinkToCabinet episode (L2) under (a) Cosmos Policy without pre-training, which fails, and (b) WATCH, which succeeds; rows as in Fig. 24: chunks 1–3 (0 to 4.8 s) from the left camera, where the grasp is visible, then chunks 5 and 7 from the camera behind the robot, which shows the transport. Both grasp the object; WATCH places it in the cabinet, whereas without pre-training the arm rises toward the cabinet and stays there until the episode times out.



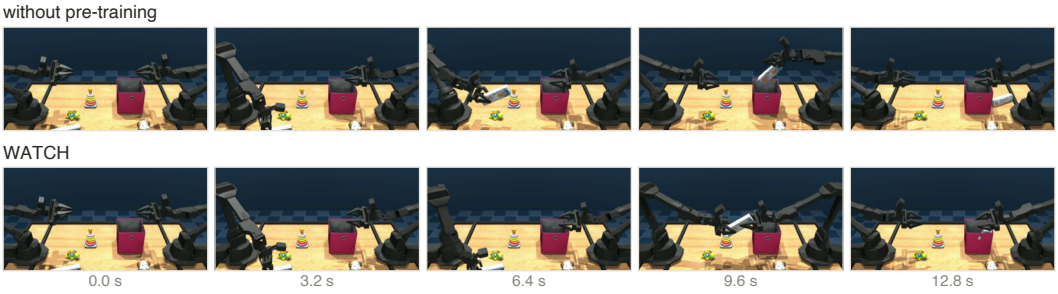
(a) Dish drainer (Easy)



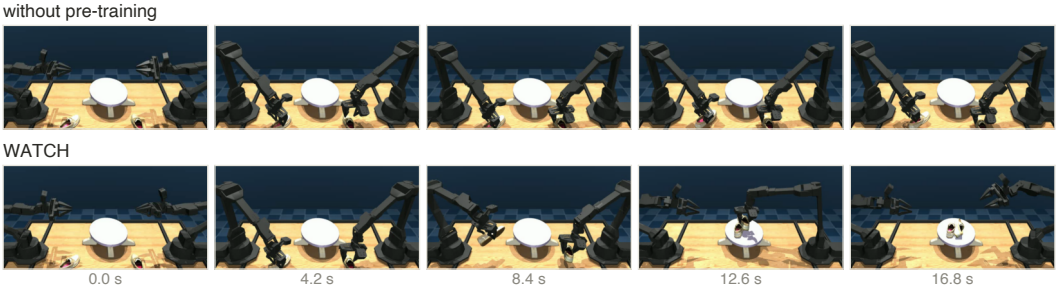
(b) Box into pot (Hard)



(c) Lift box (Easy)



(d) Handover box (Hard)



(e) Shoes table (Hard)

Figure 26: **Context-free rollouts on TableTop-Sim.** Per task, the same episode under Cosmos Policy without pre-training (top) and WATCH (bottom) at the same time steps over the WATCH episode; the failing episode runs on to the time limit. Without pre-training, the plate is lifted and dropped (a), the small box reached but not lifted (b), the arms hover around the box (c), the box is dropped before the pass (d), and the shoes are carried to the table but not released (e).