# WHAT DID WE LEARN FROM 20K HOURS OF OPEN-SOURCE ROBOT DATA?

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Robot learning is often constrained by the perceived scarcity of data, yet the community has already accumulated tens of thousands of hours of publicly available demonstrations. The key challenge is thus not collecting more data, but effectively reusing heterogeneous datasets collected across different robot platforms. We propose the camera-centric action representation as a unified interface for robot data. This representation captures the relation between vision and action while abstracting away embodiment-specific kinematic differences. Building on this idea, we present CamX, a unified corpus of 20k hours of open-source robot demonstrations from 37 sources, converted to a common camera-centric representation. To leverage CamX, we train CamUVA, a unified video-action model that uses the camera-centric action representation and identifies every camera view and action stream by its camera pose. This design allows the model to extract value from cross-embodiment datasets and generalize to unseen robot embodiments out of the box. Together, CamX and CamUVA provide a foundation for systematically studying both what matters—the design choices that enable effective scaling—and what is possible—the capabilities that emerge from scaling. Ultimately, through our empirical study, we hope to show a path toward scaling robot learning by effectively reusing the growing body of heterogeneous open-source robot data. Data samples and policy rollouts are available at camx-anonymous.github.io.

## 1 INTRODUCTION

A common narrative in robot learning is that the field suffers from a data shortage. Yet the robotics community has accumulated a surprisingly large amount of publicly available robot data. Open-source datasets now contain tens of thousands of demonstration hours across robots, tasks, and environments.[1] However, most existing policies, even those trained across multiple embodiments, do not fully exploit this vast body of data from diverse robot embodiments. The challenge therefore lies less in collecting more robot data, but more in effectively leveraging the data that already exists.

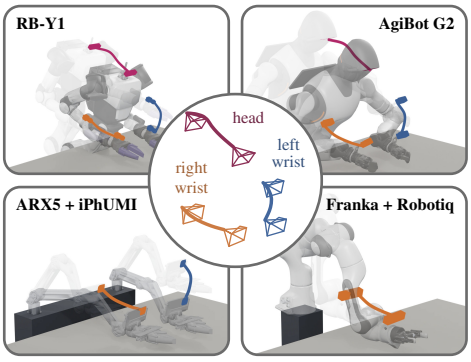



Figure 1: **Camera-centric action representation** unifies heterogeneous robot data. A camera rigidly mounted to a movable robot link (e.g., the end-effector or head) directly implies the corresponding action and task-relevant observation, without being tied to a specific embodiment.

The main obstacle to making better use of existing data is heterogeneity: robot datasets are collected with different embodiments, action spaces, camera configurations, coordinate systems, and sensor modalities. Efforts such as Open X-Embodiment (Open X-Embodiment Collaboration, 2024) have shown the benefits of aggregating data across many robot platforms, but policies trained on these corpora usually still require additional fine-tuning to adapt to a target embodiment and its action space. This suggests that simply accumulating datasets is insufficient: the representation through which data is unified matters as much as the scale of the data.

[1] UMI-Data Community: `https://umi-data.github.io/`



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.

How can we unify different robot datasets? Our key intuition is that **camera-centric action representation** offers a natural, embodiment-agnostic interface between perception and action. As shown in Fig. 1, a camera rigidly mounted to a robot link (e.g., the wrist, torso, or head) moves with that link up to a fixed rigid transform. Its motion thus captures the *critical task-relevant information*—how the robot should move its hands—while abstracting away embodiment-specific details such as joint configurations and kinematics. This makes camera-centric actions consistent across embodiments while preserving the geometric relation between visual observations and physical motion.

To test this idea at scale, we curate **CamX**, a unified corpus aggregating about 20k hours (1.7 million episodes) of publicly available robot data from 37 sources (Fig. 2). Building CamX requires converting datasets collected across diverse robot embodiments and camera configurations into a common camera-centric format, with a data correction and filtering pipeline to address inconsistencies in calibration, coordinate conventions (Appendix C.1), temporal alignment, and action representations.

Building on CamX, we develop **CamUVA** (Camera-Centric Unified Video-Action Model), a video-action model that jointly predicts visual observations and camera-centric actions. CamUVA is designed to handle the wide variation in camera setup and data format across robots, including the number and placement of cameras, image resolution, frame rate, and sequence length. To this end, we identify different views through their camera poses rather than fixed camera identities, which allows the model to generalize to arbitrary camera setups. Observations and actions are packed as variable-length sequences rather than padded to a fixed shape to accommodate heterogeneous observation spaces. Moreover, CamUVA trains jointly on video and action objectives, supporting video dynamics modeling while achieving better policy performance than action-only counterparts.

Together, CamX and CamUVA allow us to systematically study **what matters (i.e., design decisions) and what is possible (i.e., model capabilities) when training a robot model at scale**. This helps us identify a set of key design choices that enhance cross-embodiment performance, including relative camera trajectories as the action representation, camera pose for multi-view encoding, and joint video-action training. More importantly, CamUVA demonstrates capabilities that would be difficult to study from individual datasets alone: without embodiment-specific fine-tuning, it matches the performance of single-embodiment policies across the training embodiments; it also works out of the box on unseen robot setups with different camera views and gripper shapes, demonstrating **zero-shot cross-embodiment generalizability**.

In summary, this paper makes three contributions:

- **CamX:** A verified corpus of 20k hours of open-source robot demonstrations from 37 sources in a unified camera-centric representation.
- **CamUVA:** A flexible unified video-action model that supports arbitrary camera configurations, in addition to video dynamics modeling and policy learning.
- **Empirical insights:** A systematic study of design choices for learning from heterogeneous robot data, and an analysis of the factors that affect cross-embodiment transferability.

## 2 Related Work

**Cross-Embodiment Datasets and Policies.** Open-source robot data now spans single-arm (Khazatsky et al., 2024), bimanual and humanoid (Zhao et al., 2023a; Bu et al., 2025a), handheld wrist-camera (Chi et al., 2024; Shafiullah et al., 2023; Liu et al., 2026b; Lin et al., 2025) platforms, as well as aggregated collections (Open X-Embodiment Collaboration, 2024). Generalist policies bridge the embodiment gap with per-embodiment action heads or encoders (Ghosh et al., 2024; Doshi et al., 2025; Wang et al., 2024a; NVIDIA, 2025b), normalized and discretized end-effector actions (Zitkovich et al., 2023; Kim et al., 2025), padded action vectors (Black et al., 2025b), or embodiment prompts (Zheng et al., 2026). These designs can require fine-tuning or adaptation of the input and output mappings for a new robot. A second line of work seeks a shared action interface through latent actions learned from video (Ye et al., 2025; Bu et al., 2025b), universal action codes learned from robot action labels (Zheng et al., 2025), language (Zha et al., 2026), or end-effector actions expressed in a reference camera frame (Xie et al., 2025; Chen et al., 2026a; Yuan et al., 2026a; Zhang et al., 2026b; Li et al., 2026c). The latter retain the end-effector as the controlled body and map predictions to robot coordinates using a calibrated or estimated transform; CamVLA predicts this transform without supplied extrinsics (Li et al., 2026c). CamX instead treats the camera rigidly

attached to a moving link as the actuated body, so handheld, single-arm, bimanual and humanoid demonstrations share one SE(3) trajectory format with a unified camera-frame convention (Sec. 3.1).

**Video-Action Models.** Video prediction supports policy pretraining (Wu et al., 2024; Cheang et al., 2024; Hu et al., 2025), visual planning with inverse-dynamics action decoding (Du et al., 2023), and synthetic demonstration generation (Jang et al., 2025); V-JEPA 2 instead plans through action-conditioned latent predictions (Assran et al., 2025). Joint video-action models generate future frames and actions (Li et al., 2025c; Zhu et al., 2025a; Cen et al., 2025; Kim et al., 2026; Li et al., 2026b; Ye et al., 2026; Zhou et al., 2026), with recent methods adapting pretrained video generators (NVIDIA, 2025a; Team Wan et al., 2025). Cosmos Policy assigns views temporal slots and encodes actions as latent frames (Kim et al., 2026); DreamZero tiles views spatially and uses action encoders and decoders (Ye et al., 2026); UVA decodes video and actions with separate diffusion heads (Li et al., 2025c). Their view layouts and action interfaces differ, and transfer to a new robot can require fine-tuning (Ye et al., 2026). Separately, camera geometry has been injected into visual transformers as per-pixel ray conditioning (He et al., 2025; Jiang et al., 2026) or as relative geometric encodings inside attention (Miyato et al., 2024; Kong et al., 2024; Li et al., 2025b; Zhang et al., 2026a); Gamma-World (Liu et al., 2026a) extends 3D RoPE (Su et al., 2024; Wang et al., 2024b) with an agent axis. CamUVA identifies every view and action stream by its camera pose, so a single model consumes any number of views and end-effector groups.

# 3 Method

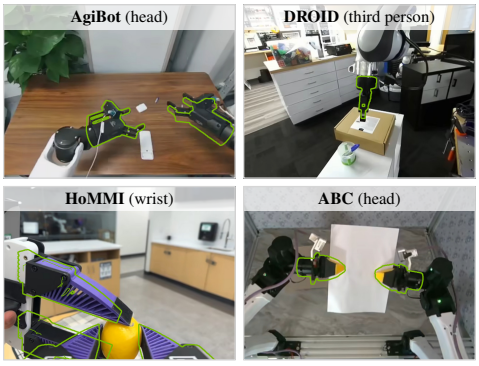
## 3.1 CamX Dataset Curation

**Unified Format for Diverse Embodiments.** For end-effector trajectories, we resolve differences in frame conventions by applying the fixed transform from each dataset's end-effector frame to its wrist camera (Appendix C.1). For joint trajectories, we compute forward kinematics using the robot's URDF and compose the resulting flange pose with the fixed flange-to-camera transform. For static third-person cameras, we use the released extrinsics. For mobile manipulators, we incorporate base motion into the camera trajectories: a camera fixed to the base captures its motion, while a wrist camera captures both base and arm motion.

We organize the data into actuator groups (e.g. left wrist, right wrist, head, etc.), each with a main camera trajectory, a gripper-width stream, and one or more RGB views, including stereo pairs. Views rigidly attached to a group are stored with fixed transforms relative to its main camera. We also record the extrinsics of the fixed third-person cameras for view identification. These conversions resolve dataset-specific differences once, so the policy receives the same camera-centric inputs and predicts the same camera-centric outputs across single-arm robots, bimanual systems, and mobile manipulators. At deployment, we map predicted relative camera trajectories to flange targets via the calibrated camera-to-flange transform.

**Data Validation.** We validate camera-to-gripper alignment by rendering the gripper mesh using the camera intrinsics and camera-to-gripper transform, then visually checking its alignment with the

**(a) Projection-Based Camera-Pose Validation**




**(b) CamX Composition**

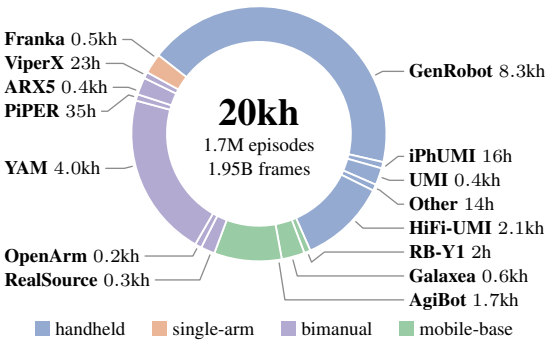



Figure 2: **CamX dataset.** We validate the camera poses by rendering the gripper mesh (green outline) in each camera view. This validates the camera frame coordinate and the camera-gripper transform. Small gaps are due to per-robot manufacturing variance and per-episode tracking error, harmless to relative camera trajectories.

observed gripper (Fig. 2a). Inspecting these overlays for every source and rig, we found calibration errors not flagged in the source metadata (Appendix C.3).

### 3.2 CamUVA Model Design

**Model Architecture.** CamUVA is a Mixture-of-Transformers (Liang et al., 2025) model with three branches per layer that share attention over a single packed token sequence. A 7B understanding branch processes the language instruction, and a 7B video branch processes the video latents; both are initialized from Cosmos3-Nano (NVIDIA, 2026). A 1B action branch, initialized from scratch, processes the camera-trajectory and gripper-width tokens. Following Fast-WAM (Yuan et al., 2026b), clean tokens attend only to language and clean tokens. At deployment, their keys and values are therefore computed only once, and only noisy tokens pass through every denoising step.

**Pose Encoding for View Identification.** Prior multi-view models identify views through spatial (Ye et al., 2026) or temporal (Kim et al., 2026) concatenation (Fig. 3(a, b)), tying view identity to the chosen layout. Simplex Rotary Agent Encoding (Liu et al., 2026a) makes identities equidistant in rotary angle space, without encoding camera geometry. We instead apply Camera Pose Encoding (CaPE) (Kong et al., 2024; Yuan et al., 2026a) to every input token (Fig. 3(d)). For a token carrying camera pose $P \in SE(3)$, we transform its query as $q \leftarrow P^{-\top} q$ and its key as $k \leftarrow Pk$ on consecutive groups of four channels. The attention logit between tokens from cameras $i$ and $j$ then becomes $q_i^\top P_i^{-1} P_j \, k_j = q_i^\top T_{ij} \, k_j$ and depends only on their relative pose $T_{ij}$. A video token carries its camera's pose at the conditioning frame, an action token its group's wrist-camera pose, and a language token the mean wrist-camera pose. All poses are expressed relative to the mean starting wrist-camera pose, removing each source's world origin; at inference they come from the camera calibration.

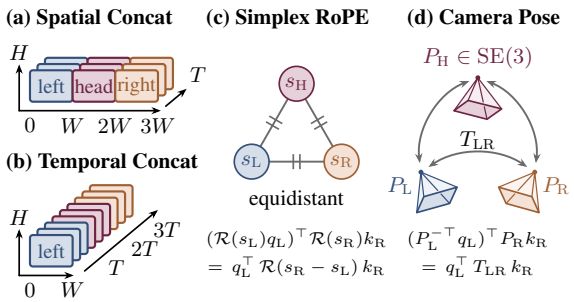



Figure 3: **View identification**. **(a, b)** By slot. **(c)** By an equidistant phase, no geometry. **(d)** By encoding absolute camera poses to all tokens, the relative transform $T_{LR}$ is calculated during attention.

**Embodiment Agnostic Encoding and Flexible Sequence Packing.** As shown in Fig. 4(a), we share the same Wan2.2 video VAE (Team Wan et al., 2025), camera-trajectory MLP, and gripper-width MLP across all embodiments, eliminating the need for embodiment-specific weights (Ghosh et al., 2024; Wang et al., 2024a; Doshi et al., 2025). We then apply interleaved 3D MRoPE (Wang et al., 2024b; Bai et al., 2025) to every stream, aligning the temporal indices of the video and action sequences by timestamp; the $h$ and $w$ indices of action tokens are set to zero. Two views thus receive identical MRoPE indices and are distinguished by CaPE alone, which places no architectural limit on the number of camera views, trajectories, and gripper widths. Finally, we flatten all input fields

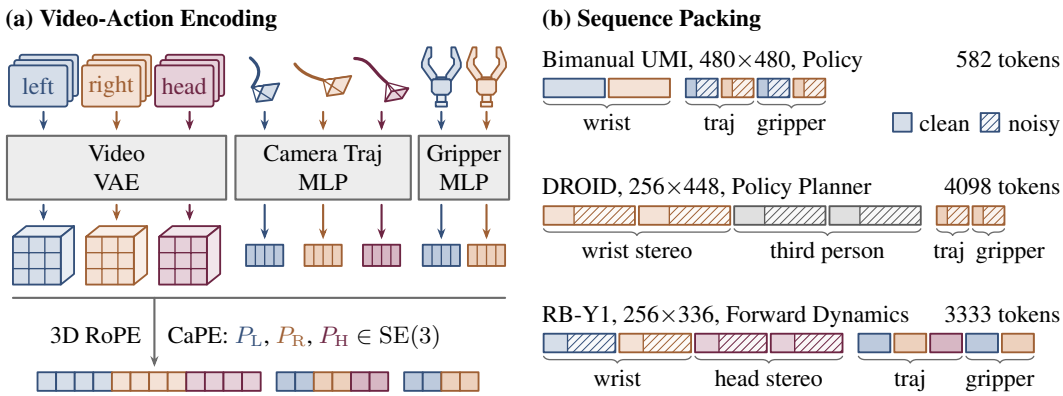



Figure 4: **Heterogeneous data encoding and packing.** **(a)** One encoder per modality is shared by all views; each token gets 3D RoPE and the CaPE of its own corresponding camera, so a view / trajectory is identified by its pose, not its slot. **(b)** One model serves every embodiment and UVA mode: only the streams in the sequence and which of their tokens are given change. The model denoises all noisy tokens at the same time.

into a single 1D token sequence and use FlexAttention (Dong et al., 2025) to implement per-sample attention masks. During training, each token attends only to tokens within its own sample, in the current or earlier chunks (Fig. 4(b)).

**Video Action Training Objectives.** We train with the five objectives of UVA (Li et al., 2025c): **Policy (P)**, **Policy Planner (PP)**, **Video Generation (VG)**, **Forward Dynamics (FD)**, and **Inverse Dynamics (ID)** (Fig. 7). We use these tags and colors to refer to the modes throughout the remainder of the paper. For all UVA modes, each chunk consists of 33 frames of observations $O_{0:t}$ and actions $A_{0:t}$, with $t = 32$. We add noise at a randomly sampled level to the modality being denoised in the generated frames, while keeping the conditioning frames clean (examples in Fig. 4(b)). We train both modalities with flow matching and compute the loss only over generated tokens. We use a flow-matching adaptation of UniPC (Zhao et al., 2023b) for sampling.

# 4 Experiments

## 4.1 What Matters? – Design Decision Study

We ablate our design choices using a shared training recipe (Appendix B.4). Unless otherwise stated, we initialize the understanding and video branches from Cosmos3-Nano-Policy-DROID (NVIDIA, 2026), train the 1B action branch from scratch, and train the model on DROID (Khazatsky et al., 2024) for about 4,000 GB300 GPU-hours, saving a checkpoint every 500 GPU-hours. We evaluate every checkpoint on all 120 tasks of the RoboLab benchmark (Yang et al., 2026b), with 10 episodes per task, and report the mean and standard deviation across the three best checkpoints of each run. Appendix D.1 provides the RoboLab rollout configuration.

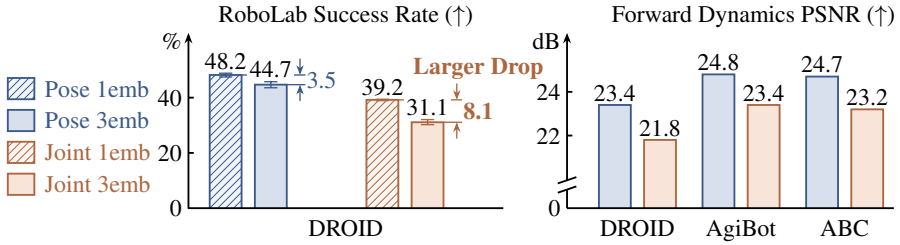



Figure 5: **Camera pose is a better aligned cross-embodiment action representation.** The same model is trained on DROID alone (1emb) or on DROID, AgiBot and ABC together (3emb). Both action spaces lose success rate under cross-embodiment training, but the joint-space policy drops 2.3× more than the camera-pose policy.

**Action Representations.** We compare the camera-pose actions described in Sec. 3.1 with joint-position actions on three teleoperated datasets in CamX: DROID (Khazatsky et al., 2024), AgiBot-World (Bu et al., 2025a), and ABC-130k (Allshire et al., 2026). Joint actions use a 7-D joint space with a single MLP encoder shared across all arms; actions for the 6-DoF YAM arms in ABC are zero-padded. Even on DROID alone, switching from camera-pose to joint actions lowers the RoboLab success rate (Fig. 5), although the base checkpoint was trained with joint actions. For cross-embodiment training, we sample the three datasets uniformly and train with three times the compute. Both action spaces yield lower success rates on DROID when the other two embodiments are added, but the joint-space policy drops by 8.1 points, more than twice the 3.5-point drop of the camera-pose policy. **FD** PSNR also favors camera pose on all three datasets. Camera pose thus gives a shared action space across embodiments, whereas joint actions must be learned per embodiment.

> ***Finding 1:*** *Camera pose outperforms joint space as a cross-embodiment action representation for policy learning, especially when training on heterogeneous robot data.*

**Camera Pose Encoding.** We compare the view-identification methods introduced in Sec. 3.2. By analogy with the "inter-gripper proprioception" of bimanual UMI (Chi et al., 2024), we hypothesize that injecting camera pose into the positional encoding improves bimanual coordination. To test this hypothesis, we build a `Shared Grasp` task in RoboLab using a bimanual YAM setup: the two arms must squeeze a box from both sides, lift it, and place it on a red target region whose distribution is overlaid in Fig. 6. All policies take both wrist-camera views as input and predict relative camera trajectories. Without pose information for view identification, the arms place the

box far less precisely: in most episodes, the box lands touching or outside the outline, lowering the success rate by 22 and 19 points.

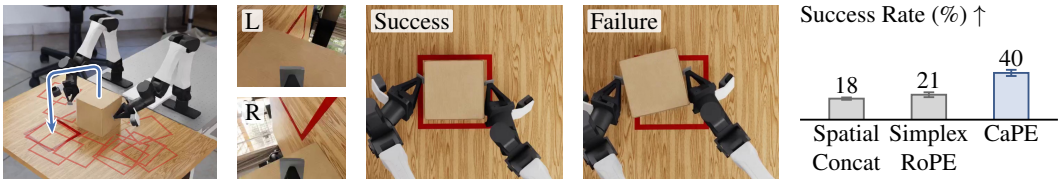



Figure 6: **Bimanual Coordination: Shared Grasp.** Both arms carry a box that is too wide for one gripper and place it on a random red target; success needs the whole footprint inside the outline (initial state: other episodes dimmed).

> ***Finding 2:*** *Identifying views by camera pose supports a flexible number of cameras (one checkpoint runs with one to four views in Sec. 4.2) and improves bimanual coordination.*

**Video Action Training Objectives.** We study how the UVA mode sampling strategy affects model performance while holding the 4k GPU-hour budget and batch size fixed. Surprisingly, naively sampling the five UVA modes with *Equal Probability* hurts both **P** and **PP** performance compared with **P**-only and **PP**-only training (Fig. 7).

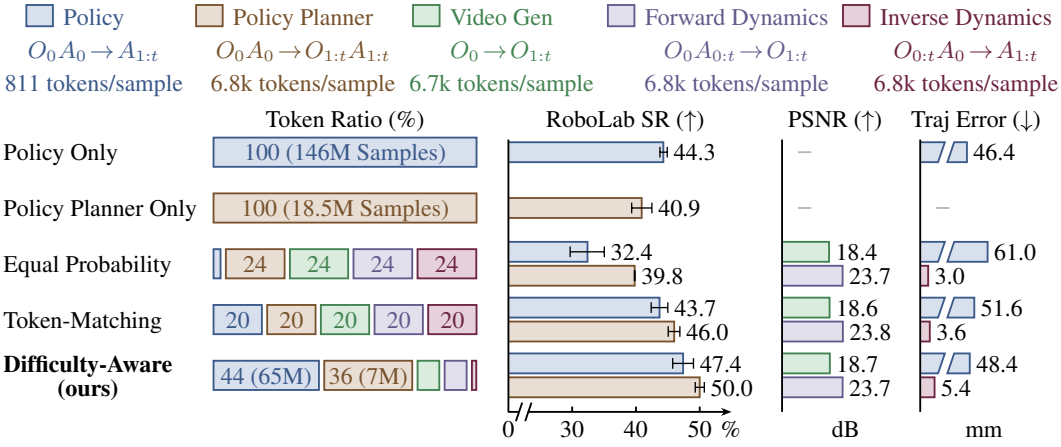



Figure 7: **UVA mode sampling strategies.** A policy sample contains one observation frame and thus 8.4× fewer tokens than a sample of the other four modes. Raising the policy sampling ratio (Token-Matching) recovers the policy success rate, and the Difficulty-Aware ratio lifts both the Policy and Policy Planner success rates above single-mode training while the video-generation and forward-dynamics PSNR stay unchanged.

We identify two main reasons. First, **a policy sample contains far fewer tokens because it includes only one observation frame**. Under *Equal Probability* sampling, policy samples therefore account for a much smaller share of tokens than the other modes, leading to poor performance. Increasing the sampling ratio of **P** to match the token count of the other modes (*Token-Matching*) brings **P** performance on par with **P**-only training and allows **PP** to surpass **PP**-only training. Second, **we use task difficulty as a sampling heuristic**: **P** and **PP** require task-relevant planning with potentially multimodal actions (Chi et al., 2023), so we prioritize these modes. We allocate fewer samples to **VG**, since **PP** also predicts future video, and to **FD** and **ID**, which condition on actions or future observations, respectively. These additional conditions can reduce uncertainty but do not make either task deterministic under partial observations; the difficulty ordering is a design hypothesis. With a hand-tuned *Difficulty-Aware* sampling ratio of 10:1:0.25:0.25:0.05, both the **P** and **PP** success rates exceed those of **P**-only and **PP**-only training, while **VG** and **FD** PSNR remain the same as under *Equal Probability* and *Token-Matching*. The **ID** trajectory error is slightly higher, but remains an order of magnitude lower than the **P** error.

> ***Finding 3:*** *Mixing multiple UVA training objectives according to their difficulty improves policy success rates while maintaining dynamics-model capability.*

**Model Scaling Study.** To study how performance scales with model size, we train a Cosmos3-Edge (NVIDIA, 2026) variant with a frozen 1.3B understanding branch, a fine-tuned 1.3B video branch, and a 0.5B action branch trained from scratch, totaling about 4B parameters (the text embedding and video encoder are the same as in the 16B Nano model). For both models, we vary the global batch size from 1 to 16 million tokens per iteration and train for up to 4k GPU-hours, evaluating Nano every 1k iterations and Edge every 2.5k iterations. Both models achieve similar GPU utilization and power draw during training.

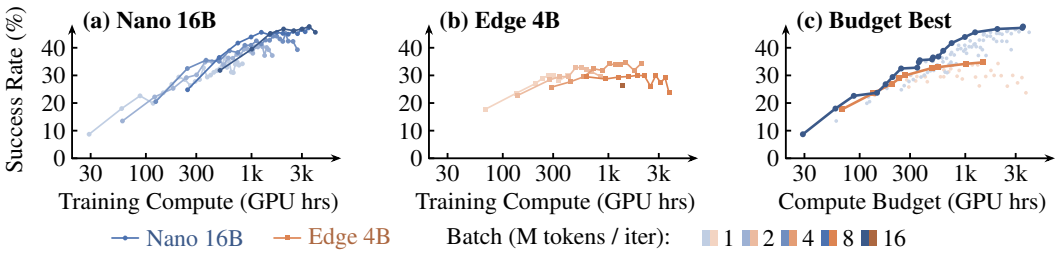



Figure 8: **Model Scaling study.** Policy-mode RoboLab success rate of **(a)** the five Nano and **(b)** the five Edge runs against the training compute consumed, one curve per global batch size, and **(c)** the best success rate either model reached within a compute budget (all checkpoints as faint dots).

Fig. 8(a–b) plots each run's RoboLab success rate against its training compute. The 16B Nano model saturates much more slowly than the 4B Edge model and reaches a much higher success rate. The larger 16B model also achieves higher peak performance with larger batches, whereas the smaller 4B model peaks at a 4M-token batch and gains nothing from a larger one under the same compute budget. In Fig. 8(c), we overlay the first two plots and highlight the best-performing checkpoint at each compute budget. At a low budget, the two models perform similarly, whereas the 16B Nano model is significantly stronger once the budget exceeds 1k GPU-hours. We attribute this advantage to the stronger prior and greater capacity of the larger pretrained backbone, which improve language grounding and task generalization, both crucial for the RoboLab evaluation.

> ***Finding 4:*** *The larger model outperforms the smaller one at high compute budgets and favors larger batches; the smaller model performs similarly at low budgets and favors more iterations.*

### 4.2 What's Possible? – Model Capability Study

Using the design choices from § 4.1, we initialize the final CamUVA model from Cosmos3-Nano-Policy-DROID and train it on the CamX training mix for 30k GB300 GPU-hours (Appendix B.4). Because all robots share the same camera-centric interface with flexible sequence packing, this single checkpoint controls single-arm, bimanual, and mobile robots with diverse camera views without any *embodiment-specific fine-tuning*. Moreover, each single-task benchmark dataset accounts for at most 1% of the training tokens, and we apply no *task-specific fine-tuning*. **All evaluations in this section use the same CamUVA checkpoint in the fastest Policy mode.**

§ 4.2.1 evaluates CamUVA on *in-distribution embodiments*, where robot and gripper exactly match the task's training data. § 4.2.2 evaluates CamUVA across embodiments, including ones seen only in other tasks *(embodiment transfer)* and ones unseen in training *(zero-shot cross-embodiment)*.

#### 4.2.1 In-Distribution Embodiment Evaluation

We evaluate the final CamUVA checkpoint on standard benchmarks and compare it with state-of-the-art methods optimized or trained on each individual benchmark.

In the simulated `RoboLab` (Yang et al., 2026b) benchmark, we compare CamUVA with four state-of-the-art baselines: two checkpoints post-trained on DROID (Cosmos3-Nano-Policy-DROID (NVIDIA, 2026) and $\pi_{0.5}$-DROID (Black et al., 2025a)) and two cross-embodiment models (LAP-3B (Zha et al., 2026) and Cloak (Piseno et al., 2026)). We evaluate each model on the full 120-task benchmark (10 episodes per task); Appendix D.1 details the rollout configurations, run counts, and uncertainty estimates. **Results:** In **P** mode (0.15 s latency, Appendix B.1), CamUVA outperforms the best cross-embodiment baseline by 32 points and is much faster than Cosmos3-

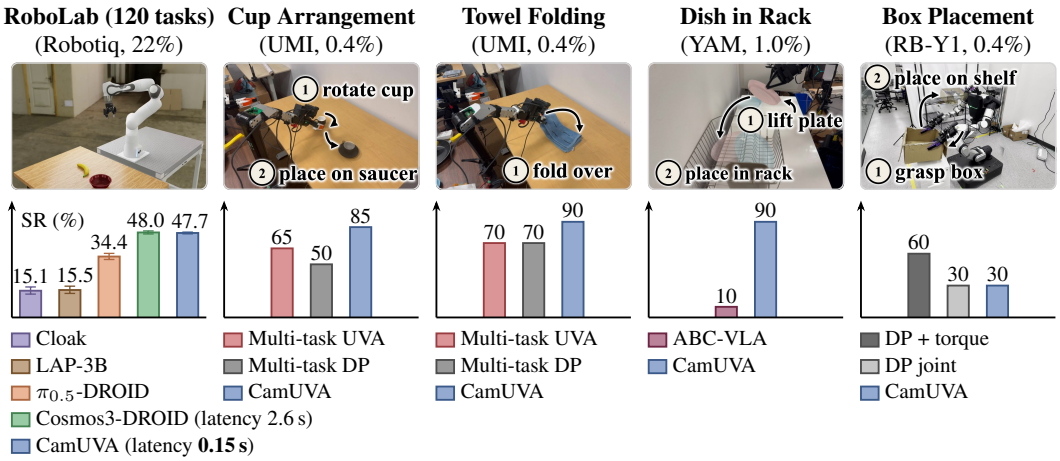



Figure 9: **CamUVA Performance (In-Distribution Embodiments).** All evaluations use a single CamUVA checkpoint without benchmark-specific post-training. Brackets name each task's gripper embodiment and share of CamUVA's training tokens. RoboLab whiskers are defined in Appendix D.1.

Nano-Policy-DROID, which supports only **PP** mode (2.6 s latency). This shows that the model retains strong task generalization and instruction following under cross-embodiment training.

In the real world, we test CamUVA on the standard single-arm `UMI` (Chi et al., 2024; Lin et al., 2025) setup using evaluation scenarios adapted from UVA (Li et al., 2025c) and compare against its reported results on the original evaluation set. We test both CamUVA and ABC-VLA (Allshire et al., 2026) on the `bimanual YAM` in the same in-the-wild environment with the same objects. We also run CamUVA on the `mobile RB-Y1` and compare its success rate with that reported by ModPack (Citron et al., 2026). All baselines are embodiment-specific models trained on the corresponding task, whereas CamUVA is a single cross-embodiment model, with each of these benchmark datasets accounting for at most 1% of the CamX training tokens (Fig. 9). **Results:** On `Cup Arrangement` and `Towel Folding`, CamUVA outperforms the multi-task UVA (Li et al., 2025c) and Diffusion Policy (Chi et al., 2023) specialists. In particular, CamUVA can recover from its mistakes, a behavior absent from the demonstrations of these tasks (Appendix E.5). On `Dish in Rack`, CamUVA loads two dishes consecutively with a 9/10 success rate in an unseen environment, while ABC-VLA can pick up the dishes but struggles to place them in the out-of-distribution environment (Appendix E.4.1). On `Box Placement`, CamUVA predicts camera trajectories and passes them to the IK-based whole-body controller of HoMMI (Xu et al., 2026) (Appendix E.3). Its success rate matches the reported 30% of ModPack's all-camera, torque-free joint-space Diffusion Policy, worse than 60% for the joint-torque variant (Citron et al., 2026).

> ***Finding 5:*** *A policy trained on the CamX mixture can be deployed directly on diverse training embodiments and achieves performance competitive with state-of-the-art models specifically optimized or post-trained on each individual benchmark.*

### 4.2.2 Cross-Embodiment Evaluation

We evaluate CamUVA's out-of-the-box embodiment generalization. Unlike variations in arm kinematics that can be abstracted away directly by the camera-centric action representation (Chi et al., 2024), **gripper** and **camera** variations require the cross-embodiment model to generalize.

Concretely, we test CamUVA on **four different tasks**, each with data collected using one specific embodiment: `Water Pouring` from UMI (Lin et al., 2025), `Cup Flipping` from iPhUMI (Patel et al., 2026; Gao et al., 2026), `Cup in Box` from the GenRobot V4 task collection (GenRobot, 2025), and `Banana in Pan` from the Robotiq gripper in the DROID (Khazatsky et al., 2024) and RoboMIND (Wu et al., 2025a) datasets. For each task, we evaluate the same checkpoint with **five different embodiments**: three present in CamX (**UMI**, **iPhUMI**, and **GenRobot**) and two unseen ones—a **new gripper with an iPhone camera** and the **UMI gripper with a RealSense camera** (in the training data, UMI grippers are paired only with GoPro and iPhone cameras), all mounted on the same ARX5 robot arm. Each bar in Fig. 10 therefore corresponds to one of three cases: the embodiment used to collect the task data (*in-distribution*), which serves as the reference; a seen em-

bodiment that never performed this task in CamX (*embodiment transfer*); or an *unseen embodiment*. For each bar, we evaluate 10 episodes with diverse backgrounds, objects, and locations to test the policy's robustness.

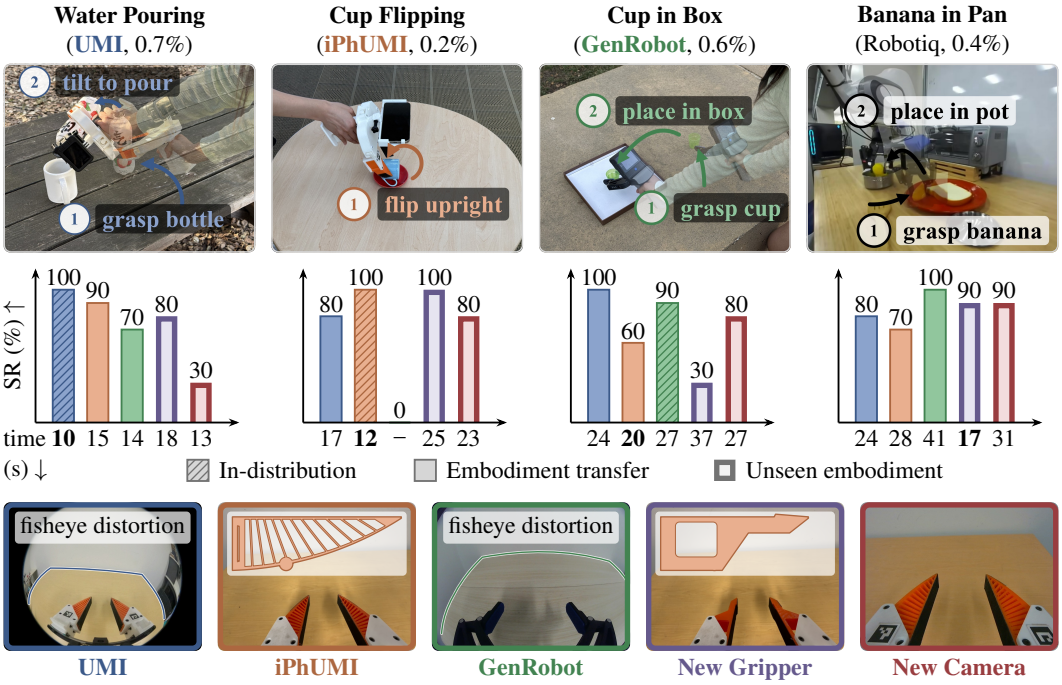



Figure 10: **Cross-embodiment task results.** Top: the four tasks, the gripper embodiment each was collected with and its share of the training tokens. Middle: success rate per task and embodiment; under each bar, the mean completion time of its successes. Bottom: the wrist views of each embodiment.

**Results:** CamUVA performs well on every task with its *in-distribution* embodiment, generally with shorter completion times. In cross-embodiment evaluation, we find that two properties matter most: gripper shape and camera type. A similar **gripper shape** implies similar interaction dynamics between the gripper and the objects: UMI and iPhUMI use identical gripper fingers and thus perform well on each other's task, while the GenRobot gripper is closest in shape to Robotiq, leading to the best performance on the Robotiq task. A similar **camera type (rectilinear or fisheye)** improves the model's spatial estimation: both GenRobot and the UMI GoPro use fisheye cameras and thus perform well on each other's task even though their grippers look completely different. The other three tested embodiments and Robotiq all use rectilinear cameras, so their cross-embodiment success rates are also higher. The largest embodiment gap appears between iPhUMI and GenRobot, where both the gripper shape and the camera type differ. Notably, the GenRobot gripper fails entirely on the `Cup Flipping` task because of errors in spatial and dynamics estimation (Appendix E.4.2).

> ***Finding 6:*** *After cross-embodiment training with a unified camera action space, task knowledge learned on one embodiment transfers to other training embodiments and unseen embodiments with high success rates. Similar gripper shapes and camera types improve transferability.*

## 5 Conclusion

20k hours of open-source robot demonstrations can train one model once they share a camera-centric interface: CamX converts 37 sources into camera trajectories and pose-identified views, and CamUVA consumes them with no robot-specific module. Camera-pose actions, a difficulty-aware mix of the UVA objectives and pose-based view identification each matter, and compute on a larger backbone remains the dominant lever. One checkpoint trained on the CamX mixture runs on diverse embodiments, is competitive with specialist policies, and transfers task knowledge to unseen gripper-and-camera pairs without new data.

## AI USE STATEMENT

In this work we used generative AI tools (LLM-based coding assistants) to draft and revise the figures, to polish the writing of the manuscript, to check the text against our experiment logs for inconsistencies, and to verify the bibliography entries. No dataset, experiment, result, claim or analysis in this paper was generated by AI; the research ideas, the CamX conversion pipeline, the model and the evaluations are the authors' own. Every AI-assisted piece of text and code was reviewed by at least one author, and figure code was checked against the underlying logged numbers. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

## ETHICS STATEMENT

CamX includes public dataset releases and research collections; access and redistribution remain subject to the terms of each source; we will release converted camera poses and metadata only where the source license permits it. We collected no new human-subject data, and the handheld sources contain the hands of the original demonstrators in the source recordings. CamUVA controls physical robots; all real-world rollouts were run with a human operator holding an emergency stop, and the model is a research artifact that is not validated for unsupervised deployment. We see no other ethics concern beyond those common to learned robot policies.

## REPRODUCIBILITY STATEMENT

Camera-pose conversion and validation are described in Sec. 3.1, action parameterization in Appendix B.3, and the model architecture, pose encoding, sequence packing and training objectives in Sec. 3.2. Appendix B.4 gives the training and evaluation protocol of the ablation experiments and the recipe of the final model, and Appendix D.1 the rollout configuration of every baseline. The real-world tasks and embodiments are shown in Figs. 9 and 10, and their setups and success criteria are specified in Appendix E. We will release the CamX conversion pipeline, source manifest, the CamUVA code and the evaluated checkpoint upon publication, together with processed data where the source terms permit redistribution.

## REFERENCES

AgiBot World Team. AgiBot World 2026. https://huggingface.co/datasets/agibot-world/AgiBotWorld2026, 2026.

Arthur Allshire, Himanshu Gaurav Singh, Ritvik Singh, Adam Rashid, Hongsuk Choi, et al. Scalable behavior cloning with open data, training, and evaluation. *arXiv preprint arXiv:2606.27375*, 2026. URL https://arxiv.org/abs/2606.27375.

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

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

Homanga Bharadhwaj, Jay Vakil, Mohit Sharma, Abhinav Gupta, Shubham Tulsiani, et al. RoboAgent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024. URL https://robopen.github.io/.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. In *Conference on Robot Learning (CoRL)*, 2025a. URL https://proceedings.mlr.press/v305/black25a.html.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, et al. $\pi_0$: A vision-language-action flow model for general robot control. In *Robotics: Science and Systems (RSS)*, 2025b. URL https://www.roboticsproceedings.org/rss21/p010.html.

Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Xindong He, Xu Huang, et al. AgiBot World Colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025a. URL https://github.com/OpenDriveLab/AgiBot-World.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, et al. Learning to act anywhere with task-centric latent actions. In *Robotics: Science and Systems (RSS)*, 2025b. URL https://www.roboticsproceedings.org/rss21/p014.html.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, et al. WorldVLA: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025. URL https://arxiv.org/abs/2506.21539.

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, et al. GR-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024. URL https://arxiv.org/abs/2410.06158.

Anzhe Chen, Yifei Yang, Zhenjie Zhu, Kechun Xu, Zhongxiang Zhou, Rong Xiong, and Yue Wang. Toward embodiment equivariant vision-language-action policy. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026a. URL https://ras.papercept.net/conferences/conferences/ICRA26/program/ICRA26_ContentListWeb_4.html#wei1i_87.

Tianxing Chen, Yue Chen, Zixuan Li, Junyuan Tang, Kailun Su, et al. RoboDojo: A unified sim-and-real benchmark for comprehensive evaluation of generalist robot manipulation policies. *arXiv preprint arXiv:2607.04434*, 2026b. URL https://arxiv.org/abs/2607.04434.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, et al. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*, 2023. URL https://www.roboticsproceedings.org/rss19/p026.html.

Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, et al. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Robotics: Science and Systems (RSS)*, 2024. URL https://www.roboticsproceedings.org/rss20/p045.html.

Hojung Choi, Yifan Hou, Chuer Pan, Seongheon Hong, Austin Patel, Xiaomeng Xu, Mark R. Cutkosky, and Shuran Song. In-the-wild compliant manipulation with UMI-FT. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026. URL https://umi-ft.github.io/.

Joshua Citron, Renee Zbizika, Zeyi Liu, and Shuran Song. ModPack: An extensible teleoperation interface for bimanual mobile manipulation. *arXiv preprint arXiv:2607.19479*, 2026. URL https://arxiv.org/abs/2607.19479.

Jaden Clark, Changhao Wang, Yihuai Gao, Seongheon Hong, Hojung Choi, Mark Cutkosky, Yifan Hou, and Shuran Song. Multisensory continual learning: Adapting pretrained visuomotor policies to force. In *CoRL*, 2026. URL https://arxiv.org/abs/2606.30988. To appear.

Daimon Robotics. Daimon-Infinity: DM-DataClaw dataset. https://modelscope.cn/datasets/daimonrobotics/Daimon-Infinity, 2026. DM-DataClaw subset; accessed 2026-09-25.

Sudeep Dasari, Oier Mees, Sebastian Zhao, Mohan Kumar Srirama, and Sergey Levine. The ingredients for robotic diffusion transformers. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025. URL https://dit-policy.github.io/.

Juechu Dong, Boyuan Feng, Driss Guessous, Yanbo Liang, and Horace He. FlexAttention: A programming model for generating fused attention variants. In *Proceedings of Machine Learning and Systems (MLSys)*, 2025. URL https://proceedings.mlsys.org/paper_files/paper/2025/hash/61a9278dfef5f871b5e472389f8d6fa1-Abstract-Conference.html.

Ria Doshi, Homer Rich Walke, Oier Mees, Sudeep Dasari, and Sergey Levine. Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation. In *Conference on Robot Learning (CoRL)*, 2025. URL https://proceedings.mlr.press/v270/doshi25a.html.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, et al. Learning universal policies via text-guided video generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/1d5b9233ad716a43be5c0d3023cb82d0-Abstract-Conference.html.

Enactic, Inc. OpenArm. https://github.com/enactic/openarm, n.d. Hardware project; accessed 2026-09-25.

Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Chenxi Wang, et al. RH20T: A comprehensive robotic dataset for learning diverse skills in one-shot. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024. URL https://rh20t.github.io/.

Haoquan Fang, Jiafei Duan, Donovan Clay, Sam Wang, Shuo Liu, et al. MolmoAct2: Action reasoning models for real-world deployment. *arXiv preprint arXiv:2605.02881*, 2026. URL https://arxiv.org/abs/2605.02881.

Zipeng Fu, Tony Z. Zhao, and Chelsea Finn. Mobile ALOHA: Learning bimanual mobile manipulation using low-cost whole-body teleoperation. In *Conference on Robot Learning (CoRL)*, pp. 4066–4083, 2025. URL https://proceedings.mlr.press/v270/fu25b.html.

Yihuai Gao, Jeff Jinyun Liu, Shuang Li, and Shuran Song. Gated memory policy: In-context memorization and adaptation. *arXiv preprint arXiv:2604.18933*, 2026. URL https://arxiv.org/abs/2604.18933.

GenRobot. 10Kh-RealOmin-OpenData. https://huggingface.co/datasets/genrobot2025/10Kh-RealOmin-OpenData, 2025.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, et al. Octo: An open-source generalist robot policy. In *Robotics: Science and Systems (RSS)*, 2024. URL https://www.roboticsproceedings.org/rss20/p090.html.

Huy Ha, Yihuai Gao, Zipeng Fu, Jie Tan, and Shuran Song. UMI-on-Legs: Making manipulation policies mobile with manipulation-centric whole-body controllers. In *Conference on Robot Learning (CoRL)*, 2025. URL https://proceedings.mlr.press/v270/ha25a.html.

Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. CameraCtrl: Enabling camera control for video diffusion models. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/f98fd73d59d8494489ea970747b91fe4-Abstract-Conference.html.

Chengkai Hou, Kun Wu, Jiaming Liu, Zhengping Che, Di Wu, et al. RoboMIND 2.0: A multimodal, bimanual mobile manipulation dataset for generalizable embodied intelligence. *arXiv preprint arXiv:2512.24653*, 2025. URL https://arxiv.org/abs/2512.24653.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, et al. Video prediction policy: A generalist robot policy with predictive visual representations. In *International Conference on Machine Learning (ICML)*, 2025. URL https://proceedings.mlr.press/v267/hu25g.html.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, et al. DreamGen: Unlocking generalization in robot learning through video world models. In *Conference on Robot Learning (CoRL)*, 2025. URL https://proceedings.mlr.press/v305/jang25a.html.

Tao Jiang, Tianyuan Yuan, Yicheng Liu, Chenhao Lu, Jianning Cui, et al. Galaxea open-world dataset and G0 dual-system VLA model. *arXiv preprint arXiv:2509.00576*, 2025. URL https://arxiv.org/abs/2509.00576.

Tianchong Jiang, Jingtian Ji, Xiangshan Tan, Jiading Fang, Anand Bhattad, Vitor Guizilini, and Matthew R. Walter. Do you know where your camera is? View-invariant policy learning with camera conditioning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026. URL https://ras.papercept.net/conferences/conferences/ICRA26/program/ICRA26_ContentListWeb_3.html#tuat1_04.

Shi Jin, Yuntian Wang, Yuhui Duan, Di Wu, Gaoqi Dong, et al. UMI-Bench 1.0: An open and reproducible real-world benchmark for tabletop robotic manipulation with UMI data. *arXiv preprint arXiv:2606.10382*, 2026. URL https://arxiv.org/abs/2606.10382.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, et al. DROID: A large-scale in-the-wild robot manipulation dataset. In *Robotics: Science and Systems (RSS)*, 2024. URL https://www.roboticsproceedings.org/rss20/p120.html.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, et al. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, et al. Cosmos Policy: Fine-tuning video models for visuomotor control and planning. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/748becc400a57c0e31cfe6a2e7951467-Abstract-Conference.html.

Xin Kong, Shikun Liu, Xiaoyang Lyu, Marwan Taher, Xiaojuan Qi, and Andrew J. Davison. EscherNet: A generative model for scalable view synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Kong_EscherNet_A_Generative_Model_for_Scalable_View_Synthesis_CVPR_2024_paper.html.

Geonhyup Lee, Youngjin Lee, Kangmin Kim, Seongju Lee, Sangjun Noh, Seunghyeok Back, and Kyoobin Lee. ManipForce: Force-guided policy learning with frequency-aware representation for contact-rich manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026a. URL https://ras.papercept.net/conferences/conferences/ICRA26/program/ICRA26_ContentListWeb_5.html#thi2i_257.

Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, Winson Han, Wilbert Pumacay, Angelica Wu, Rose Hendrix, Karen Farley, Eli VanderBilt, Ali Farhadi, Dieter Fox, and Ranjay Krishna. MolmoAct: Action reasoning models that can reason in space. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026b. URL https://ras.papercept.net/conferences/conferences/ICRA26/program/ICRA26_ContentListWeb_3.html#tui1i_327.

Chuanyu Li, Chaoyi Liu, Daotan Wang, Shuyu Zhang, Lusong Li, et al. ViTaMIn-B: A reliable and efficient visuo-tactile bimanual manipulation interface. *arXiv preprint arXiv:2511.05858*, 2025a. URL https://arxiv.org/abs/2511.05858.

Hong Li, Yue Xu, Yihan Tang, Yankang Dong, Chenyuan Liu, Chenyang Yu, Xuyang Li, Siyuan Huang, Yujun Shen, Nan Xue, and Yong-Lu Li. AetheRock: An arm-worn robot teaching system for force-guided vision-tactile learning. *arXiv preprint arXiv:2606.09777*, 2026a. URL https://arxiv.org/abs/2606.09777.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, et al. Causal world modeling for robot control. In *Robotics: Science and Systems (RSS)*, 2026b. URL https://www.roboticsproceedings.org/rss22/p016.html.

Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 15984–16009. Curran Associates, Inc., 2025b. doi: 10.52202/085713-0540. URL https://papers.nips.cc/paper_files/paper/2025/hash/17a7075094632c88cccdd86270ad715b-Abstract-Conference.html.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. In *Robotics: Science and Systems (RSS)*, 2025c. URL https://www.roboticsproceedings.org/rss21/p074.html.

Wenhao Li, Xueying Jiang, Quanhao Qian, Deli Zhao, Shijian Lu, Gongjie Zhang, and Ran Xu. From fixed to free cameras: Calibration-free view-robust vision-language-action model. *arXiv preprint arXiv:2607.05396*, 2026c. URL https://arxiv.org/abs/2607.05396.

Weixin Liang, Lili Yu, Liang Luo, Srini Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen-tau Yih, Luke Zettlemoyer, and Xi Victoria Lin. Mixture-of-Transformers: A sparse and scalable architecture for multi-modal foundation models. *Transactions on Machine Learning Research (TMLR)*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=Nu6N69i8SB.

Fanqi Lin, Yingdong Hu, Pingyue Sheng, Chuan Wen, Jiacheng You, and Yang Gao. Data scaling laws in imitation learning for robotic manipulation. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/88b7b2c896506daabc8d3fd587055167-Abstract-Conference.html.

Tao Lin, Yilei Zhong, Yuxin Du, Jingjing Zhang, Jiting Liu, Yinxinyu Chen, Encheng Gu, Ziyan Liu, Hongyi Cai, Yanwen Zou, Lixing Zou, Zhaoye Zhou, Gen Li, and Bo Zhao. Evo-1: Lightweight vision-language-action model with preserved semantic alignment. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13397–13406, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Lin_Evo-1_Lightweight_Vision-Language-Action_Model_with_Preserved_Semantic_Alignment_CVPR_2026_paper.html.

Fangchen Liu, Chuanyu Li, Yihua Qin, Jing Xu, Pieter Abbeel, et al. ViTaMIn: Learning contact-rich tasks through robot-free visuo-tactile manipulation interface. *arXiv preprint arXiv:2504.06156*, 2025a. URL https://arxiv.org/abs/2504.06156.

Fangfu Liu, Kai He, Tianchang Shen, Tianshi Cao, Sanja Fidler, et al. Gamma-World: Generative multi-agent world modeling beyond two players. *arXiv preprint arXiv:2605.28816*, 2026a. URL https://arxiv.org/abs/2605.28816.

Kehui Liu, Zhongjie Jia, Yang Li, Zhaxizhuoma, Pengan Chen, et al. FastUMI-100K: Advancing data-driven robotic manipulation with a large-scale UMI-style dataset. *arXiv preprint arXiv:2510.08022*, 2025b. URL https://arxiv.org/abs/2510.08022.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, et al. RDT-1B: A diffusion foundation model for bimanual manipulation. In *International Conference on Learning Representations (ICLR)*, 2025c. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/49f80e4d2471ad4f2edf4f5f1ab62339-Abstract-Conference.html.

Songming Liu, Bangguo Li, Kai Ma, Lingxuan Wu, Hengkai Tan, et al. RDT2: Exploring the scaling limit of UMI data towards zero-shot cross-embodiment generalization. In *International Conference on Machine Learning (ICML)*, 2026b. URL https://icml.cc/virtual/2026/poster/65782.

Zeyi Liu, Cheng Chi, Eric Cousineau, Naveen Kuppuswamy, Benjamin Burchfiel, et al. ManiWAV: Learning robot manipulation from in-the-wild audio-visual data. In *Conference on Robot Learning (CoRL)*, 2025d. URL https://proceedings.mlr.press/v270/liu25c.html.

Jianlan Luo, Charles Xu, Fangchen Liu, Liam Tan, Zipeng Lin, et al. FMB: A functional manipulation benchmark for generalizable robotic learning. *International Journal of Robotics Research (IJRR)*, 44(4):592–606, 2025. doi: 10.1177/02783649241276017. URL https://doi.org/10.1177/02783649241276017.

Takeru Miyato, Bernhard Jaeger, Max Welling, and Andreas Geiger. GTA: A geometry-aware attention mechanism for multi-view transformers. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/20e6b4dd2b1f82bc599c593882f67f75-Abstract-Conference.html.

Tomohiro Motoda, Masaki Murooka, Ryoichi Nakajo, Muhammad A. Muttaqien, Koshi Makihara, et al. AIST bimanual manipulation dataset. https://aistairc.github.io/aist_bimanip_site/, 2025.

Ruiqian Nai, Boyuan Zheng, Junming Zhao, Haodong Zhu, Sicong Dai, Zunhao Chen, Yihang Hu, Yingdong Hu, Tong Zhang, Chuan Wen, and Yang Gao. Humanoid manipulation interface: Humanoid whole-body manipulation from robot-free demonstrations. *arXiv preprint arXiv:2602.06643*, 2026. URL https://arxiv.org/abs/2602.06643.

NeoteAI Team and Fudan TEAI Team. $\mathcal{N}_0$-Foundation: Towards the age of tactile intelligence. *arXiv preprint arXiv:2608.29601*, 2026. URL https://arxiv.org/abs/2608.29601.

NVIDIA. Cosmos world foundation model platform for physical AI. *arXiv preprint arXiv:2501.03575*, 2025a. URL https://arxiv.org/abs/2501.03575.

NVIDIA. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025b. URL https://arxiv.org/abs/2503.14734.

NVIDIA. Cosmos 3: Omnimodal world models for physical AI. *arXiv preprint arXiv:2606.02800*, 2026. URL https://arxiv.org/abs/2606.02800.

Open X-Embodiment Collaboration. Open X-embodiment: Robotic learning datasets and RT-X models. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903, 2024. doi: 10.1109/ICRA57147.2024.10611477. URL https://doi.org/10.1109/ICRA57147.2024.10611477.

Austin Patel, Ben Pekarek, Joel Enrique Castro Hernandez, and Shuran Song. Behavior prompting policy: Demonstrations as prompts for manipulation. *arXiv preprint arXiv:2606.30457*, 2026. URL https://arxiv.org/abs/2606.30457.

Michael Piseno, Guy Tevet, and C. Karen Liu. Cloak: Zero-shot cross-embodiment manipulation by masking the end-effector from the VLA. *arXiv preprint arXiv:2606.22836*, 2026. URL https://arxiv.org/abs/2606.22836.

Omar Rayyan, John Abanes, Mahmoud Hafez, Anthony Tzes, and Fares Abu-Dakka. MV-UMI: A scalable multi-view interface for cross-embodiment learning. *arXiv preprint arXiv:2509.18757*, 2025. URL https://arxiv.org/abs/2509.18757.

RealSource. RealSource World: A large-scale real-world dual-arm manipulation dataset. https://huggingface.co/datasets/RealmanRobot/RealSource-World, 2025.

Mingyo Seo, H. Andy Park, Shenli Yuan, Yuke Zhu, and Luis Sentis. LEGATO: Cross-embodiment imitation using a grasping tool. *IEEE Robotics and Automation Letters*, 10(3):2854–2861, 2025. doi: 10.1109/LRA.2025.3535182. URL https://doi.org/10.1109/LRA.2025.3535182.

Nur Muhammad Mahi Shafiullah, Anant Rai, Haritheja Etukuru, Yiqian Liu, Ishan Misra, Soumith Chintala, and Lerrel Pinto. On bringing robots home. *arXiv preprint arXiv:2311.16098*, 2023. URL https://arxiv.org/abs/2311.16098.

Simple AI. HiFi-UMI: Learning deployable manipulation policies from high-fidelity UMI data alone. *arXiv preprint arXiv:2607.25895*, 2026. URL https://arxiv.org/abs/2607.25895.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/j.neucom.2023.127063. URL https://doi.org/10.1016/j.neucom.2023.127063.

Tony Tao, Mohan Kumar Srirama, Jason Jingzhou Liu, Kenneth Shaw, and Deepak Pathak. DexWild: Dexterous human interactions for in-the-wild robot policies. In *Robotics: Science and Systems (RSS)*, 2025. doi: 10.15607/RSS.2025.XXI.075. URL https://www.roboticsproceedings.org/rss21/p075.html.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. URL https://arxiv.org/abs/2503.20314.

Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. BridgeData V2: A dataset for robot learning at scale. In *Conference on Robot Learning (CoRL)*, pp. 1723–1736, 2023. URL https://proceedings.mlr.press/v229/walke23a.html.

Dian Wang, Jisang Park, Xiaomeng Xu, Han Zhang, Shuran Song, and Jeannette Bohg. Mixture of frames policy: Multi-frame action denoising for bimanual mobile manipulation. In *Conference on Robot Learning (CoRL)*, 2026. URL https://mofpo.github.io/. To appear.

Lirui Wang, Xinlei Chen, Jialiang Zhao, and Kaiming He. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024a. URL https://papers.nips.cc/paper_files/paper/2024/hash/e0f393e7980a24fd12fa6f15adfa25fb-Abstract-Conference.html.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, et al. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024b. URL https://arxiv.org/abs/2409.12191.

Ziming Wang. UMI-3D: Extending universal manipulation interface from vision-limited to 3D spatial perception. *arXiv preprint arXiv:2604.14089*, 2026. URL https://arxiv.org/abs/2604.14089.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, et al. Unleashing large-scale video generative pre-training for visual robot manipulation. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/2c37c5bcef24b9541550261dcd63261b-Abstract-Conference.html.

Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, et al. RoboMIND: Benchmark on multi-embodiment intelligence normative data for robot manipulation. In *Robotics: Science and Systems (RSS)*, 2025a. URL https://www.roboticsproceedings.org/rss21/p152.html.

Longyan Wu, Jieji Ren, Chenghang Jiang, Junxi Zhou, Shijia Peng, Ran Huang, Guoying Gu, Li Chen, and Hongyang Li. TAMEn: Tactile-aware manipulation engine for closed-loop data collection in contact-rich tasks. *arXiv preprint arXiv:2604.07335*, 2026a. URL https://arxiv.org/abs/2604.07335.

Longyan Wu, Checheng Yu, Jieji Ren, Li Chen, Yufei Jiang, et al. FreeTacMan: Robot-free visuo-tactile data collection system for contact-rich manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026b. URL https://ras.papercept.net/conferences/conferences/ICRA26/program/ICRA26_ContentListWeb_3.html#tui2i_177.

Shihan Wu, Xuecheng Liu, Shaoxuan Xie, Pengwei Wang, Xinghang Li, et al. RoboCOIN: An open-sourced bimanual robotic data collection for integrated manipulation. *arXiv preprint arXiv:2511.17441*, 2025b. URL https://arxiv.org/abs/2511.17441.

Sicheng Xie, Lingchen Meng, Zijie Diao, Haidong Cao, Zhiying Du, et al. Unify robot actions in camera frame. *arXiv preprint arXiv:2511.17001*, 2025. URL https://arxiv.org/abs/2511.17001.

Mengda Xu, Han Zhang, Yifan Hou, Zhenjia Xu, Linxi Fan, Manuela Veloso, and Shuran Song. DexUMI: Using human hand as the universal manipulation interface for dexterous manipulation. In *Conference on Robot Learning (CoRL)*, pp. 437–459, 2025a. URL https://proceedings.mlr.press/v305/xu25b.html.

Xiaomeng Xu, Jisang Park, Han Zhang, Eric Cousineau, Aditya Bhat, Jose Barreiros, Dian Wang, Jeannette Bohg, and Shuran Song. HoMMI: Learning whole-body mobile manipulation from human demonstrations. In *Robotics: Science and Systems (RSS)*, 2026. URL https://www.roboticsproceedings.org/rss22/p205.html.

Yue Xu, Litao Wei, Pengyu An, Qingyu Zhang, and Yong-Lu Li. exUMI: Extensible robot teaching system with action-aware task-agnostic tactile representation. In *Conference on Robot Learning (CoRL)*, 2025b. URL https://proceedings.mlr.press/v305/xu25e.html.

Siyuan Yang, Linzheng Guo, Ouyang Lu, Zhaxizhuoma, Daoran Zhang, et al. VISTA: Vision-grounded and physics-validated adaptation of UMI data for VLA training. *arXiv preprint arXiv:2606.04708*, 2026a. URL https://arxiv.org/abs/2606.04708.

Xuning Yang, Rishit Dagli, Alex Zook, Hugo Hadfield, Ankit Goyal, et al. RoboLab: A high-fidelity simulation benchmark for analysis of task generalist policies. In *Robotics: Science and Systems (RSS)*, 2026b. URL https://www.roboticsproceedings.org/rss22/p096.html.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, et al. Latent action pretraining from videos. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/45d74e190008c7bff2845ffc8e3facd3-Abstract-Conference.html.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026. URL https://arxiv.org/abs/2602.15922.

Haoqi Yuan, Zhixuan Liang, Anzhe Chen, Ye Wang, Haoyang Li, et al. Qwen-RobotManip technical report: Alignment unlocks scale for robotic manipulation foundation models. *arXiv preprint arXiv:2606.17846*, 2026a. URL https://arxiv.org/abs/2606.17846.

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*, 2026b. URL https://arxiv.org/abs/2603.16666.

Lihan Zha, Asher James Hancock, Mingtong Zhang, Tenny Yin, Yixuan Huang, Dhruv Shah, Allen Z. Ren, and Anirudha Majumdar. LAP: Language-action pre-training enables zero-shot cross-embodiment transfer. In *Robotics: Science and Systems (RSS)*, 2026. URL https://www.roboticsproceedings.org/rss22/p203.html.

Cheng Zhang, Boying Li, Meng Wei, Yan-Pei Cao, Camilo Cruz Gambardella, Dinh Phung, and Jianfei Cai. Unified camera positional encoding for controlled video generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026a. URL https://openaccess.thecvf.com/content/CVPR2026/html/Zhang_Unified_Camera_Positional_Encoding_for_Controlled_Video_Generation_CVPR_2026_paper.html.

Jason Y. Zhang, Amy Lin, Moneish Kumar, Tzu-Hsuan Yang, Deva Ramanan, and Shubham Tulsiani. Cameras as rays: Pose estimation via ray diffusion. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/66175cc3d241c1702521dd836cd08585-Abstract-Conference.html.

Tianle Zhang et al. JoyAI-RA 0.1: A foundation model for robotic autonomy. *arXiv preprint arXiv:2604.20100*, 2026b. URL https://arxiv.org/abs/2604.20100.

Zongzheng Zhang, Jingrui Pang, Zhuo Yang, Kun Li, Minwen Liao, Saining Zhang, Guoxuan Chi, Jinbang Guo, Huan-ang Gao, Modi Shi, Dongyun Ge, Yao Mu, Jiayuan Gu, Rui Chen, Hao Dong, Huazhe Xu, Li Yi, Yixin Zhu, Hang Zhao, Pengwei Wang, Shanghang Zhang, Guocai Yao, Jianyu Chen, Hongyang Li, and Hao Zhao. Dexora: Open-source VLA for high-DoF bimanual dexterity. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026c. URL https://ras.papercept.net/conferences/conferences/ICRA26/program/ICRA26_ContentListWeb_4.html#weat1_01.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems (RSS)*, 2023a. URL https://www.roboticsproceedings.org/rss19/p016.html.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. UniPC: A unified predictor-corrector framework for fast sampling of diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023b. URL https://papers.nips.cc/paper_files/paper/2023/hash/9c2aa1e456ea543997f6927295196381-Abstract-Conference.html.

Zhaxizhuoma, Kehui Liu, Jeff Guan, Zhongjie Jia, Ziniu Wu, et al. FastUMI: A scalable and hardware-independent universal manipulation interface with dataset. In *Conference on Robot Learning (CoRL)*, 2025. URL https://proceedings.mlr.press/v305/zhaxizhuoma25a.html.

Jinliang Zheng, Jianxiong Li, Dongxiu Liu, Yinan Zheng, Zhihao Wang, et al. Universal actions for enhanced embodied foundation models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Zheng_Universal_Actions_for_Enhanced_Embodied_Foundation_Models_CVPR_2025_paper.html.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, et al. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/630d293833e09e1ecd892a898a20b074-Abstract-Conference.html.

Pengfei Zhou, Shengcong Chen, Di Chen, Jiaxu Wang, Rongjun Jin, et al. $\tau_0$-WM: A unified video-action world model for robotic manipulation. *arXiv preprint arXiv:2606.01027*, 2026. URL https://arxiv.org/abs/2606.01027.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Zhou_On_the_Continuity_of_Rotation_Representations_in_Neural_Networks_CVPR_2019_paper.html.

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 *Robotics: Science and Systems (RSS)*, 2025a. URL https://www.roboticsproceedings.org/rss21/p015.html.

Xinyue Zhu, Binghao Huang, and Yunzhu Li. Touch in the wild: Learning fine-grained manipulation with a portable visuo-tactile gripper. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025b. URL https://papers.neurips.cc/paper_files/paper/2025/hash/e1fcd183ab33714a8464e4e9a20ac710-Abstract-Conference.html.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning (CoRL)*, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

# A Limitations and Future Work

Camera-centric actions require a camera rigidly mounted on each end-effector group. Datasets that provide only a head-mounted (e.g., egocentric) or third-person camera can therefore be used for video-generation co-training, but not as supervision for action generation. CamX includes only grippers as end-effectors, not dexterous or human hands; future work could extend the model by replacing the gripper-width MLP encoder, while wrist motion would still be controlled through the wrist-camera pose. Finally, due to compute limits, the scaling study covers only two model sizes, both initialized from pretrained models rather than trained from scratch.

# B CamUVA Model Details

## B.1 Model Inference Latency

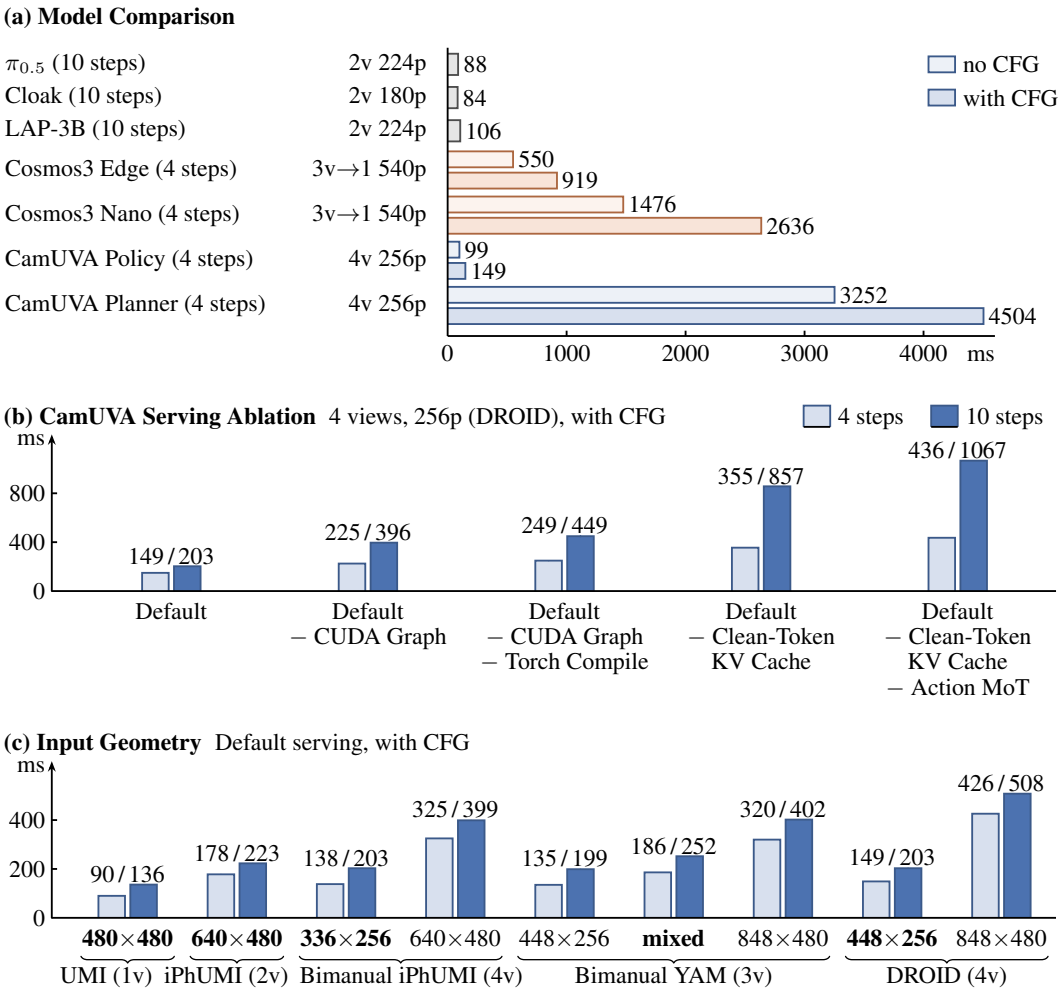



Figure 11: **Inference latency on one RTX 5090.** Mean of 20 requests after warm-up; the ±1 std spread is below the line width everywhere. (a) Every policy as deployed in RoboLab, with its input geometry (v = views; Cosmos3 stitches three cameras into one 540p frame). CamUVA **Policy** mode does not introduce significant latency compared to VLA baselines. (b) Both serving optimization (Torch compile, CUDA graph) and model architecture (Fast-WAM-like clean-token KV Caching, separate 1B action MoT instead of sharing the 7B video branch) contributes to the fast policy inference speed. (c) CamUVA is trained with both 256p and 480p resolutions and multiple aspect ratios (16:9, 4:3, 1:1, 16:13 etc.) with augmentation. It also supports mixed-resolution (head camera 480p and wrist cameras 256p) for some of the embodiments. During deployment, we choose the one with highest quality while maintaining short inference latency (bolded).

Fig. 11 measures the wall-clock latency of one action-chunk request on a single RTX 5090, each number the mean of 20 requests after warm-up. For both Cosmos3 (NVIDIA, 2026) and CamUVA, we cache the text tokens of the understanding branch, so only the video and action branches need to be loaded at inference, which fits comfortably on a single 5090 with bf16 weights. Note that with the Fast-WAM-style clean-token KV cache (Yuan et al., 2026b) and a separate 1B action branch (instead of sharing the 7B video branch), each action denoising step becomes cheap, so running 10 or more denoising steps is feasible at deployment.

### B.2 Camera Pose Encoding Details

We apply CaPE (Kong et al., 2024) to the lower-frequency half of the channels of the interleaved 3D MRoPE (Wang et al., 2024b; Bai et al., 2025), as a design choice motivated by their slower phase variation over our short sequences and low-resolution video grids. We group every four channels of the query and key of each token and apply the $SE(3)$ transform of its camera pose $P$ to each group, $q \leftarrow P^{-\top} q$ and $k \leftarrow Pk$. The main camera trajectory and the gripper width of an actuator group use the pose of its main camera; stereo and other rigidly attached cameras derive their poses from the main camera pose through a fixed transform, $P_{\text{attached}} = P_{\text{main}}\, T_{\text{attached}}$. In all UVA modes, every token of a chunk uses the camera pose of its first frame, even in **Inverse Dynamics**, where the future camera poses are given. We use camera poses rather than dense ray representations (Zhang et al., 2024; He et al., 2025) or projective encodings such as PRoPE (Li et al., 2025b), since accurate camera intrinsics and lens-distortion calibration are not guaranteed in publicly available datasets. Ray-based encodings can represent fisheye cameras when the lens model is known (Zhang et al., 2026a).

### B.3 Action Space Details

For the camera trajectories, we use anchored relative poses with respect to the starting frame of each chunk: given the camera poses $P_0, \ldots, P_t \in SE(3)$ with $t = 32$, the action at step $k$ is $A_k = P_0^{-1} P_k$ for $k = 0, \ldots, t$, so that the conditioning pose $A_0 = I$ is always the identity. One option is to feed a one-step velocity in place of $A_0 = I$ as the conditioning action, but we find that the model frequently replans after misgrasps, where a velocity input may lead to overfitting and hurt the corrective behavior. For the rotation, we use the first two columns of the rotation matrix (Zhou et al., 2019), which is equivalent to the two-row convention of UMI and Diffusion Policy (Chi et al., 2023; 2024) as long as it is applied consistently. Compared to frame-wise deltas $P_{k-1}^{-1} P_k$, anchored relative poses are friendlier to the normalization range, since the jitter between consecutive frames naturally cancels out.

For the gripper widths, we use the metric-scale absolute width between the gripper tips, which tells the model how wide the gripper is actually opened. Unlike the dummy camera pose $A_0 = I$, the conditioning gripper width helps the model infer the future gripper motion. We augment the full gripper width with a constant offset, $w_k \leftarrow w_k + \delta$, where $\delta$ is drawn once per gripper with a standard deviation of 3 mm and clipped at $\pm 9$ mm, to simulate inaccurate calibration and the gaps between data collection devices, so the model still infers relative gripper motion and overcomes the embodiment gaps.

### B.4 Training Configuration

Table 1 summarizes the training configuration of the ablation study of Sec. 4.1 and of the final CamUVA model of Sec. 4.2; the two share every setting except the training data, batch and learning rate schedule.

**Data filtering.** We exclude failed demonstrations and sources whose captions or camera poses failed the audit of Sec. 3.1, and hold out evenly spaced episodes of every source for validation. A chunk is rejected when the wrist does not move by 1 cm or $10^\circ$ within its first 16 steps and by 2 cm within 32 steps at DROID speed, since chunks that start idle teach the policy to stay still; DROID, whose idle frames are labeled per frame, instead masks their loss. Windows overlapping a pose spike, jump, timestamp gap, invalid quaternion or gripper glitch of an offline trajectory-quality scan (under 1% of the windows) and clips with a torn, foreign or frozen frame are dropped.

Table 1: Training configuration of the ablation runs and of the final CamUVA model. Learning rates, warm-ups and weight decay are given as video branch / action branch; the action branch trains from scratch, and the understanding branch, the video VAE and the text embedding are frozen in every run.

| Setting | Ablation Study | Final Model |
|---|---|---|
| Initialization | Cosmos3-Nano-Policy-DROID | same |
| Training data | DROID | CamX mix (Table 2) |
| Chunk | 33 frames, 1% multi-chunk | 33 frames, single chunk |
| Resolution | 480p and 256p mixed | same |
| Tokens per GPU | 65,536, sequence packed | same |
| Global batch (tokens / iter.) | 16.8M | 67M, 33.5M after 12.8k iter. |
| Iterations (evaluated) | 8k, every 1k evaluated | 14k |
| Compute (GB300 GPU-h) | 4k | 30k |
| Optimizer | AdamW, $\beta = (0.9, 0.95)$, clip 1.0 | same |
| Peak learning rate | $10^{-4}$ / $5 \times 10^{-4}$ | same |
| Weight decay | 0 / 0.05 | same |
| Warm-up (iterations) | 1k / 300 | 3k / 1k |
| Schedule after warm-up | constant | linear to 0.3 / 0.12 over 20k, then flat |
| UVA mode ratio **P**:**PP**:**VG**:**FD**:**ID** | 10:1:0.25:0.25:0.05 | same |
| Loss weights | 1 / 1 / 0.1 (video / traj. / gripper) | same |
| Evaluated weights | EMA | same |

**Data augmentation.** Each view passes through a random resized crop (scale 0.8–1.0), illuminant and gamma jitter, color jitter, sharpening, Gaussian blur and motion blur; the crop and the illuminant are shared by all views of a sample, the other operations are drawn per camera, and blur strengths scale with the frame width. A 480p source is emitted at 480p or 256p with equal probability (sources with a head camera add head at 480p with wrists at 256p), and with probability 0.3 at another aspect ratio by center crop or side padding, which weakens the correlation between the token grid and the embodiment. Sources deployed with a subset of their recorded views draw that subset in training, e.g., DROID's wrist stereo pair alone or GenRobot's main camera versus its stereo pair.

**Action normalization.** Each camera trajectory is 33 relative poses $P_0^{-1} P_k$, written as the translation and the first two rotation columns, plus a gripper width; row 0 is the identity conditioning row and rows 1–32 are predicted. Instead of statistics fitted to each dataset, every channel is mapped by a fixed linear map whose raw window shrinks toward the identity with the step index: translation $\pm 0.1$ m at step 1 widening linearly to $\pm 1$ m at step 16, rotation from 30% of its window to the full window at step 8, with values clipped at twice the window. Since the loss is a plain mean squared error in normalized space, equal physical translation errors before clipping receive about 100 times more squared-error weight at the first step than at the last; this normalization raised the RoboLab success rate of DROID-only runs by 4 points. At deployment the map is inverted, the rotation columns are re-orthogonalized by Gram–Schmidt, and the translations are smoothed by a Savitzky–Golay filter (window 7, order 2).

**Speed normalization and FPS modulation.** The sources record at 10–60 fps and their operators move at different speeds, so a fixed frame stride would make a 33-step chunk cover very different amounts of motion. For every episode we measure the 90th percentile of the gripper-tip speed and divide it by the DROID median of 0.245 m/s to obtain a normalized speed $s$; the frame stride is then chosen so that the sampled rate is about $15\,\mathrm{Hz} \times s$, so that one step covers on average DROID's 1.6 cm of tip travel, and the rate is jittered by up to $1.25\times$ during training. The resulting frame rate $f$ enters the network only through the temporal MRoPE index: video latent $m$ sits at $t = 24m/f$ and action step $i$ at $t = 6i/f$. At deployment the executor sets the robot's speed through its playback rate and reports the speed-normalized frame rate instead of its camera frame rate.

**Checkpoint selection and variance.** Every run of the design study in Sec. 4.1 costs 4k GB300 GPU-hours or more, and the study comprises more than twenty such runs, so repeating every arm with three seeds was beyond our compute budget. Instead, we evaluate every saved checkpoint of every run on the full RoboLab benchmark (120 tasks, 10 episodes each) and report the mean and standard deviation of the three best checkpoints of that run; averaging three checkpoints damps checkpoint-to-checkpoint fluctuation, and the same rule applies to every arm. The reported standard deviation thus measures the variation between checkpoints of one run, not across seeds, and the selection uses the benchmark we report, so the numbers are best-checkpoint estimates that are

comparable across arms. The *Difficulty-Aware* arm of Fig. 7, trained with two seeds, gives best-three means of 48.7 and 46.1 in **P** mode and 49.5 and 50.5 in **PP** mode (the figure reports their mean). The 0.5-point spread of the final checkpoint in Fig. 9 is the variation between three runs of one rollout configuration (Appendix D.1). Gaps of a few points between arms should be read with this seed spread in mind; the conclusions of Sec. 4.1 rest on larger gaps, such as the 8.1-point drop of joint actions (Fig. 5), the 22- and 19-point drops without pose encoding (Fig. 6), and the 15-point loss of **P** success under *Equal Probability* sampling.

# C CamX Dataset Details

## C.1 End-Effector and Camera Frame Conventions

Fig. 12 shows why CamX identifies every actuator group by its camera rather than by its end-effector: the five datasets place five different end-effector frames on their grippers, whereas the optical frame of the wrist camera follows one standard convention.

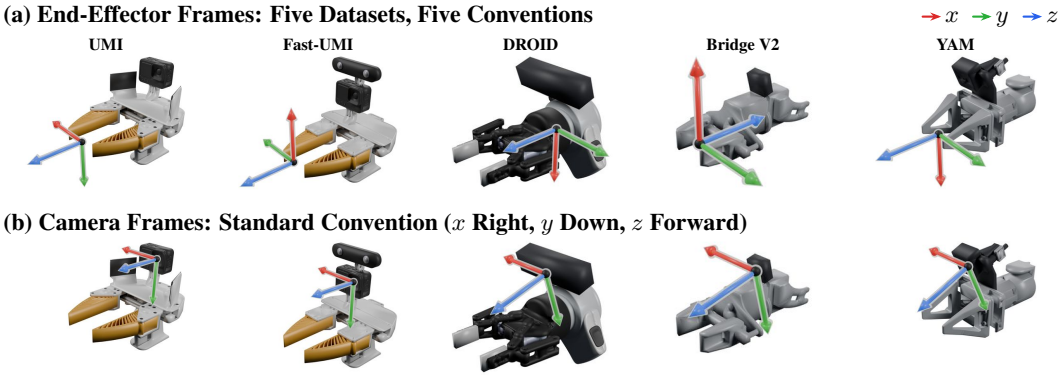



Figure 12: **End-effector and camera frames.** (a) Five datasets, five end-effector frames. (b) The wrist camera's optical frame, the standard convention (OpenCV[2], MATLAB[3]) and the frame of every CamX pose.

## C.2 CamX Sources

CamX holds 70 converted sources across 24 embodiment groups, 25.9k hours in total. Table 2 lists the 37 sources (15 embodiments, 18.6k hours, 1.74M episodes) that form the final model's training mix, the numbers quoted in the main paper, with their sample ratios; Table 3 lists the remaining 33 sources (7.2k hours) that are converted but outside that mix. Counts describe the imported or inventoried subsets, rather than necessarily matching the full releases reported in the cited papers. Some entries use release-derived estimates, so the aggregate inventory is approximate. The UMI group holds the datasets recorded with UMI grippers and their research variants; hand-held grippers of other designs are grouped under Other research projects, and the large-scale commercial datasets TacUMI, HiFi-UMI, GenRobot and DataClaw form separate groups. The token share is the fraction of the final model's training tokens drawn from each source (Sec. 4.2); it is assigned per source and task rather than in proportion to hours. We balance the token shares such that the model absorbs diverse embodiment configuration, rather than being dominated by the embodiment that has the most data. The sources of Table 3 are outside the training mix for one of four reasons:

- [A] they were not included when the training mix was frozen;
- [B] they are held out because their camera calibration is not fully validated;
- [C] multiple actuator groups do not share the global coordinate frame: the bimanual DataClaw exports, Vista-UMI and the bimanual FastUMI-100K release track each hand in its own frame;
- [D] dexterous-hand datasets recorded with a wrist camera; the current model has no dexterous-hand action head, so they are not in the training recipe yet and are listed for future training.

[2]OpenCV: `https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html`
[3]MATLAB: `https://mathworks.com/help/vision/ug/camera-calibration.html`

Table 2: **CamX training mix.** One row per embodiment and source in the final model's training mix, grouped by form factor; a release recorded on several robots appears under each. Datasets are task, scene, lab or station collections; counts come from converted data or the inventoried source releases. Hours are frames divided by the source frame rate, and views are camera streams. Release-derived estimates are included in the totals. Ratio is the share of the final model's training tokens drawn from the source. The converted sources outside the mix are in Table 3.

| Embodiment / source | Datasets | Episodes | Hours | Views | Ratio (%) |
|---|---|---|---|---|---|
| HAND-HELD GRIPPERS (5 embodiments) | *749* | *1.27M* | *10.9k* | | *23.38* |
| **UMI** | **85** | **93.5k** | **418** | | **6.02** |
| Data Scaling Laws (Lin et al., 2025) | 4 | 17.1k | 19.0 | 1 | 1.94 |
| exUMI (Xu et al., 2025b) | 8 | 1.49k | 3.0 | 1 | 0.14 |
| FastUMI (Zhaxizhuoma et al., 2025; Liu et al., 2025b) | 34 | 48.7k | 209 | 1 | 1.64 |
| ManiWAV (Liu et al., 2025d) | 5 | 1.13k | 2.7 | 1 | 0.34 |
| MV-UMI (Rayyan et al., 2025) | 2 | 652 | 3.2 | 2 | 0.04 |
| Touch in the Wild (Zhu et al., 2025b) | 10 | 1.47k | 8.6 | 1 | 0.43 |
| UMI (Chi et al., 2024) | 4 | 2.49k | 9.1 | 1–2 | 0.40 |
| UMI-Benchmark (Jin et al., 2026) | 10 | 19.6k | 159 | 1–2 | 1.00 |
| UMI-on-Legs (Ha et al., 2025) | 2 | 179 | 0.9 | 1 | 0.05 |
| ViTaMIn (Liu et al., 2025a) | 6 | 747 | 3.3 | 1 | 0.04 |
| **iPhUMI** | **24** | **6.43k** | **16.4** | | **2.85** |
| Behavior Prompting (Patel et al., 2026) | 6 | 1.56k | 3.3 | 4 | 1.00 |
| Gated Memory Policy (Gao et al., 2026) | 3 | 2.10k | 6.7 | 2 | 0.20 |
| HoMMI (Xu et al., 2026) | 3 | 437 | 2.0 | 6 | 0.65 |
| MoFPO (Wang et al., 2026) | 2 | 387 | 0.8 | 6 | 0.40 |
| UMIFT (Choi et al., 2026) | 10 | 1.95k | 3.6 | 2 | 0.60 |
| **Other research projects** | **8** | **5.62k** | **13.7** | | **0.30** |
| Dobb-E (Homes of New York) (Shafiullah et al., 2023) | 8 | 5.62k | 13.7 | 1 | 0.30 |
| **GenRobot** | **52** | **679k** | **8.31k** | | **9.21** |
| GenRobot 10Kh (GenRobot, 2025) | 52 | 679k | 8.31k | 2–3 | 9.21 |
| **HiFi-UMI** | **580** | **482k** | **2.14k** | | **5.00** |
| HiFi-UMI-2K (Simple AI, 2026) | 580 | 482k | 2.14k | 4 | 5.00 |
| SINGLE-ARM ROBOTS (1 embodiment) | *33* | *91.2k* | *491* | | *22.10* |
| **Franka** | **33** | **91.2k** | **491** | | **22.10** |
| DROID (Khazatsky et al., 2024) | 33 | 91.2k | 491 | 3–4 | 22.10 |
| BIMANUAL ROBOTS (6 embodiments) | *878* | *264k* | *5.03k* | | *39.51* |
| **YAM** | **318** | **157k** | **4.04k** | | **29.95** |
| ABC (Allshire et al., 2026) | 286 | 125k | 3.33k | 3–4 | 24.95 |
| MolmoAct 2 (bimanual YAM) (Fang et al., 2026) | 32 | 32.2k | 704 | 3 | 5.00 |
| **ARX5** | **478** | **57.7k** | **423** | | **6.52** |
| RDT (Liu et al., 2025c) | 303 | 6.11k | 29.7 | 3 | 1.03 |
| RoboCOIN (Wu et al., 2025b) | 105 | 29.7k | 234 | 3 | 2.46 |
| RoboDojo (Chen et al., 2026b) | 6 | 600 | 6.2 | 3 | 0.11 |
| RoboMIND (Wu et al., 2025a) | 64 | 21.3k | 153 | 3 | 2.92 |
| **RealSource RS-02** | **41** | **26.7k** | **346** | | **1.00** |
| RealSource-World (RealSource, 2025) | 41 | 26.7k | 346 | 3 | 1.00 |
| **OpenArm** | **3** | **6.98k** | **162** | | **0.50** |
| OpenArm (Enactic, Inc., n.d.) | 3 | 6.98k | 162 | 3 | 0.50 |
| **ALOHA** | **25** | **13.3k** | **22.8** | | **0.85** |
| ALOHA (Zhao et al., 2023a) | 14 | 815 | 2.8 | 3 | 0.10 |
| BiPlay (Dasari et al., 2025) | 6 | 12.3k | 18.5 | 4 | 0.70 |
| Cosmos Policy (Kim et al., 2026) | 5 | 185 | 1.5 | 3 | 0.05 |
| **AgileX PiPER** | **13** | **2.64k** | **34.9** | | **0.69** |
| RoboDojo (Chen et al., 2026b) | 12 | 1.20k | 9.5 | 3 | 0.19 |
| XVLA-Soft-Fold (Zheng et al., 2026) | 1 | 1.44k | 25.4 | 3 | 0.50 |
| MOBILE MANIPULATORS (3 embodiments) | *338* | *117k* | *2.22k* | | *15.01* |
| **AgiBot** | **108** | **89.3k** | **1.66k** | | **10.50** |
| AgiBot World 2026 (AgiBot World Team, 2026) | 37 | 27.5k | 575 | 3 | 5.00 |
| AgiBot World Beta (Bu et al., 2025a) | 71 | 61.8k | 1.08k | 3 | 5.50 |
| **Galaxea R1-Lite** | **228** | **27.2k** | **564** | | **4.11** |
| Galaxea Open-World (Jiang et al., 2025) | 171 | 22.6k | 541 | 3 | 3.56 |
| RoboCOIN (Wu et al., 2025b) | 57 | 4.59k | 23.5 | 4 | 0.55 |
| **Rainbow RB-Y1m** | **2** | **200** | **2.4** | | **0.40** |
| ModPack (Citron et al., 2026) | 2 | 200 | 2.4 | 4 | 0.40 |
| **Total** (15 embodiments, 37 sources) | **1,998** | **1.74M** | **18.6k** | | **100.00** |

Table 3: **CamX sources outside the training mix.** Converted sources that the final model did not train on, in the layout of Table 2; the bracketed letter gives the reason (notes in Sec. C.2).

| Embodiment / source | Datasets | Episodes | Hours | Views | Note |
|---|---|---|---|---|---|
| HAND-HELD GRIPPERS (6 embodiments) | *1,002* | *472k* | *4.49k* | | |
| **UMI** | **61** | **62.3k** | **331** | | |
| FastUMI-100K (dual-arm) (Liu et al., 2025b) | 19 | 50.8k | 262 | 2 | [C] |
| HuMI (Nai et al., 2026) | 5 | 827 | 2.7 | 2 | [A] |
| UMI-3D (Wang, 2026) | 3 | 4.60k | 10.5 | 1 | [A] |
| Vista-UMI (Yang et al., 2026a) | 30 | 5.00k | 53.9 | 2 | [C] |
| ViTaMIn-B (Li et al., 2025a) | 4 | 1.08k | 2.2 | 2 | [A] |
| **iPhUMI** | **3** | **159** | **0.4** | | |
| MuSe (Clark et al., 2026) | 3 | 159 | 0.4 | 2 | [A] |
| **Other research projects** | **54** | **9.14k** | **22.9** | | |
| AetheRock (Li et al., 2026a) | 11 | 3.76k | 8.5 | 1 | [A] |
| FreeTacMan (Wu et al., 2026b) | 30 | 3.68k | 8.8 | 1 | [A] |
| LEGATO (Seo et al., 2025) | 3 | 450 | 2.7 | 2 | [A] |
| ManipForce (Lee et al., 2026a) | 6 | 597 | 1.1 | 2 | [A] |
| TAMEn (Wu et al., 2026a) | 4 | 652 | 1.8 | 2 | [A] |
| **TacUMI** | **276** | **124k** | **2.99k** | | |
| OpenNeoData (NeoteAI Team & Fudan TEAI Team, 2026) | 276 | 124k | 2.99k | 1–2 | [A] |
| **DataClaw** | **582** | **264k** | **1.12k** | | |
| DM-DataClaw (Daimon Robotics, 2026) | 582 | 264k | 1.12k | 1–2 | [C] |
| **Dexterous hands** | **26** | **11.4k** | **36.3** | | |
| DexUMI (Xu et al., 2025a) | 16 | 1.85k | 3.9 | 1 | [D] |
| DexWild (Tao et al., 2025) | 10 | 9.51k | 32.4 | 3–5 | [D] |
| SINGLE-ARM ROBOTS (6 embodiments) | *1,021* | *165k* | *1.71k* | | |
| **UR5** | **397** | **82.1k** | **1.05k** | | |
| OpenNeoData (NeoteAI Team & Fudan TEAI Team, 2026) | 91 | 32.9k | 688 | 2 | [A] |
| RH20T (Fang et al., 2024) | 163 | 1.30k | 20.9 | 4–8 | [A] |
| RoboMIND 2.0 UR (Hou et al., 2025) | 143 | 48.0k | 342 | 6 | [B] |
| **Flexiv Rizon 4** | **308** | **26.1k** | **483** | | |
| OpenNeoData (NeoteAI Team & Fudan TEAI Team, 2026) | 75 | 20.8k | 410 | 1 | [A] |
| RH20T (Fang et al., 2024) | 233 | 5.32k | 73.1 | 6–9 | [A] |
| **Franka** | **134** | **26.2k** | **94.2** | | |
| FMB (Luo et al., 2025) | 5 | 8.39k | 31.3 | 4 | [A] |
| MolmoAct (Franka) (Lee et al., 2026b) | 2 | 7.90k | 30.9 | 3 | [A] |
| RH20T (Fang et al., 2024) | 97 | 911 | 11.1 | 6–8 | [A] |
| RoboSet (Bharadhwaj et al., 2024) | 30 | 9.00k | 20.9 | 1–4 | [A] |
| **KUKA LBR iiwa** | **176** | **2.23k** | **28.8** | | |
| RH20T (Fang et al., 2024) | 176 | 2.23k | 28.8 | 7–9 | [A] |
| **WidowX 250S** | **1** | **26.0k** | **51.5** | | |
| Bridge V2 (Walke et al., 2023) | 1 | 26.0k | 51.5 | 4 | [A] |
| **xArm6** | **5** | **2.54k** | **3.5** | | |
| Evo-1 xArm6 (Lin et al., 2026) | 5 | 2.54k | 3.5 | 2 | [A] |
| BIMANUAL ROBOTS (5 embodiments) | *448* | *75.9k* | *1.01k* | | |
| **ARX5** | **139** | **37.0k** | **732** | | |
| OpenNeoData (NeoteAI Team & Fudan TEAI Team, 2026) | 139 | 37.0k | 732 | 2–3 | [A] |
| **AgileX PiPER** | **18** | **12.2k** | **157** | | |
| OpenNeoData (NeoteAI Team & Fudan TEAI Team, 2026) | 18 | 12.2k | 157 | 3 | [A] |
| **ALOHA** | **122** | **10.9k** | **52.4** | | |
| AIST Bimanual Manipulation (Motoda et al., 2025) | 116 | 10.6k | 49.8 | 4 | [B] |
| Mobile ALOHA (Fu et al., 2025) | 6 | 278 | 2.6 | 3 | [A] |
| **OpenArm** | **27** | **3.66k** | **30.6** | | |
| OpenArm community (Enactic, Inc., n.d.) | 27 | 3.66k | 30.6 | 2–4 | [A] |
| **AIRBOT MMK2 + XHAND** | **142** | **12.2k** | **40.5** | | |
| Dexora (Zhang et al., 2026c) | 142 | 12.2k | 40.5 | 4 | [D] |
| **Total** (17 embodiments, 33 sources) | **2,471** | **713k** | **7.22k** | | |

### C.3 CAMERA POSE VALIDATION

We render the gripper from its recorded pose into the camera image. If it does not line up with the gripper in the frame, we inspect the pose, calibration, gripper model, or video decoding. We also compare image motion with the stored camera trajectory to catch timing and camera-assignment errors.

**Gripper projection into image plane.** In the OpenNeo UR split, the projected end-effector pose missed the gripper in the image, which led us to check forward kinematics from the measured joints. It disagreed with the stored poses, so we rebuilt the pose stream from the joints. In ABC-130K, CAD overlays identified each episode's gripper variant, letting us assign the matching wrist-camera extrinsics. Some DROID episodes contain only raw ZED SVO files. During decoding, the ZED's IMU compensation can flip the video 180°, which we corrected.

**Comparison of image motion with stored camera trajectory.** An overlay can look right even when trajectory timing or the world frame is wrong. In RoboCOIN, comparing image motion with the stored trajectory exposed a nine-frame (300 ms) state lead and left-right wrist-video swaps in eight datasets; we corrected both.

# D RoboLab Rollout Details

## D.1 Rollout Configurations

Every RoboLab number in this paper comes from a full-benchmark run: 120 tasks, 10 episodes each, RoboLab's default instruction wording and its own success detector. RoboLab leaves several rollout choices to the policy client, which we swept for every model; Table 4 lists the reported settings and the runs behind them follow. Figure 9 reports the mean and $\pm 1$ standard deviation over three runs for Cosmos3-Nano-Policy-DROID, $\pi_{0.5}$-DROID, and CamUVA. Cloak and LAP-3B each have one run, with a 95% $\mathrm{Beta}(k+1, n-k+1)$ credible interval for $k$ successes in $n$ episodes. The choices are the gripper command (the predicted width, or binarized at 0.5 as the official DROID clients do), the over-shoulder camera poses (RoboLab's hand-placed defaults or the mean DROID poses), the sampling steps and guidance, and two treatments that bring the rendered frames closer to DROID recordings: AV1 re-encoding at constant quality 36, the compression of the training videos, and an auto-exposure and white-balance adjustment fitted to DROID.

Table 4: Final RoboLab rollout configuration of every model: action interface, chunk length / executed steps, gripper command, over-shoulder camera poses, frame treatment, sampling steps and guidance weight; LAP-3B and CamUVA predict poses that our client converts to joint targets by IK.

| Model | Actions | Chunk | Gripper | Exteriors | Frames | Steps / guidance |
|---|---|---|---|---|---|---|
| Cosmos3-Nano-DROID | joint targets | 32 / 32 | continuous | hand-placed | clean | 4 / 3.0 |
| $\pi_{0.5}$-DROID | joint targets | 15 / 15 | continuous | dataset mean | AV1, exposure | 10 / – |
| Cloak | joint targets | 16 / 8 | continuous | dataset mean | AV1, exposure | 10 / – |
| LAP-3B | pose deltas | 16 / 8 | binary | hand-placed | clean | 10 / – |
| CamUVA (**P**) | camera poses | 32 / 32 | continuous | dataset mean | AV1 | 4 / 2.0 |

**CamUVA.** Our server runs in **P** mode with four sampling steps, a 32-step chunk of camera-relative poses executed in full through absolute end-effector IK, a continuous gripper, both eyes of the stereo wrist camera and the two exteriors at $256 \times 448$, and AV1 re-encoding at quality 36 because the model was trained on such video. The three reported runs score 47.2 / 47.8 / 48.1% with guidance 2.0. The design study of Sec. 4.1 evaluates every checkpoint at guidance 3.0.

**Cosmos3-Nano-Policy-DROID.** The published weights run in our serving stack in their only mode, **PP**, with four UniPC steps, guidance 3.0 and a 32-step joint-position chunk executed in full. The official preset (binary gripper, hand-placed exteriors, clean frames) scores 37.8–39.5% over five runs, above the published 36.8%. The continuous gripper raises it to 48.8%; dataset-mean exteriors lower it (34.1–34.7% binary, 44.7% continuous); guidance 2.5 / 3.5 / 4.0 give 44.8 / 47.6 / 47.0%; ten sampling steps and the two frame treatments move it by less than 2 points. We therefore keep the official preset with a continuous gripper (48.8, 48.2 and 47.1% in the three reported runs).

**$\pi_{0.5}$-DROID.** The openpi `pi05_droid` checkpoint runs behind openpi's own server, and a bridge reproduces the RoboLab reference client: resize-with-pad to 224 px, joint-position actions, the 15-step chunk executed in full, openpi's default 10 flow steps. The reference configuration scores 28.8% (published: 28.0%). The continuous gripper and the dataset-mean exteriors give 30.9 and 29.0% alone and 32.5% together; AV1 re-encoding at quality 30 / 36 / 42 adds nothing (32.8 / 32.3 / 33.2%); the exposure adjustment on top of quality 36 gives 36.0%, the best row, and 32.6 and 34.8% in its two repeats.

**Cloak.** The authors' server, a $\pi_{0.5}$ DROID fine-tune that masks the gripper out of the wrist view with a rendered silhouette, receives our wrist-camera extrinsics and runs through the same bridge: absolute joint targets, 8 of 16 chunk steps executed, continuous gripper as in its own client, 10 flow steps. Its own client configuration scores 13.0%, and $\pi_{0.5}$'s best configuration (dataset-mean exteriors, both frame treatments) 15.1%, which we report.

**LAP-3B.** The authors' server predicts 16 flange-pose deltas in the robot base frame; our bridge computes its proprioceptive state by forward kinematics, turns the deltas into joint targets by damped least-squares IK, executes 8 of 16 steps and binarizes the gripper at 0.5 as its reference client does. The reference configuration scores 10.7% and $\pi_{0.5}$'s best configuration 10.3%. On a 12-task subset (120 episodes, 29 successes in the reference cell) we then swept bridge-side fixes: a two-step lead of the executed chunk over the measured joints (33), position-priority IK (24), feeding the commanded rather than the measured gripper state back to the policy (34), and the three together with an open threshold of 0.15 on that state (42). On the full benchmark these fixes score 15.5%, the reported number.

# E Real-World Experiment Details

## E.1 In-Distribution Experimental Setup

We evaluate four tasks on their training embodiment, following the protocol of the corresponding baseline. Figs. 13 and 14 show the UMI tasks' initial wrist-camera views in evaluation order. Success criteria and outcomes follow.

**`Cup Arrangement` and `Towel Folding` (UMI).** Both tasks use one ARX X5 arm with a UMI gripper and fisheye wrist camera. Following UVA's multi-task evaluation (Li et al., 2025c), one checkpoint handles both tasks; test states vary object placement, distractors or unseen objects, backgrounds, and gripper color. Baseline rates are from UVA's original evaluation set. The instructions ask the robot to orient and place a cup on a saucer, and to fold a towel in half toward the right. `Cup Arrangement` covers 20 states (Fig. 13) and succeeds when the cup stands on the saucer with its handle pointing left. CamUVA succeeds in 17 of 20 trials (85%): it leaves the cup unplaced once and knocks it over twice. `Towel Folding` covers 20 states (Fig. 14) and succeeds when the towel is folded with its left half on top. CamUVA succeeds in 18 of 20 trials (90%).

**`Dish in Rack` (bimanual YAM).** Two YAM arms with RealSense D405 wrist and head cameras receive "Load the plates into the dish rack." The white table, wire rack, and two plates (drawn from five colors) are shared with the ABC-VLA baseline and unseen by both models. Success requires both plates upright in the rack. CamUVA succeeds in 9 of 10 trials (90%); in the failure it leaves the first plate across the rack and does not pick up the second. ABC-VLA succeeds in 1 of 10 (Appendix E.4.1).

**`Box Placement` (mobile RB-Y1).** This follows ModPack's `Box Transfer` protocol (Citron et al., 2026): from a marked starting point, the robot moves an open box from a cabinet to either of two empty rack shelves. Success requires the box to rest stably on the target shelf.

## E.2 Cross-Embodiment Experimental Setup

**Training coverage.** Of the five deployment embodiments in Fig. 10, UMI, iPhUMI, and GenRobot appear in CamX; the other two do not appear as complete setups (Table 2). The New Gripper uses an iPhone camera but has novel finger geometry. The D405 used to replace UMI's GoPro appears on other training embodiments, but never in combination with the UMI gripper fingers. The environment and distractor objects were unseen in the training data, showing that our model generalizes well to diverse environments.

**Initial states.** Each bar in Fig. 10 represents ten episodes; all embodiments see the same ten initial states per task. Figs. 15–18 show the UMI wrist-camera view for each state, in evaluation order. States vary background, object position, and object identity or pose. Times under the bars give the recorded duration of successful episodes, measured from first to last wrist-camera frame.

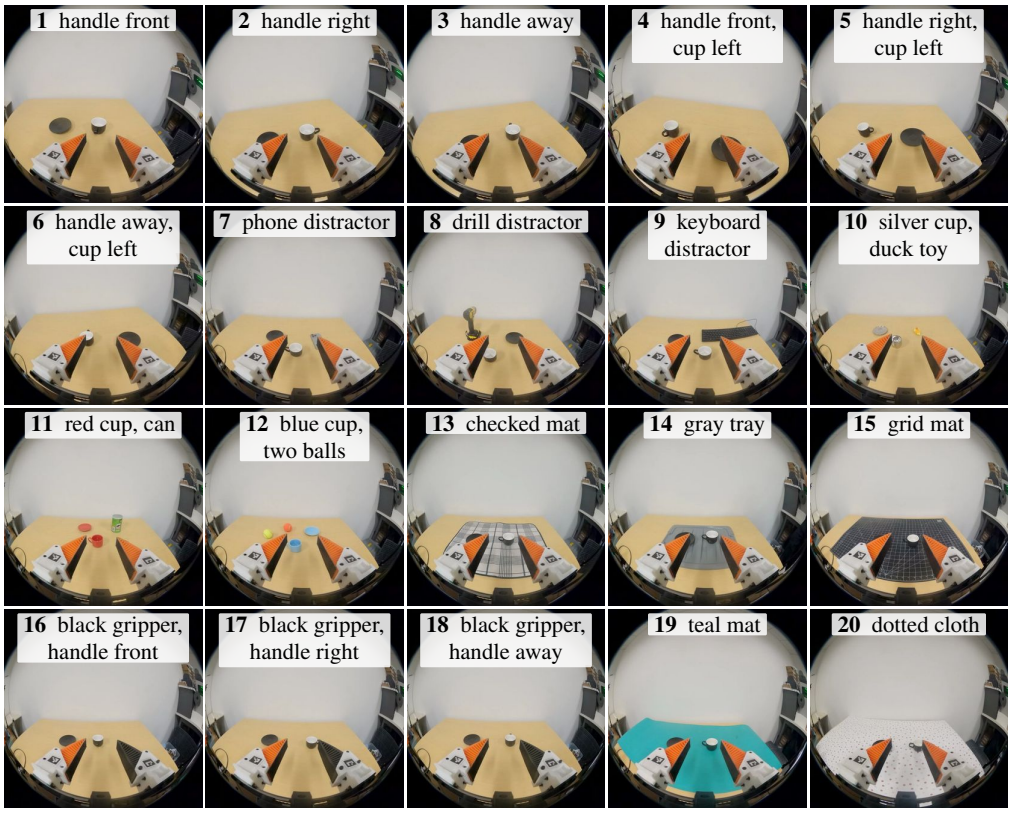



Figure 13: **`Cup Arrangement` initial states** (20 episodes, UMI gripper). In order: cup placement (1–6), distractors (7–9), cup and saucer variants (10–12), backgrounds (13–15, 19–20), and gripper color (16–18).

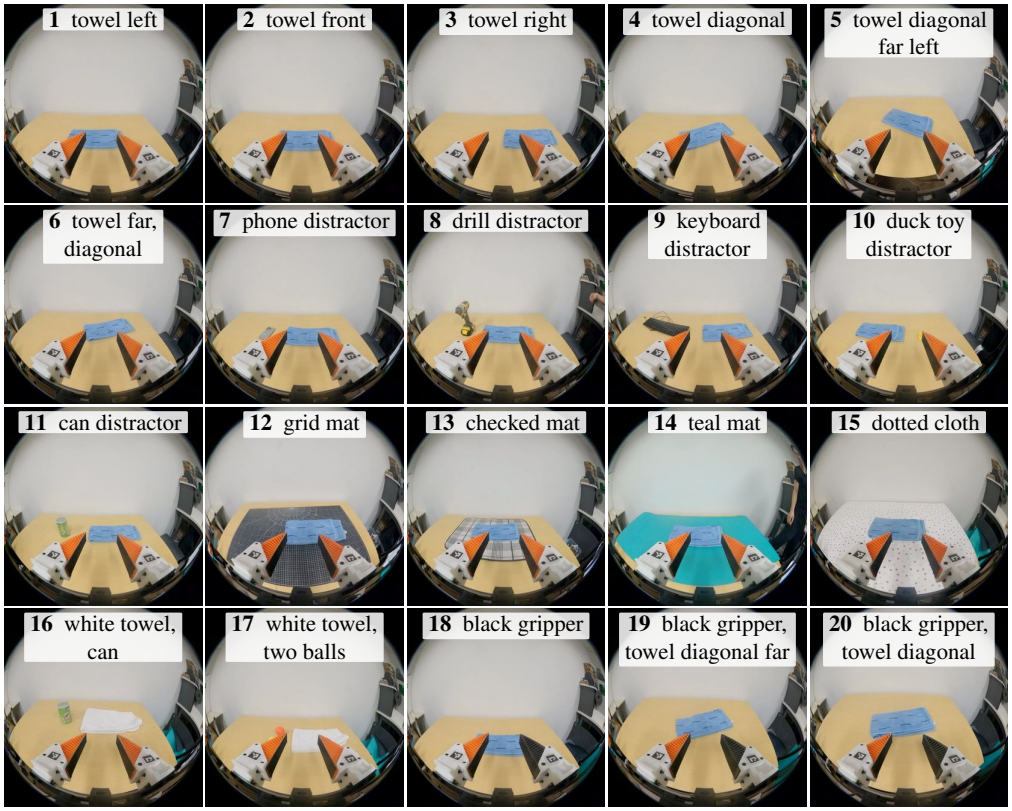



Figure 14: **`Towel Folding` initial states** (20 episodes, UMI gripper). In order: towel placement (1–6), distractors (7–11), backgrounds (12–15), towel variants (16–17), and gripper color (18–20).

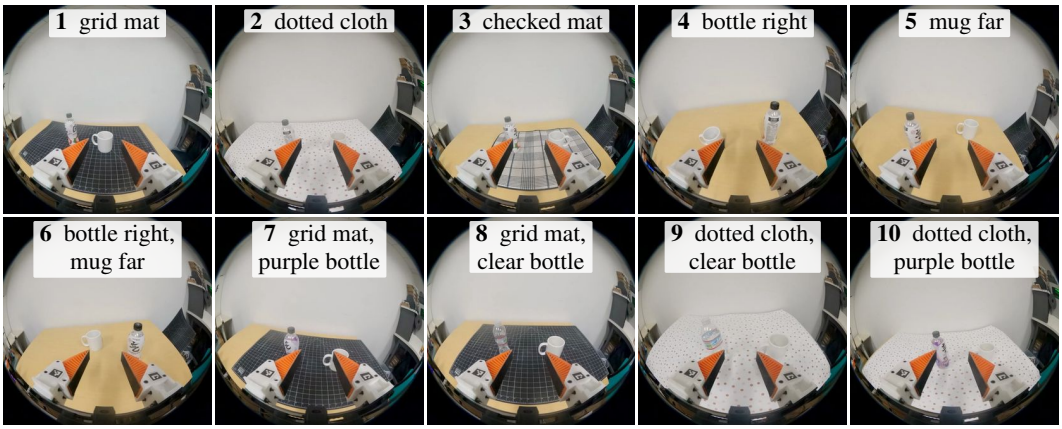



Figure 15: **`Water Pouring` initial states** (UMI gripper, ten states shared across embodiments). In order: backgrounds (1–3), bottle placement (4–6), and bottle variants (7–10).

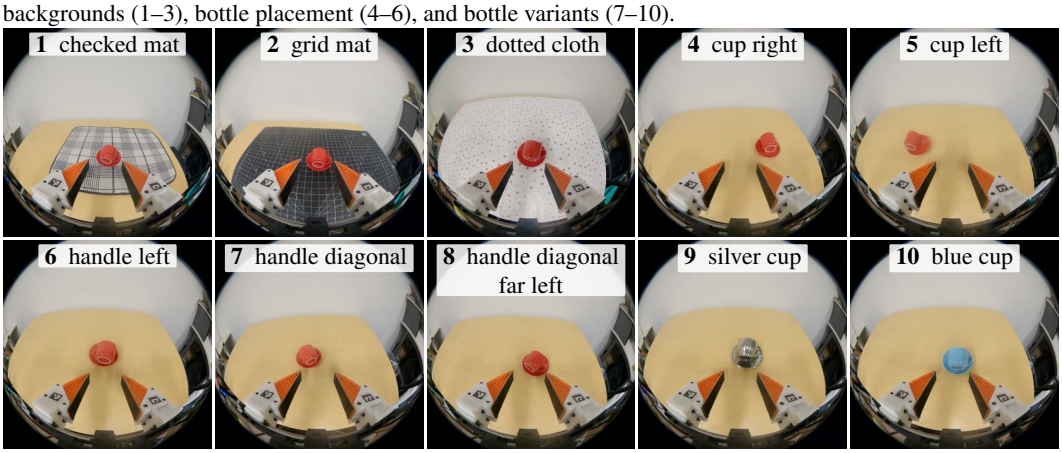



Figure 16: **`Cup Flipping` initial states** (UMI gripper, ten states shared across embodiments). In order: backgrounds with the cup upside down (1–3), cup pose (4–8), and cup variants (9–10).

### E.3 Mobile Manipulation Experimental Setup

We deploy the policy on a Rainbow Robotics RB-Y1[4], a wheeled bimanual manipulator with an omnidirectional mecanum base, a six-axis torso, two 7-DoF arms with grippers, and a pan-tilt head. The policy observes three cameras, one on each wrist and one on the left eye of the head's stereo pair, streamed at 15 fps, center-cropped to 4:3, and resized to $640 \times 480$ as in training. All whole-body quantities are expressed in the robot's base odometry frame; no external localization or base calibration is used.

The policy predicts motion in camera space, and a three-layer controller converts these predictions into joint-level commands. For each of the `left wrist`, `right wrist`, and `head` cameras, the policy outputs 32 future relative camera poses. The first layer re-anchors each trajectory to the camera pose measured at image-capture time in the base odometry frame, then applies fixed camera-to-effector extrinsics to obtain 6-DoF wrist pose trajectories and a head pose trajectory. The policy thus never reasons about the base or joint configuration.

The second layer converts these targets into arm, head, torso, and base joint targets with a damped weighted least-squares whole-body inverse-kinematics (IK) solver. Each wrist target drives its 7-DoF arm, with a posture regularizer toward the mean initial demonstration configuration resolving redundancy. The head target is realized jointly by the pan-tilt head and the base: its orientation drives the pan and tilt joints, while its translation induces planar base motion when the head alone cannot reach the target. The torso is fixed at the demonstrated posture by a hard zero-velocity constraint, and base height, roll, and pitch are pinned by a large weight, leaving planar $x$, $y$, and yaw as the base's free motion. A single predicted head trajectory therefore yields coordinated head-joint and base motion, while the wrist trajectories yield arm motion.

[4]RB-Y1 Documentation: `https://rainbowrobotics.github.io/rby1-dev/`

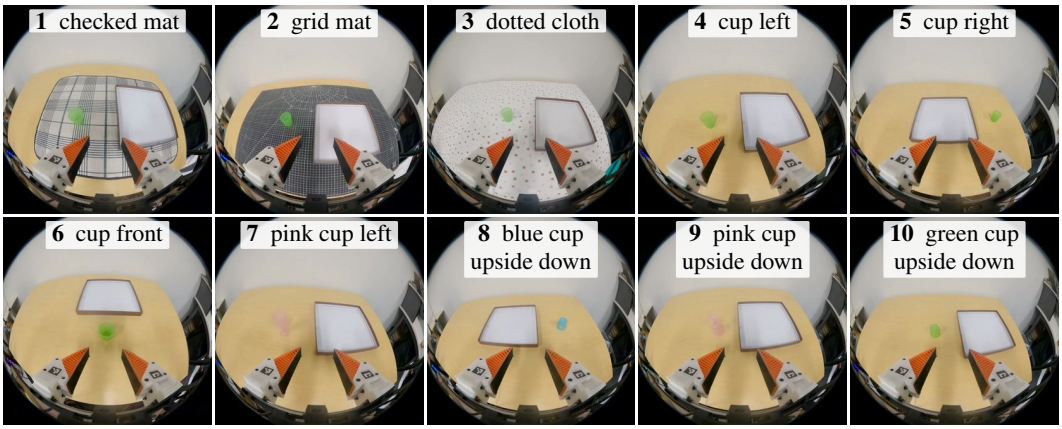



Figure 17: **`Cup in Box` initial states** (UMI gripper, ten states shared across embodiments). In order: backgrounds (1–3, cup upside down in 1–2), cup position and orientation (4–6), and cup variants (7–10); left/right are relative to the box.

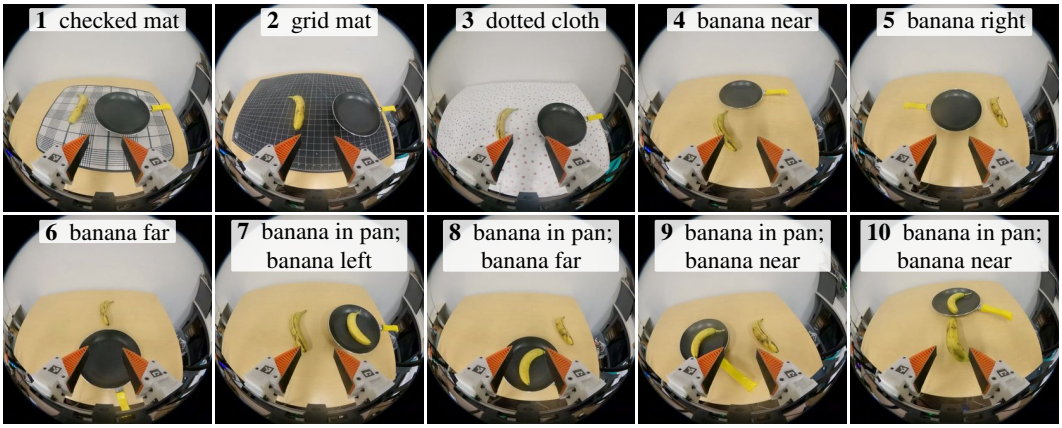



Figure 18: **`Banana in Pan` initial states** (UMI gripper, ten states shared across embodiments). In order: backgrounds (1–3), banana placement (4–6), and a second banana in the pan (7–10).

The third layer tracks the IK trajectories with a 100 Hz joint controller built on the released whole-body controller of HoMMI (Xu et al., 2026). To compensate for differing actuator delays, trajectories are sampled with a lookahead of 0.10 s for the arms, 0.22 s for the head, and 0.12 s for the base. The base is tracked separately: each cycle, we first solve the IK to convergence for the latency-compensated base target to obtain a goal base pose, which the base tracks with proportional feedback on the pose error plus the goal velocity as feed-forward. The arm and head targets are then solved using the average of the measured and goal base poses, decoupling base tracking error from wrist motion.

Joint and base commands are sent at 100 Hz. Each new policy chunk is cross-faded into the executing trajectory over 0.4 s to avoid discontinuities. Relative to the released HoMMI controller, our implementation replaces the base-control loop with the two-stage procedure above and raises the driver low-pass cutoff from 1 Hz to 10 Hz. Together, these layers map relative camera-pose predictions directly to executable whole-body motion while compensating for actuator latency.

### E.4 Failure Mode Analysis

#### E.4.1 Baseline Failure Modes

The ABC-VLA baseline succeeded in only one of ten rollout episodes (Sec. 4.2.1). Most failures, however, did not originate from grasping contact itself. Once the policy committed to a plate, the gripper usually closed firmly and lifted without slippage, and handover generally succeeded when the receiving wrist camera could see the incoming plate. Instead, failures concentrated during transport and placement, when the visual configuration differed from those represented in the baseline's

training data. We observed four recurring failure modes across the ten rollouts; Figs. 19 and 20 show the two most common.

**Loss of visual grounding during transport.** The policy often lifted the plate too high, until neither the table nor the dish rack remained visible in the wrist camera, and then remained in place until the rollout timed out. A similar failure occurred when one wrist camera was temporarily occluded. In two of the ten episodes, the plate briefly blocked one arm's wrist camera during a handover or while approaching the rack. Both arms then began moving upward toward the third-person camera and continued rising until we stopped the rollout for safety.




Figure 19: **Baseline transport-stage failure.** Head view over the left wrist view; time since the episode start. After the handover the left arm raises the plate instead of lowering it, until only the plate and the wall remain in the wrist image, and holds it there until timeout.

**Failure to release over the rack.** Another recurring failure occurred after the arm reached the rack but did not release the plate. This happened both after direct transport and after handover. The gripper hovered over the rack without opening, leaving the plate suspended and preventing the episode from completing or proceeding to the second plate. Some rollouts remained in this state for an extended period, substantially increasing their duration.

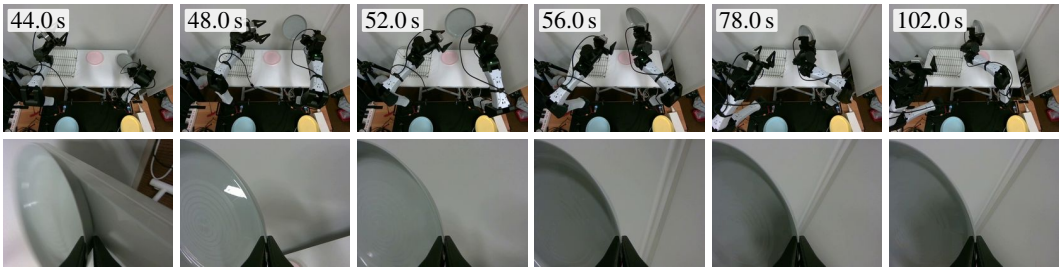



Figure 20: **Baseline placement-stage failure: refusal to release.** Head view over the right wrist view; time since the episode start. The right arm lifts the plate, carries it to the rack and hovers beside it with the gripper closed until the episode ends.

**Premature or misaligned release.** In other rollouts, the policy released the plate before reaching the rack or while still approaching it. The plate either fell beside the rack or was dropped during transport, after which the arms often stalled for up to two minutes. In one rare case, the arm returned to the fallen plate and attempted another placement, but then triggered the wrist-camera occlusion failure described above and failed the episode.

**Failure to re-engage after a partial success.** After placing the first plate, the arms often drifted into uncoordinated motion rather than returning to the table for the second plate. This was not a grasping limitation: the policy had successfully grasped a plate from the same table moments earlier. Instead, with one slot in the rack now occupied, the (out-of-distribution) scene differed from the configurations from which the policy had learned to initiate the next pick.

Overall, the failures concentrated in stages where the visual configuration changed and became out-of-distribution compared to ABC-VLA's training data during our rollout. Losing sight of the table or rack led to prolonged upward motion, while camera occlusion could trigger uncontrolled upward movement, and reaching the rack could result in either failure to release or premature release. Even

after the first successful placement, the policy often failed to re-engage with the remaining task. In contrast, basic grasping and handover were generally successful when the relevant visual context remained available.

### E.4.2 CamUVA Failure Modes

GenRobot fails all ten `Cup Flipping` episodes (Fig. 21) by overshooting the cup before closing its gripper. The demonstrations use iPhUMI, with a UMI gripper and a rectilinear iPhone camera; GenRobot differs in both gripper geometry and camera type, using a fisheye stereo camera. These differences may affect contact and spatial estimation. One possible source of the overshoot is depth estimation: the iPhone's main and ultrawide cameras used in training are closer together than GenRobot's left and right stereo cameras. The model may therefore overestimate the cup's distance with GenRobot and move the gripper too far forward.

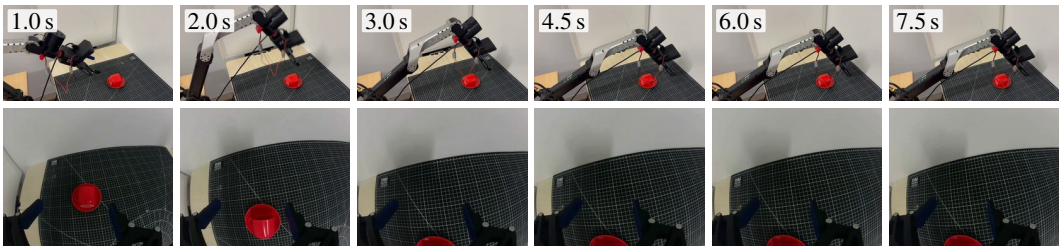



Figure 21: **GenRobot on `Cup Flipping`: no grasp in ten episodes.** Third-person view over the GenRobot wrist view, time since the episode start. The wrist pitches up on approach, the cup passes between the open fingers and sinks below the lower edge of the wrist image, and the arm hovers there without descending until the operator stops the episode.

## E.5 Emergent Recovery Behavior

Across our real-world cross-embodiment experiments, the policy exhibited failure recovery behaviors that were never explicitly demonstrated or supervised. These behaviors appeared consistently under unseen cameras, grippers, embodiments, objects, and environments. The tasks referenced below are `cup-in-box`, `banana-in-pan`, `water-pour`, and `cup-flip` on single-arm setups, and `dish-in-rack` on the bimanual YAM setup. We group the observed behaviors into five categories based on when the failure occurred and how the policy responded.

**Iterative grasp refinement at initial contact** (Fig. 22). The most common initial-contact failure was an insufficient grasp caused by the gripper not closing far enough, leaving the object roughly in place. The policy typically stayed at the object and refined the grasp in place, tightening the closure and adjusting the grasp orientation in small increments until the object could be lifted. The number of increments generally scaled with the initial grasp error. In `cup-in-box` with the UMI gripper, a single re-open and re-grasp was usually sufficient. With a previously unseen gripper of different finger length and shape, the initial grasp was almost always too loose across `cup-flip`, `cup-in-box`, `banana-in-pan`, and `water-pour`; the policy progressively increased closure across successive attempts, or through continuous re-grasp motions within a single attempt, until the object was held firmly enough to complete the task. In `cup-in-box` with the cup placed upside down, where the effective gripper width must be narrower and more precise, refinement also extended to orientation: the policy rotated the grasp while narrowing the gripper width across attempts, including smaller sub-attempts consisting of minor nudges and movements.

**Retract, re-observe, and reattempt** (Fig. 23). When a grasping failure left the object's location uncertain, the policy sometimes retracted the arm to recover a clear view before trying again. We observed this primarily in two situations. In `banana-in-pan`, the banana often slipped during the lift and fell to a location partly occluded by the gripper; the policy lifted and retracted the arm, waited until the banana was visible, and then re-approached and re-grasped it. When the gripper instead struck a cup or water bottle from the side and pushed it out of the immediate workspace, the policy similarly retracted to locate the object before reattempting the grasp. In other rollouts with

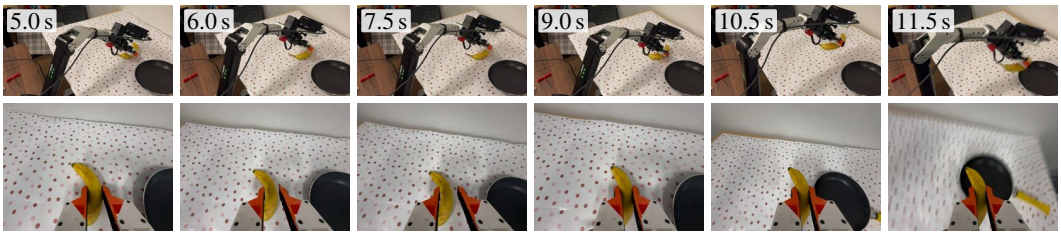



Figure 22: **Iterative grasp refinement at initial contact** (`Banana in Pan`, unseen gripper). Third-person view over wrist view, time since the episode start. The first closure is too loose and the banana slips out of the fingers during the lift. Instead of retracting, the policy returns to the same spot, closes the gripper further and lifts again; with this gripper the first grasp was almost always too loose, and the closure was tightened over one or more re-grasps until the object held.

the same sideways strike, however, the policy followed the displaced object directly, as described next.

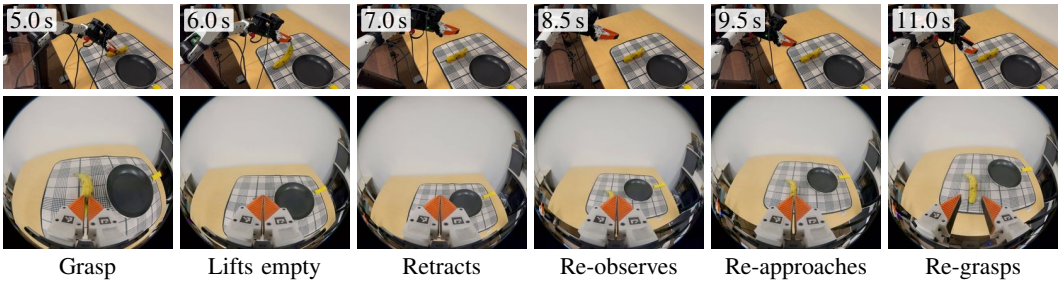



Figure 23: **Retract, re-observe and reattempt** (`Banana in Pan`, UMI). Third-person view over wrist view. The gripper closes on the banana and lifts empty, and the banana comes to rest partly hidden by the gripper. The arm pulls back and up until the banana and the pan are both in the wrist view again, then re-approaches and re-grasps it. The same retraction followed when a cup or bottle had been struck out of the immediate workspace.

**Tracking and re-grasping displaced objects** (Fig. 24). When the gripper struck an object and it visibly rolled or slid, the policy often followed it to its new location and pose rather than returning to the original grasp point or retracting. It moved in the direction of the displacement, searched for the object, and re-grasped it in its resulting pose. In `cup-in-box`, the gripper sometimes tilted the cup onto its side and caused it to roll away; the policy followed the roll and re-grasped the cup in its new pose, adapting its grasp orientation and gripper width over several attempts. In `water-pour` with the unseen gripper, a loose grasp similarly allowed the bottle to slip and roll sideways; the policy tracked the roll, adjusted its gripper width and grasp orientation, and completed the next attempt.

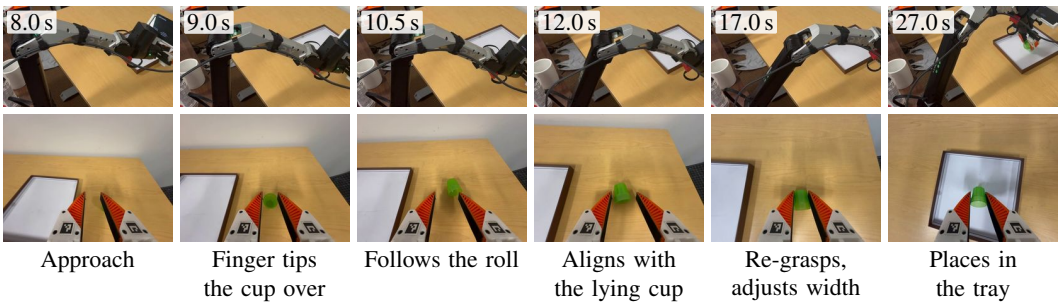



Figure 24: **Tracking and re-grasping a displaced object** (`Cup in Box`, iPhUMI). Third-person view over wrist view. A fingertip tips the cup over and it rolls onto its side. Rather than returning to the original grasp point, the policy follows the roll, aligns the jaws with the lying cup and re-grasps it while adjusting the gripper width to the new pose, then places it in the tray.

**Re-grasping after drops during transport** (Fig. 25). During object transport, slipping or insufficient gripper force sometimes caused a successfully lifted object to be dropped. The policy did not abandon the task and exhibited two distinct responses. First, it recovered directly at the landing site using the iterative refinement described above. In `banana-in-pan`, for example, after dropping the banana en route, the policy re-grasped it where it landed and, when necessary, first nudged it into an easier-to-grasp orientation. Second, during some pick-move-place tasks, the policy performed in-hand correction after re-grasping, slightly re-orienting the object during transport rather than committing to a single grasp until it reached the pan or box.

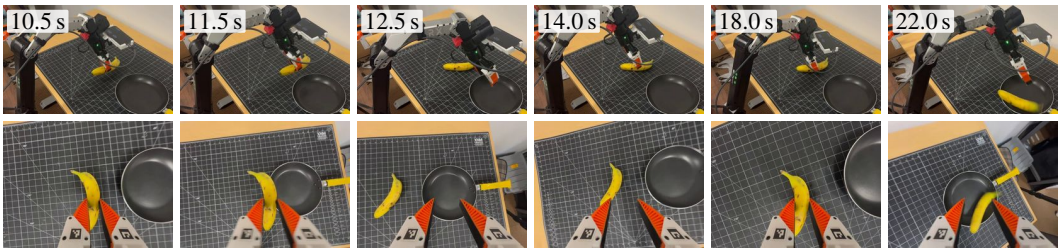



Figure 25: **Re-grasping after a drop during transport** (`Banana in Pan`, iPhUMI). Third-person view over wrist view. The banana slides out of the fingers on the way to the pan and lands beside it in a new pose. The arm turns back, re-grasps the banana where it landed and completes the placement; in other rollouts the policy first nudged an awkwardly landed object into an easier orientation before re-grasping.

**Recovery in multi-step and bimanual manipulation** (Figs. 26 and 27). In longer-horizon tasks, failures also occurred after grasp acquisition, during manipulation or transfer between arms. The policy chained the behaviors above into recovery loops spanning multiple stages of the task. In `cup-flip`, the robot must grasp a tilted or sideways cup, lower its wrist to flip the cup approximately flat, and release it onto the saucer. When the wrist did not lower sufficiently or the cup bounced after a high release, the policy re-grasped it with an adjusted orientation and repeated the flip and release until the cup sat flat. In the bimanual YAM `dish-in-rack` task, the unseen environment, different lighting, and slightly reflective tabletop led to shallow edge grasps, causing plates to be missed or dropped during transport. The policy applied the same refinement strategy, increasing grasp depth and aligning the gripper with the plate edge until the grasp was firm enough for handover. When plates were subsequently dropped during lifting or handover, the better-positioned arm re-grasped the plate and resumed the handover, allowing recovery to be shared across both arms rather than tied to the arm that dropped the object.

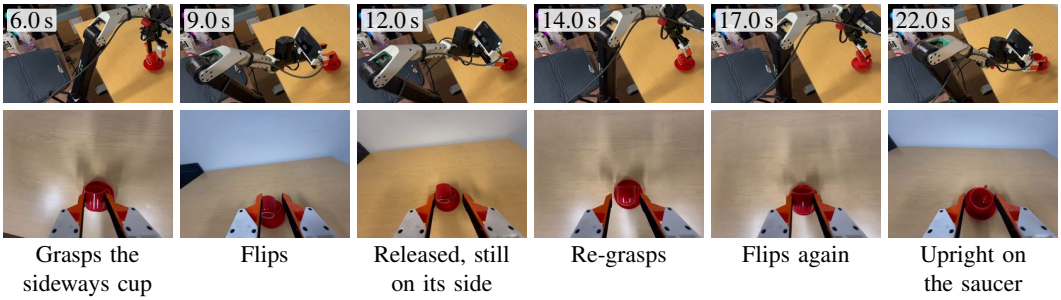



Figure 26: **Recovery in a multi-step task** (`Cup Flipping`, unseen gripper). Third-person view over wrist view. The first grasp, flip and release leaves the cup lying on its side on the saucer. The policy re-grasps the cup from above with an adjusted orientation, repeats the flip and release, and the cup comes to rest upright. Such loops over the whole grasp, flip and release sequence continued until the cup sat flat.

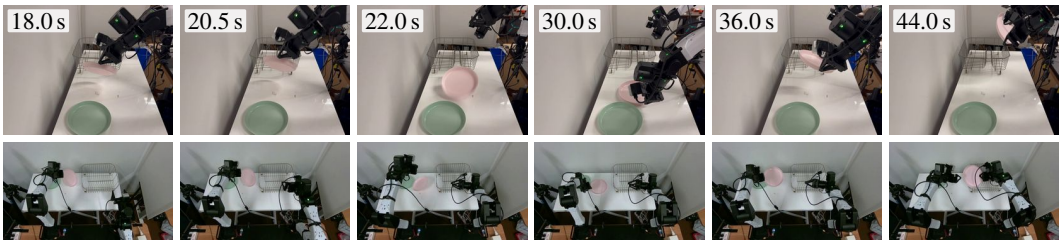



Figure 27: **Recovery in bimanual manipulation** (`Dish in Rack`, YAM). Third-person view over head view. The left arm lifts the plate and the right arm comes in for an in-air handover, during which the plate drops flat onto the table. The left arm re-pinches the plate where it landed, lifts it again and the second handover succeeds; whichever arm was better positioned took over such re-grasps.

Overall, the policy recovered from a wide range of failures under unseen or out-of-distribution cameras, grippers, embodiments, objects, and environments. Initial grasp failures were often handled by refining gripper width or orientation in place, retracting to re-observe the scene, or tracking displaced objects to their new locations. After drops during transport, the policy re-grasped and re-oriented the object before continuing. These behaviors also carried over to later stages of multi-step and bimanual tasks, including repeated flip attempts in `cup-flip` and grasp-depth and orientation correction before handover in `dish-in-rack`; in most cases, the policy ultimately completed the task.