# Can Robots Imagine Better Actions? World Action Models vs VLAs on Cross-Embodiment Transfer and Learning from Heterogeneous Data

Dohyung Kim, Eugene Seo, Rohan Walia, Eryk Halicki
ETH Zürich
3D Vision - 2026S
Supervised by: Julia Chen and Boyang Sun
{dohkim, eugseo, rwalia, ehalicki}@student.ethz.ch

## Abstract

*Imitation learning is severely bottlenecked by the high cost of collecting data across diverse environments and robot morphologies. World Action Models (WAMs) address this by predicting actions from generative world model latents. Pioneer frameworks like DreamZero claim WAMs fundamentally outperform traditional Vision-Language-Action models (VLAs) in cross-embodiment transfer and learning from heterogeneous data. In this work, we empirically evaluate these claims by benchmarking a representative WAM, MimicVideo, against state-of-the-art VLAs (SmolVLA and $\pi_{0.5}$). By systematically isolating morphological scaling from data composition across simulation and real-world tasks, we reveal a nuanced performance landscape. We find that while WAMs successfully decouple task understanding from kinematics, cross-embodiment transfer remains heavily constrained by the structural similarity between the source and target robots. Furthermore, although WAMs significantly improve zero-shot generalization in heterogeneous multi-task settings, they are not uniformly superior: they fail in homogeneous, contact-rich tasks where minor video prediction artifacts cause catastrophic execution failures. Ultimately, WAMs offer strong semantic compositionality but remain highly sensitive to visual domain shifts and latent prediction inaccuracies.*

## 1. Introduction

The pursuit of general-purpose robotic manipulation relies heavily on the ability of agents to learn from diverse human demonstrations. Imitation Learning allows robots to acquire complex, multi-step manipulation skills by mapping visual observations directly to low-level motor commands. While this end-to-end approach has yielded impressive results in tightly constrained laboratory environments, it faces a fundamental scalability bottleneck. Collecting high-quality, teleoperated demonstration data is exceptionally expensive and time-consuming. Furthermore, policies trained on specific robotic hardware or within specific laboratory settings often fail to generalize when deployed on a different robot arm or when tasked with manipulating novel objects under varying lighting conditions. This brittleness severely limits the deployment of intelligent robots in dynamic, real-world environments.

To address the limitations of standard imitation learning, the robotics community has increasingly turned to Vision-Language-Action models (VLAs). These architectures process natural language instructions alongside visual streams to output joint trajectories. However, standard VLAs still fundamentally rely on direct mappings to action spaces, which inherently ties their learned representations to the specific kinematics and dynamics of the training hardware. Recently, a paradigm shift has emerged with the introduction of World Action Models (WAMs). Instead of mapping pixels directly to motor torques, WAMs operate by predicting actions derived from the latent spaces of generative world models. By leveraging large-scale video prediction, these models can essentially "imagine" the visual consequences of an action sequence before executing it.

The pioneering WAM framework, DreamZero [26], posits two major claims regarding the superiority of World Action Models over standard VLAs. First, the authors claim that WAMs enable stronger cross-embodiment transfer. By leveraging a video model that inherently understands physical scene dynamics, including object motion and contact, a WAM should efficiently transfer to a novel robot with limited robot-specific fine-tuning. Second, the authors claim that WAMs are superior at learning from heterogeneous, multi-task data distributions. Standard VLAs often suffer from catastrophic forgetting or policy collapse when trained on highly diverse datasets. WAMs bypass these bottlenecks because the massive capacity of the video backbone robustly encodes broad physical knowledge, enabling suc-

cessful generalization across heterogeneous tasks.

In this paper, we systematically and empirically evaluate both of these foundational claims. We utilize the recently proposed WAM architecture, MimicVideo [13], and rigorously compare its performance against two leading VLAs: SmolVLA [23] and $\pi_{0.5}$ [9]. Through this extensive benchmarking, we aim to provide the community with a clear, unbiased understanding of the true capabilities and current limitations of World Action Models in modern robotics.

## 2. Related Work

### 2.1. Imitation Learning and VLAs

Behavioral Cloning (BC) [15] and Imitation Learning (IL) algorithms like DAgger [16] have historically formed the backbone of visuomotor policy training. Recent breakthroughs in action representation, such as Diffusion Policies [4] and Action Chunking with Transformers (ACT) [27], have significantly improved the precision and robustness of these policies in continuous, high-frequency control tasks.

The introduction of VLA models significantly advanced this paradigm by integrating large pre-trained vision-language models, allowing robots to condition their actions on semantic text prompts. Pioneering architectures like RT-1 [1] and RT-2 [2] demonstrated that massive, internet-scale pre-training could yield emergent semantic reasoning in robot control. Open-source efforts like OpenVLA [10] and the Octo model [18] further democratized these capabilities, serving as robust baselines for multi-embodiment manipulation. While models like SmolVLA and $\pi_{0.5}$ represent the current state-of-the-art in this domain, their output space remains inextricably linked to specific hardware geometries. Consequently, cross-embodiment transfer typically requires extensive target-domain data or complex, handcrafted kinematic alignment procedures.

### 2.2. World Models in Robotics

World models learn the underlying transition dynamics of an environment, allowing an agent to simulate future states based on hypothetical actions [6]. Prominent model-based reinforcement learning frameworks like the Dreamer series (DreamerV1-V3) [7] and DayDreamer [24] have demonstrated the immense sample-efficiency benefits of planning within a compact latent space for physical robots.

Recent advancements have scaled this concept to raw pixel inputs using large-scale video generative models and latent dynamics models, such as UniSim [25] and Genie [3]. By predicting future video frames or latent representations, these models implicitly capture physical laws, object permanence, and occlusion from unstructured video data. WAMs extend this line of research by directly extracting control policies from these rich, predictive latent spaces, utilizing the world model as a morphology-agnostic intermediary between perception and action.

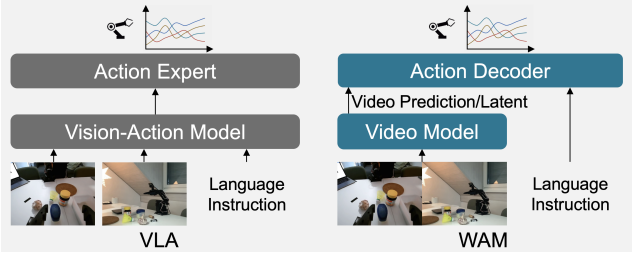
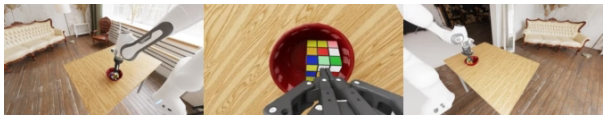
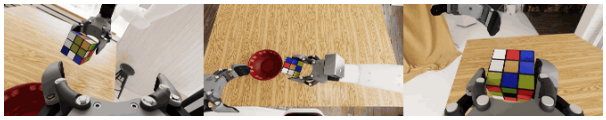



Figure 1. Architectures of compared frameworks. WAM utilizes a video model to synthesize future frames and infers actions based on these visual predictions, whereas traditional VLAs map observations directly through a Vision-Language Model to an action decoder. For the WAM architecture, we show an abstracted version of MimicVideo, whereas DreamZero utilizes joint prediction of both video and action spaces.



(a) DROID checkpoint



(b) AgiBot G1 checkpoint

Figure 2. Representative Isaac Sim snapshots from the preliminary DreamZero evaluation on the cube-to-bowl task. Each row shows left/body/right camera views.

## 3. Methodology and Experimental Setup

To rigorously test the core claims surrounding WAM, we established an evaluation framework designed to isolate individual variables. By fixing the underlying task suite and varying only the core studied factor—either the physical robot embodiment or the statistical composition of the training data—we eliminate confounding variables. Our primary evaluation metric is the task success rate. We evaluate this across 2 distinct initialization seeds with 10 rollouts each for the first experiment (Section 3.2), and over 10 rollouts per model and task for the second (Section 3.3).

As illustrated in Figure 1, we compare the modular WAM structure against the end-to-end VLA pipelines. To evaluate these frameworks, we constructed a comprehensive suite of tasks spanning both simulation and real-world environments, as detailed below.

### 3.1. Preliminary DreamZero Evaluation

Our project originally aimed to evaluate DreamZero's cross-embodiment transfer stability across real-world robot platforms. In the project proposal, we planned to establish a Franka-based baseline [5] and study asymmetric transfer to a target embodiment such as Reachy [14], while varying the amount of target-domain play data available for adaptation. However, access to the target embodiments was limited, making a controlled real-world cross-embodiment study infeasible within the project timeline. We therefore shifted the evaluation effort toward simulation.

We first attempted to use MuJoCo [19] as the simulation backend, but the DreamZero evaluation pipeline did not align well with the simulator interface and policy assumptions. We then transitioned to Isaac Sim [12], where we successfully deployed the official DreamZero checkpoints (Figure 2). The DROID checkpoint was executed on single-arm tabletop manipulation tasks, and the AgiBot G1 checkpoint was rolled out on dual-arm tasks after reducing high-frequency action jitter. These experiments confirmed that the DreamZero policy stack could be deployed in our simulation setup. However, task performance was limited, especially for the AgiBot checkpoint. Since this checkpoint was released primarily for fine-tuning rather than direct inference, its low success rate was difficult to interpret as evidence against DreamZero or the WAM formulation itself. Additional qualitative rollout frames are provided in the supplementary material (Sec. 6).

In parallel, we converted the EgoVerse grocery-bagging dataset into the LeRobot format required by DreamZero and trained the model for the bimanual Franka and ORCA-hand embodiment. However, evaluation posed a significant challenge: the trained policy required a matching simulation environment for the grocery-bagging task. Building this environment introduced additional variables, including camera calibration, object alignment, embodiment modeling, and contact dynamics. As these real-to-sim and evaluation-stack errors accumulated, it became difficult to distinguish policy failures from environment-construction failures.

Furthermore, DreamZero's substantial parameter count severely strained our available compute resources, exacerbating these evaluation challenges. In contrast, MimicVideo offers a significantly more tractable parameter count while fully preserving the core WAM predictive methodology.

For these reasons, we refined our final research question. Rather than reporting an inconclusive DreamZero-only benchmark restricted by computational constraints, we evaluated the broader claims motivated by DreamZero in more controlled and computationally feasible settings: cross-embodiment transfer with MimicVideo in simulation, and heterogeneous-data learning on the SO-101 [17] real-world setup. This pivot allowed us to rigorously isolate the two central claims while avoiding confounds from an incomplete DreamZero evaluation environment and compute bottlenecks.

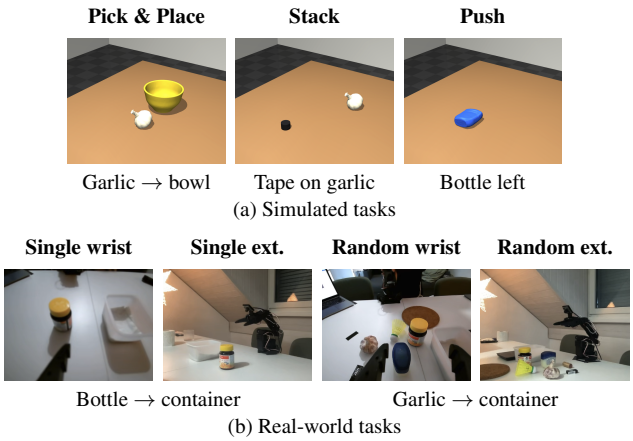



Figure 3. Task suite utilized for evaluation. Simulation tasks include Pick & Place, Stack, and Push variations in MuJoCo. Real-world tasks involve Single Task and Random Task distributions conducted on the SO-101 robot.

### 3.2. Experiment 1: Evaluating Cross-Embodiment Transfer

The first experiment was designed specifically to test the capacity of WAMs to separate high-level task understanding from low-level embodiment execution. Because access to a variety of physical embodiments in the real world was severely limited, and to ensure environmental consistency while preventing real-world visual confounders from affecting the morphological tests, this experiment was conducted entirely within the MuJoCo simulation environment utilizing the tasks shown in Figure 3.

Our base model for this evaluation was MimicVideo, initially pre-trained on BridgeData V2 [22] utilizing the WidowX 250 robotic arm [20], and subsequently fine-tuned on a standard pick-and-place task. To evaluate transferability, we froze the core visual and predictive pathways of the pre-trained model. We then fine-tuned only the lightweight action decoder network on four distinct target embodiments: the SO-101 [17], the Franka Panda [5], the Universal Robots UR5e [21], and the KUKA iiwa 14 [11] (see Figure 4). This adaptation phase was strictly limited to only 15 minutes of teleoperated play data (here defined as non-repetitive data from simple tasks with a large variety of manipulated objects) for each specific target embodiment. Rather than training a single, universal decoder across all target robots, we trained a distinct, independent action decoder for each individual embodiment. This allowed us to rigorously assess the isolated transfer quality and kinematic adaptation required for each specific morphology. For this simulation experiment, we kept MimicVideo's original end-

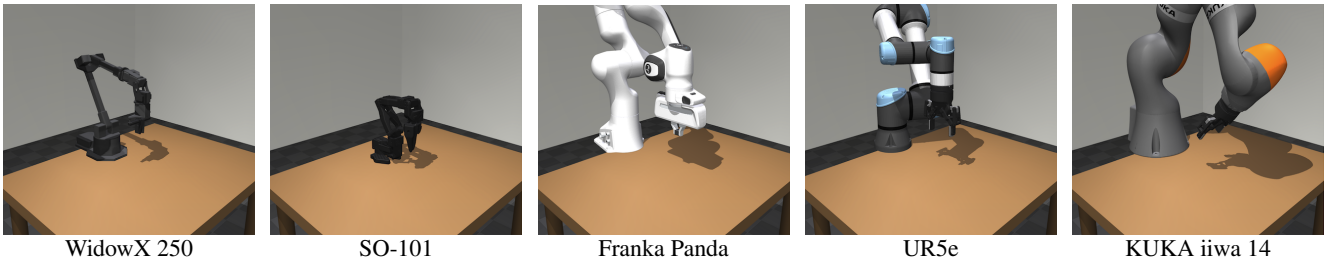



Figure 4. Robot embodiments evaluated for cross-embodiment transfer. The base model was pre-trained on the WidowX 250 (far left) and adapted to the SO-101, Franka Panda, Universal Robots UR5e, and KUKA iiwa 14 target embodiments.

| Dataset | Episodes | Frames | Duration |
|---|---|---|---|
| Homogeneous | 110 | 43,095 | 23.94 min |
| Heterogeneous | 153 | 54,647 | 30.36 min |

Table 1. Summary of the SO-101 datasets used in the real-world experiments. Both datasets were recorded at 30 FPS.

| Task type | Active tasks | Episodes |
|---|---|---|
| Pick-and-place | 23 | 69 |
| Stack | 14 | 42 |
| Push | 14 | 42 |

Table 2. Task-type summary of the heterogeneous SO-101 dataset. Each active task description has three recorded episodes.

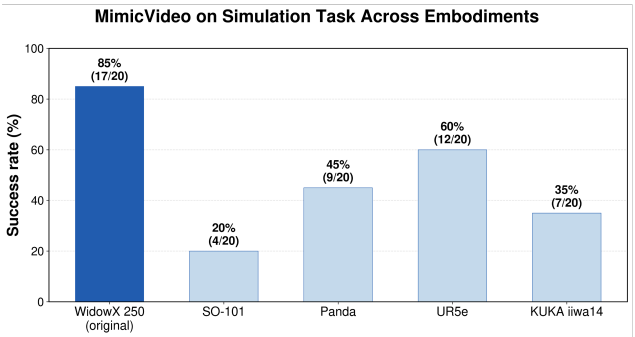



Figure 5. Success rates of MimicVideo on the simulation task across different target embodiments. While the original WidowX 250 achieves 85%, performance drops significantly as the morphological gap from the source robot widens, with the SO-101 achieving only 20%.

effector-space action formulation. This representation is suitable for cross-embodiment transfer because the output dimension does not depend on the robot's number of joints, and the simulated data collection and rollout pipeline consistently uses inverse kinematics during both demonstration collection and policy execution.

### 3.3. Experiment 2: Evaluating Data Heterogeneity

The second experiment evaluates the effect of data composition on policy performance while keeping the robot hardware fixed. Unlike the previous experiment, this study was conducted on the real SO-101 robotic arm. We collected two training datasets with varying levels of diversity, as detailed in Tables 1 and 2.

We also adapted the MimicVideo action decoder for this experiment. The original MimicVideo checkpoint predicts end-effector trajectories, where the current end-effector pose serves as the reference frame for the future action sequence. However, in our SO-101 setup, demonstrations are collected through leader-arm teleoperation, and the follower arm directly tracks joint positions. Therefore, we replaced the original end-effector decoder with a joint-space decoder that predicts per-timestep joint-position deltas, reinitializing the final decoder layer to match the SO-101 output dimension. This modification avoids adding an inverse kinematics (IK) conversion step exclusively for MimicVideo, which would introduce an additional controller prior and potentially produce joint trajectories that deviate from the teleoperated demonstrations.

We adopted a two-stage training approach for MimicVideo: first, we fine-tuned the video backbone using LoRA [8]. Then, we froze all video-related parameters and fully trained only the modified action decoder. We compared its performance against two VLA baselines: SmolVLA and $\pi_{0.5}$. While the exact action parameterization differs across methods—SmolVLA predicts absolute joint positions, $\pi_{0.5}$ predicts relative joint actions, and our MimicVideo variant predicts joint-position deltas—all methods are evaluated uniformly through the SO-101 joint-control interface.

## 4. Results

### 4.1. Cross-Embodiment Transfer Performance

The results from our simulation-based cross-embodiment experiment, visualized in Figure 5, illuminate both the capabilities and the severe limitations of adapting WAMs to new morphologies using minimal demonstration data.

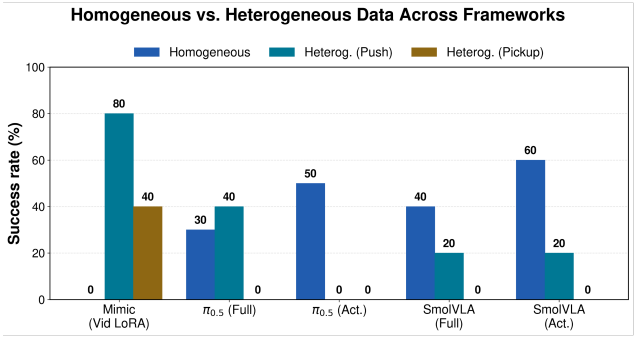



Figure 6. Performance comparison of Homogeneous vs. Heterogeneous data across frameworks. MimicVideo (Vid LoRA) maintains a baseline robustness in heterogeneous tasks, whereas standard VLAs experience significant performance degradation or total collapse when presented with multi-task data.

When evaluated on the original source embodiment (WidowX 250), the baseline MimicVideo model achieved a highly robust success rate of 85%. As we transitioned to novel target embodiments, performance degraded across the board. The Universal Robots UR5e maintained a respectable 60% success rate. The Franka Panda achieved 45%, while the highly redundant KUKA iiwa 14 dropped to 35%. Finally, the SO-101 exhibited the most severe degradation, completing only 4 out of 20 rollouts for a meager 20% success rate. These results indicate that while 15 minutes of play data transfers basic dynamics, the ultimate success rate is inextricably linked to the structural and kinematic similarity between the source and target robots.

### 4.2. Homogeneous Data Performance

The real-world experiments on the SO-101 arm revealed stark contrasts in how different architectural paradigms handle data diversity, as detailed in Figure 6.

In the Homogeneous setting, traditional VLA baselines notably outperformed MimicVideo. SmolVLA and $\pi_{0.5}$ (with action-decoder-only tuning) achieved 60% and 50% success rates, respectively. In contrast, MimicVideo failed completely in this setting, yielding a 0% success rate.

To understand this discrepancy, we qualitatively analyzed the failure modes of both approaches. For the VLA baselines, the primary bottleneck was grasp robustness; when the object's initial position varied slightly, the models often struggled to secure a stable grasp. MimicVideo, however, exhibited a completely different failure pattern. The policy consistently succeeded in reaching and grasping the bottle, but the failures mainly occurred during the post-grasp transport and placement phase (Figure 7).

This placement failure stems directly from the WAM paradigm. As illustrated in Figure 8, comparing the predicted "dream" video to the ground truth reveals significant temporal and spatial discrepancies. While the ground truth shows a slow, deliberate lift and precise alignment with the container, the predicted future advances the transport motion far too aggressively. In the prediction, the bottle lacks spatial alignment and essentially skips past the container. Because the action decoder strictly follows this flawed visual plan, the physical rollout mirrors the prediction error, resulting in a failed placement.

In short, MimicVideo failed in the homogeneous setting not because it could not grasp the bottle, but because its predicted future advanced the post-grasp transport too aggressively, causing the action decoder to follow a visual plan in which the bottle passed by the container instead of being placed into it. This highlights a critical limitation of video-prediction-based policies like MimicVideo and DreamZero: in low-diversity, contact-rich tasks where precise spatial alignment is required, the quality of the predicted video becomes a strict bottleneck. Small prediction errors in temporal progression or height are severely amplified during downstream action execution.

### 4.3. Heterogeneous Data Performance

Conversely, the performance landscape shifted dramatically in the Heterogeneous setting. Here, MimicVideo achieved the strongest results, reaching 80% success on push tasks and 40% on pickup tasks. Notably, these tasks involved zero-shot compositional scenarios—unseen combinations of seen objects and tasks—as well as varied spatial arrangements, where MimicVideo successfully executed the push maneuvers. The VLA baselines, however, suffered severe degradation. Fully-tuned $\pi_{0.5}$ dropped to 40% on push and 0% on pickup, with its action-only variant collapsing entirely (0% across both). SmolVLA similarly degraded, managing only 20% on push tasks and failing all pickups. These results indicate that while MimicVideo struggles with the strict precision required in homogeneous contact-rich tasks, it highly benefits from the semantic compositionality learned from heterogeneous visual-task data, maintaining robustness where standard VLAs collapse.

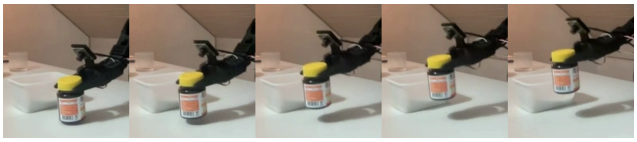
## 5. Discussion and Conclusion

### 5.1. Analysis of Cross-Embodiment Transfer

Our evaluation of cross-embodiment transfer indicates that while the theoretical promise of WAMs is morphology-agnostic task reasoning, empirical success rates remain heavily dependent on the kinematic similarity between the source and target robots. A mere 15 minutes of play data allows the action decoder to map imagined latent states to basic joint velocities for structurally similar arms, but it struggles to compensate for vastly different kinematic chains without significantly more target-domain data. For instance, transferring policies from the small, 6-DoF WidowX 250 source robot to larger 6-DoF industrial arms like the UR5e



Figure 7. MimicVideo rollout trajectory for the homogeneous bottle-to-container task. Frames are sampled every minute. The rollout shows that the policy grasps the bottle but fails during the post-grasp transport and placement phase.



(a) Ground truth



(b) Video prediction

Figure 8. Failure case in homogeneous video prediction. After grasping the bottle, the predicted future moves it past the container rather than into it. Both rows use the same fixed crop in the external camera view.

yields moderate performance degradation. However, transferring to highly redundant 7-DoF systems with different joint limits like the Franka Panda or KUKA iiwa 14, or to arms with vastly different scale and workspace configurations like the SO-101, results in a severe drop in success rates.

## 5.2. Analysis of Data Heterogeneity

The data heterogeneity experiment reveals a more nuanced picture than a simple WAM-over-VLA conclusion. In the homogeneous setting, VLA action-decoder tuning was more reliable than MimicVideo, suggesting that direct action-space policies can still be preferable when the task distribution is narrow and the required behavior is dominated by precise contact execution. MimicVideo's failure in this setting highlights a key weakness of WAMs: errors in video prediction near contact events can translate into incorrect action decoding.

However, the heterogeneous setting supports the central motivation behind WAMs. While the VLA baselines degraded sharply, especially on pickup tasks, MimicVideo maintained strong performance on push tasks and nonzero performance on pickup tasks. This suggests that world-model-based prediction can help compose behaviors across diverse objects and instructions, but its advantage depends on the fidelity of the predicted visual trajectory.

One implementation detail is important when interpreting these results. The SO-101 baselines do not use identical action parameterizations: SmolVLA predicts absolute joint positions, $\pi_{0.5}$ predicts relative joint actions, and our MimicVideo variant predicts joint-position deltas. Therefore, this experiment should not be read as a perfectly action-space-matched benchmark. Instead, it compares the models under a common SO-101 joint-control interface while avoiding an additional inverse-kinematics layer for only MimicVideo.

Overall, our results support the heterogeneous-data claim only partially: WAMs improve multi-task generalization, but they are not uniformly superior and remain fragile in contact-rich regimes.

## 5.3. Limitations and Future Work

The scope of this empirical study was constrained by restricted access to diverse physical hardware and the immense computational resources required to train generative video models at scale. Future work should evaluate these findings in more realistic and diverse simulation platforms, as well as across broader real-world robot setups. In particular, it remains unclear how much play data is needed to improve video-action policies, and how the benefit changes with the available play budget.

Another important direction is to better understand and mitigate the video-prediction bottleneck observed in the SO-101 homogeneous setting. Future studies should investigate whether the post-grasp prediction error appears consistently across other homogeneous tasks, why it emerges, and how latent video predictions can be refined around physical contact and placement. Beyond this failure analysis, future work should also test video-action policies in more generalized cross-embodiment real-world settings, including transfer from single-arm to multi-arm systems. Finally, motivated by DreamZero's claim that human video data can help learn robot action spaces, it remains important to evaluate whether incorporating human datasets improves sample efficiency or downstream real-world performance in our setting.

## References


[1] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, K. Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. In *Robotics: Science and Systems*, 2022. 2

[2] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, K. Choromanski, Tianli Ding, Danny Driess, Kumar Avinava Dubey, Chelsea Finn, Peter R. Florence, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 3256–3299. PMLR, 2023. 2

[3] Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, M. Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. *International Conference on Machine Learning*, 2024. 2

[4] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin CM Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems*. Robotics: Science and Systems Foundation, 2023. 2

[5] Franka Emika. Panda collaborative robot. https://www.franka.de/, 2020. 3

[6] David Ha and Jürgen Schmidhuber. World models. *arXiv.org*, 2018. 2

[7] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv.org*, 2023. 2

[8] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021. 4

[9] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $_{0.5}$: a vision-language-action model with open-world generalization, 2025. 2

[10] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, A. Balakrishna, Suraj Nair, Rafael Rafailov, E. Foster, Grace Lam, Pannag R. Sanketi, et al. Openvla: An open-source vision-language-action model. *Conference on Robot Learning*, 2024. 2

[11] KUKA. Lbr iiwa 14 r820. https://www.kuka.com/en-de/products/robotics-systems/industrial-robots/lbr-iiwa, 2014. 3

[12] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, M. Macklin, David Hoeller, N. Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. In *NeurIPS Datasets and Benchmarks*, 2021. 3

[13] Jonas Pai, Liam Achenbach, Victoriano Montesinos, Benedek Forrai, Oier Mees, and Elvis Nava. mimic-video: Video-action models for generalizable robot control beyond vlas. In *arXiv.org*, 2025. 2

[14] Pollen Robotics. Reachy: An open-source humanoid robot. https://www.pollen-robotics.com/reachy/, 2021. 3

[15] Dean A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *Advances in Neural Information Processing Systems*. 2015. 2

[16] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics*, pages 627–635, 2010. 2

[17] Standard Bots. So-101 robotic arm. https://standardbots.com/, 2024. 3

[18] O. Team, Dibya Ghosh, H. Walke, Karl Pertsch, Kevin Black, Oier Mees, S. Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. In *Robotics: Science and Systems*. Robotics: Science and Systems Foundation, 2024. 2

[19] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, IEEE, 2012. 3

[20] Trossen Robotics. Widowx 250 robot arm. https://www.trossenrobotics.com/widowx-250-robot-arm.aspx, 2021. 3

[21] Universal Robots. Ur5e collaborative robot. https://www.universal-robots.com/products/ur5-robot/, 2018. 3

[22] H. Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Maximilian Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Q. Vuong, Andre Wang He, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pages 1723–1736. PMLR, 2023. 3

[23] Linfeng Wang and Deok Jin Lee. Smolvla: An open-source vision-language-action model for modern robotics. https://smolvla.net, 2025. 2

[24] Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. Daydreamer: World models for physical robot learning. In *Conference on Robot Learning*, pages 2226–2240. PMLR, 2022. 2

[25] Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *International Conference on Learning Representations*, 2023. 2

[26] Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, Y. Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. In *arXiv.org*, 2026. 1

[27] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems*. Robotics: Science and Systems Foundation, 2023. 2

## Author Contributions

### Work Distribution

The project was a highly collaborative effort, with the workload distributed among the teammates as follows (all authors contributed to the final paper writing and editing):

- **Dohyung Kim:** Configured the Isaac Sim environment and evaluated DreamZero checkpoints for the preliminary DreamZero evaluations. Co-executed the majority of the real-world data collection and physical robot rollouts on the SO-101. Independently managed the deployment, fine-tuning, and evaluation of the $\pi_{0.5}$ model for the real-world experiments.
- **Eugene Seo:** Configured the MuJoCo environment for the preliminary DreamZero evaluations and conducted the DROID bimanual robustness test detailed in the supplementary material. Co-executed the majority of the real-world data collection and physical robot rollouts on the SO-101. Independently managed the deployment, fine-tuning, and evaluation of the MimicVideo architecture for the real-world experiments.
- **Rohan Walia:** Fine-tuned and evaluated the DreamZero AgiBot checkpoint on the converted datasets during the preliminary phase. Solely executed the comprehensive simulation-based cross-embodiment study, which included establishing the simulation teleoperation pipeline and fine-tuning MimicVideo on the collected task-specific and play data across all target robot embodiments.
- **Eryk Halicki:** Managed the complex dataset conversion pipelines (e.g., converting the EgoVerse dataset to the LeRobot format) for the preliminary DreamZero study. Independently managed the deployment, fine-tuning, and evaluation of the SmolVLA baseline for the real-world experiments.

## Code Implementation

To ensure reproducibility and strictly isolate the variables in our experiments, our submitted codebase consists of a mixture of custom implementations and modified upstream repositories.

### Custom Implementation

The following components were written entirely by our team or heavily modified for this study:

- **Simulation and Teleoperation (`vam_cross`):** We developed this module from scratch. It contains the MuJoCo-based cross-embodiment simulation environment, the teleoperation pipeline used for target-domain data collection across the five different robot arms, the definition of the custom task suite, and the export scripts for converting HDF5 records into Zarr datasets.
- **WAM Modifications (`mimic-video`):** We implemented critical modifications to the upstream MimicVideo framework to support our physical hardware. Specifically, we rewrote the action decoder to directly output joint-space deltas for the SO-101 leader-follower interface (bypassing external inverse kinematics) and built the data preprocessing pipeline to ingest the custom SO-101 homogeneous and heterogeneous datasets.
- **Baselines/Data Conv. (`dreamzero/dreamdifferent`):** We built extensive custom extensions under the `dreamdifferent` module. This includes the configuration, deployment, and evaluation wrappers for the VLA baselines (SmolVLA and $\pi_{0.5}$), as well as the conversion scripts to translate the EgoVerse grocery dataset into the LeRobot format.

### External Code and Assets

Our project builds upon the following external open-source codebases and libraries:

- **Model Backbones:** We utilized the official repositories and base weights for MimicVideo (built on Cosmos-Predict2), DreamZero, SmolVLA (via the Hugging Face LeRobot library), and $\pi_{0.5}$ (via the OpenPI framework).
- **Simulators and Datasets:** Our simulation environments relied heavily on the standard APIs provided by MuJoCo and Isaac Sim. Pre-training and baseline data were sourced from public repositories, including BridgeData V2 and EgoVerse.

# Can Robots Imagine Better Actions? World Action Models vs VLAs on Cross-Embodiment Transfer and Learning from Heterogeneous Data

Supplementary Material

## 6. Additional DreamZero Rollouts

This section provides additional qualitative frames from our preliminary DreamZero evaluation in Isaac Sim. Each rollout is shown as temporally ordered frames, separated by camera view. These rollouts verify that the official checkpoints could be deployed in our setup, but we do not use them as a conclusive benchmark.

In all figures, rows from top to bottom correspond to the body, left, and right camera views, and columns show uniformly sampled frames over time.

### 6.1. DROID Checkpoint

We first tested the official DreamZero-DROID checkpoint using the Isaac Sim evaluation script provided by the DreamZero repository. The checkpoint ran without additional model-side modification and produced stable rollouts in the DROID single-arm tabletop setup. We evaluated three instructions: "put the cube in the bowl," "pick up the can and put it in the mug," and "put the banana in the bin." All three tasks were completed successfully in our rollouts, confirming that the official DROID checkpoint and evaluation script worked as expected in our setup. Representative rollouts are shown in Figure 9.

### 6.2. AgiBot G1 Checkpoint

We then tested the official DreamZero-AgiBot G1 checkpoint in Isaac Sim on the same set of tabletop instructions. The initial rollout showed high-frequency jitter in the predicted actions. Following the action-chunk smoothing procedure described in DreamZero [26], we upsampled each predicted action chunk, applied a Savitzky-Golay filter, and downsampled it back to the original action resolution. This removed the visible jitter and made the rollout stable enough for qualitative evaluation.

After smoothing, the cube-to-bowl task occasionally succeeded, but the overall success rate remained low. The can-to-mug and banana-to-bin tasks did not reliably complete. Representative rollouts are shown in Figure 10. We interpret these results cautiously because the official repository presents the AgiBot G1 checkpoint primarily as a post-training or fine-tuning checkpoint for new embodiments, rather than as the main released inference checkpoint.[1]

[1] https://github.com/dreamzero0/dreamzero

### 6.3. DROID Bimanual Robustness Test

We additionally tested whether a DreamZero DROID checkpoint trained for single-arm manipulation could be deployed in a bimanual scene. The setup followed the original DROID-style input as closely as possible: two RGB camera views and right-arm proprioceptive state were provided to the policy, and only the right arm was controlled. The additional left arm was introduced to test robustness to an embodiment change that should ideally be irrelevant to the single-arm task.

The rollout showed limited robustness in this setting (Figure 11). The policy performance degraded when the additional arm appeared in the camera views, especially in the side-view input. This suggests that the checkpoint was sensitive not only to the controlled embodiment, but also to visual changes introduced by the surrounding robot morphology. We therefore treat this experiment as a qualitative robustness probe rather than a formal benchmark.

## 7. SO-101 Dataset Details

We collected two SO-101 real-world datasets using leader-arm teleoperation. The demonstrations were recorded from a leader arm and replayed as trajectories for the SO-101 follower arm. The heterogeneous dataset involves 18 object or target categories: black eraser, black pen, black scale, black tape, blue bottle, cork, cork pad, garlic, pink phone, red pen, scissors, shuttlecock, white container, white phone, white tape, yellow bottle, yellow bowl, and yellow pen (Table 3).

Figures 12 and 13 show representative SO-101 demonstrations used in our real-world experiments. In each figure, the top row shows the external front camera view and the bottom row shows the wrist camera view. Columns correspond to uniformly sampled frames from a single demonstration episode.

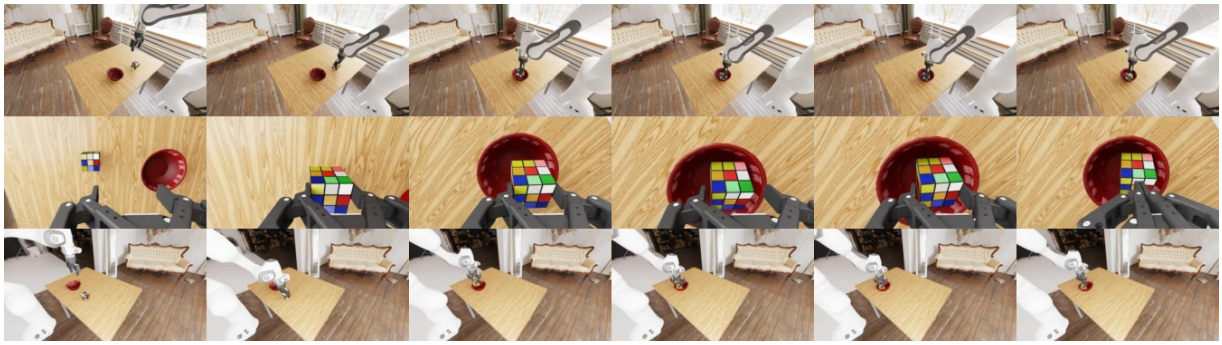
## 8. SO-101 Heterogeneous Experiment Details

This section provides additional details and qualitative visualizations for the SO-101 real-world experiments. We include rollout examples for SmolVLA, $\pi_{0.5}$, and MimicVideo across the heterogeneous pick-and-place and push tasks. These rollouts complement the quantitative success rates reported in the main paper by illustrating how each model behaves under object-layout changes.

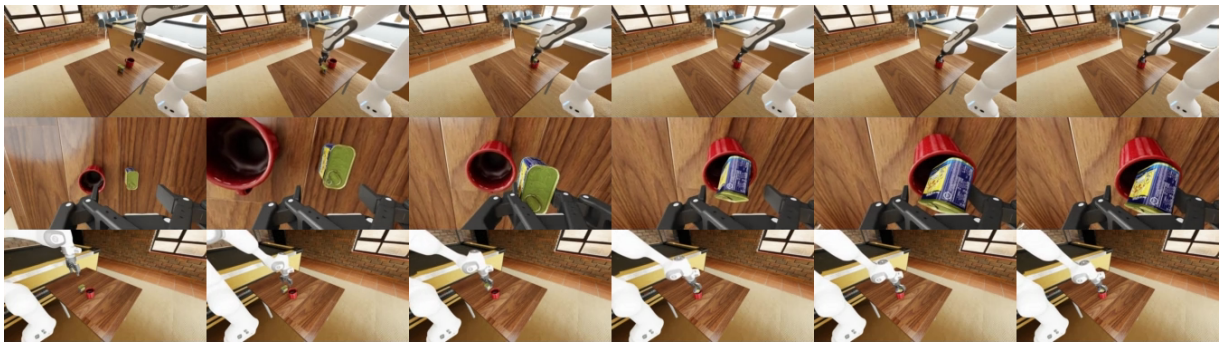
For the push task, we additionally tested robustness by changing the initial left-right placement of the target object. This was not intended as a separate in-domain/out-

of-domain split, but as a qualitative probe of whether each policy could adapt its motion to the current object configuration. VLAs exhibited overfit behavior: instead of adjusting the pushing trajectory according to the shifted phone location, the policy repeated a similar motion pattern from the training distribution and failed to interact with the object correctly. Representative rollout visualizations are shown in Figures 14 to 16.

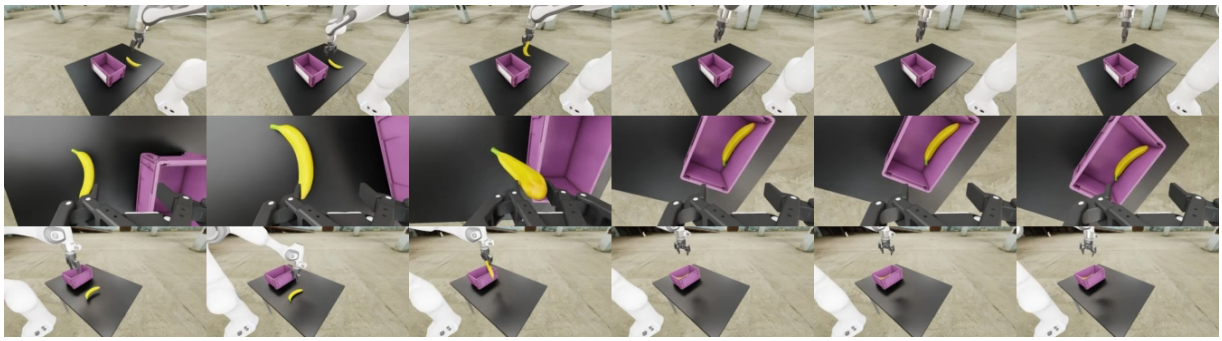
We also visualize representative MimicVideo future-video predictions in Figure 17 and analyze a homogeneous prediction failure case in Figure 8. These examples illustrate how prediction quality affects downstream action execution, especially in contact-rich pick-and-place tasks.



(a) Put the cube in the bowl

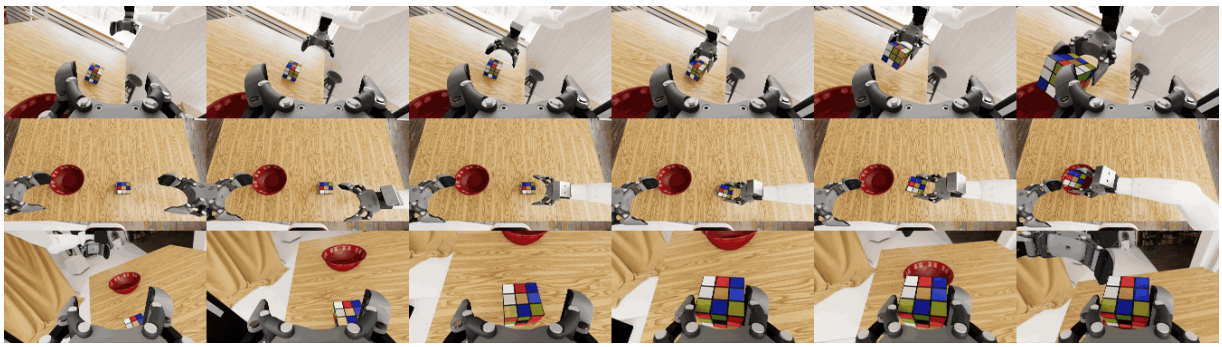


(b) Pick up the can and put it in the mug



(c) Put the banana in the bin

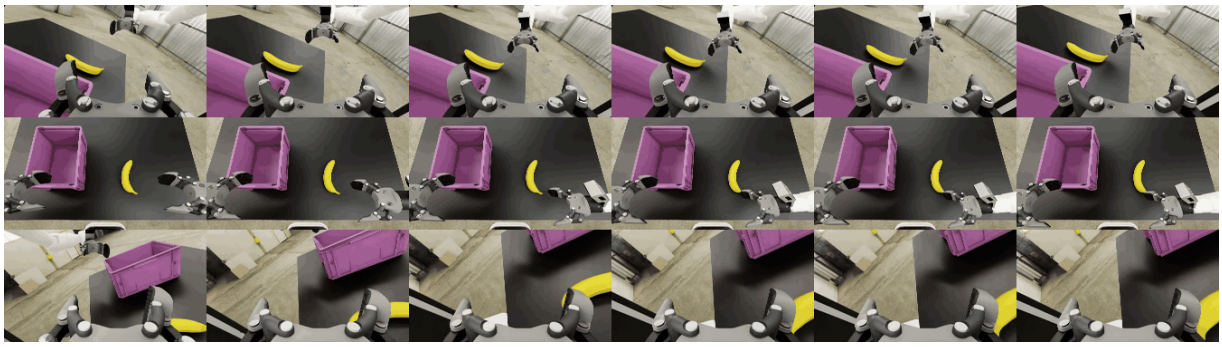
Figure 9. DROID checkpoint rollouts in Isaac Sim. In each subfigure, rows correspond to camera views and columns show uniformly sampled frames over time.



(a) Put the cube in the bowl

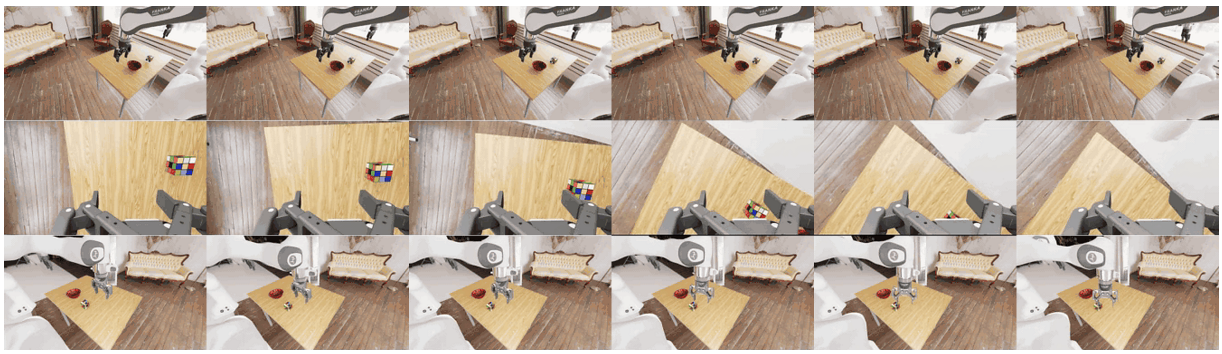


(b) Pick up the can and put it in the mug

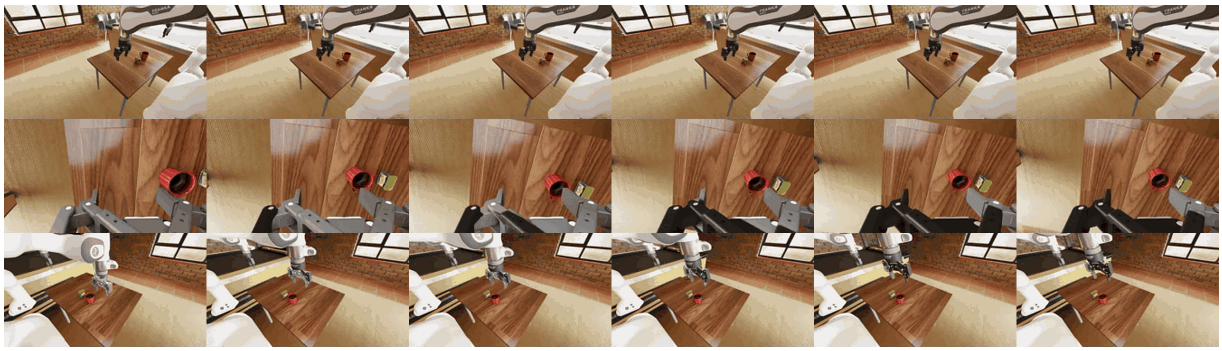


(c) Put the banana in the bin

Figure 10. AgiBot G1 checkpoint rollouts in Isaac Sim. In each subfigure, rows correspond to camera views and columns show uniformly sampled frames over time.



(a) Bimanual robustness rollout 1



(b) Bimanual robustness rollout 2

Figure 11. DROID checkpoint rollouts in the bimanual robustness test. In each subfigure, rows correspond to camera views and columns show uniformly sampled frames over time.

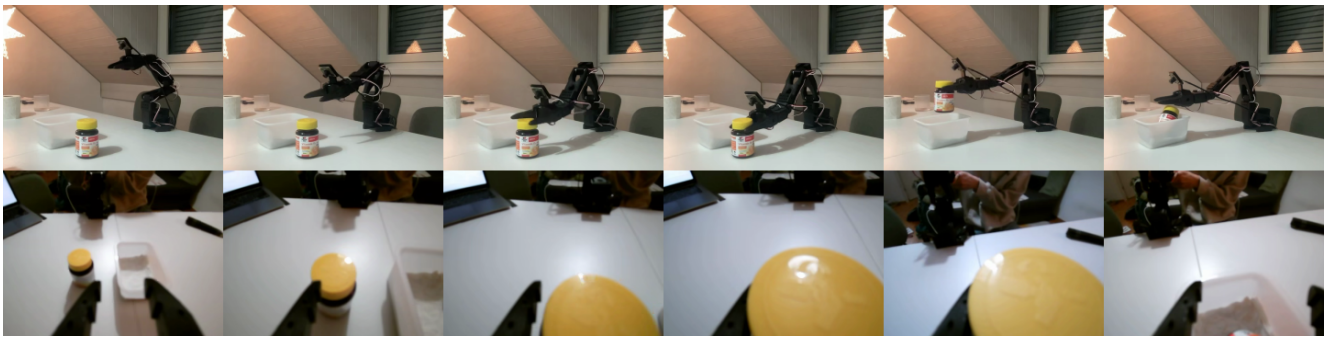


Figure 12. Representative homogeneous SO-101 demonstration for picking up the bottle and placing it into the container.

| Task type | Task description |
|---|---|
| Pick-and-place | pick up the garlic and place it into the white container |
| | pick up the yellow bottle and place it into the white container |
| | pick up the cork and place it into the white container |
| | pick up the blue bottle and place it into the white container |
| | pick up the shuttlecock and place it into the white container |
| | pick up the black eraser and place it into the white container |
| | pick up the black tape and place it into the white container |
| | pick up the blue bottle and place it on the black scale |
| | pick up the shuttlecock and place it on the black scale |
| | pick up the yellow bottle and place it on the black scale |
| | pick up the black eraser and place it on the black scale |
| | pick up the cork and place it on the black scale |
| | pick up the garlic and place it on the black scale |
| | pick up the black tape and place it on the black scale |
| | pick up the blue bottle and place it on the cork pad |
| | pick up the shuttlecock and place it on the cork pad |
| | pick up the yellow bottle and place it on the cork pad |
| | pick up the black eraser and place it on the cork pad |
| | pick up the cork and place it on the cork pad |
| | pick up the garlic and place it on the cork pad |
| | pick up the black tape and place it on the cork pad |
| | pick up the black pen and place it into the white container |
| | pick up the red pen and place it into the yellow bowl |
| Stack | stack the black tape on the garlic |
| | stack the shuttlecock on the garlic |
| | stack the black tape on the yellow bottle |
| | stack the shuttlecock on the yellow bottle |
| | stack the blue bottle on the yellow bottle |
| | stack the black eraser on the yellow bottle |
| | stack the garlic on the yellow bottle |
| | stack the cork on the yellow bottle |
| | stack the black tape on the blue bottle |
| | stack the shuttlecock on the blue bottle |
| | stack the yellow bottle on the blue bottle |
| | stack the black eraser on the blue bottle |
| | stack the garlic on the blue bottle |
| | stack the cork on the blue bottle |
| Push | push the yellow bottle to the left |
| | push the blue bottle to the left |
| | push the black tape to the left |
| | push the black eraser to the left |
| | push the shuttlecock to the right |
| | push the garlic to the right |
| | push the white tape to the right |
| | push the pink phone to the right |
| | push the yellow bowl to the left |
| | push the red pen to the left |
| | push the black pen to the left |
| | push the white phone to the right |
| | push the yellow pen to the right |
| | push the scissors to the right |

Table 3. Full active task list of the heterogeneous SO-101 dataset. The dataset contains 51 active task descriptions, each with three recorded episodes.



(a) Pick-and-place: pick up the garlic and place it into the white container



(b) Stack: stack the black tape on the garlic



(c) Push: push the yellow bottle to the left

Figure 13. Representative heterogeneous SO-101 demonstrations. In each subfigure, the top row shows the external front camera view and the bottom row shows the wrist camera view. Columns correspond to uniformly sampled frames from a single demonstration episode.



(a) MimicVideo



(b) $\pi_{0.5}$

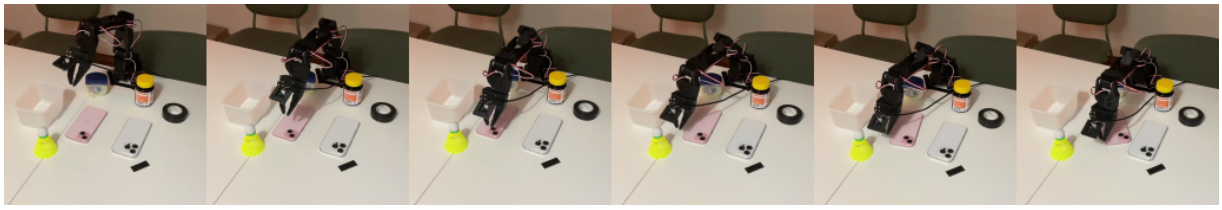


(c) SmolVLA

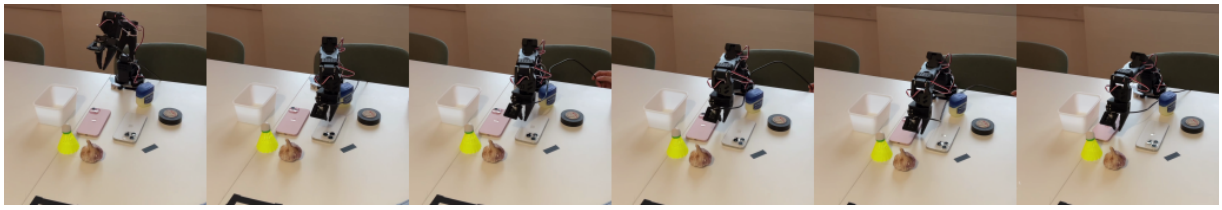
Figure 14. Representative real-world rollouts for the heterogeneous pick-and-place task, "pick up the garlic and place it into the white container."



(a) MimicVideo



(b) $\pi_{0.5}$

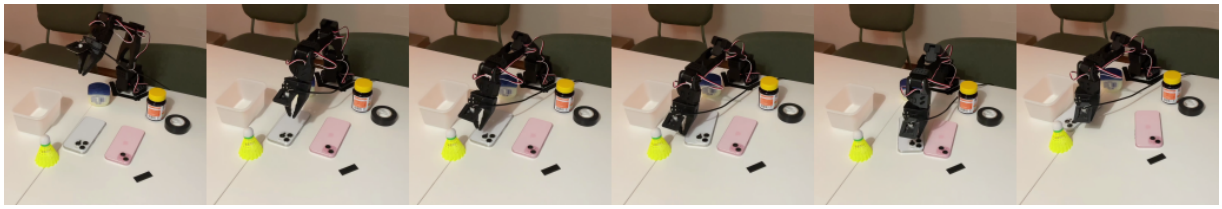


(c) SmolVLA

Figure 15. Representative real-world rollouts for the heterogeneous push task, "push the pink phone to the right," with the pink phone initially placed on the right side of the workspace.



(a) MimicVideo

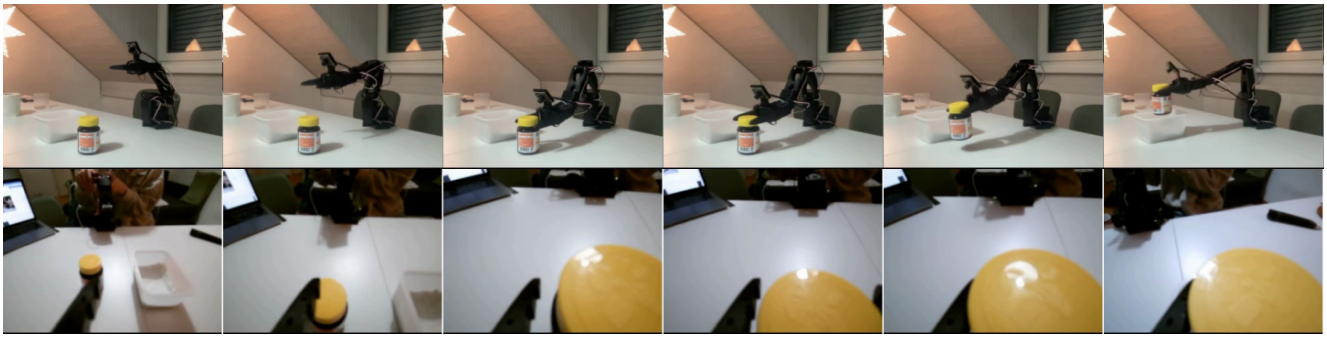


(b) $\pi_{0.5}$

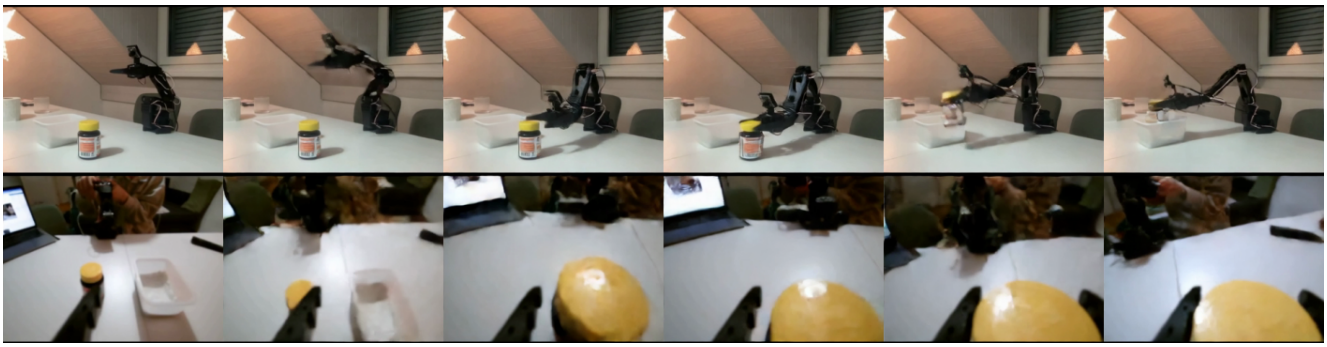


(c) SmolVLA

Figure 16. Representative real-world rollouts for the heterogeneous push task, "push the pink phone to the right," with the pink phone initially placed on the left side of the workspace.



(a) Homogeneous ground truth



(b) Homogeneous prediction



(c) Heterogeneous ground truth



(d) Heterogeneous prediction

Figure 17. Ground-truth and predicted video rollouts for representative SO-101 tasks. Each image contains two camera views, with the external view on top and the wrist view on the bottom. Columns correspond to uniformly sampled frames over time.