WORLD ACTION LEARNING VIA INTERACTION-CENTRIC SPECTRAL LATENT GUIDANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning general-purpose robot policies requires large-scale real-world interaction data, yet collecting robot demonstrations through teleoperation remains expensive and difficult to scale. Egocentric videos provide a rich source of human interaction experience that shares task-relevant semantics with robotic manipulation, creating an opportunity to align human and robot actions in a shared latent action space for cross-embodiment knowledge transfer. However, existing latent action approaches typically infer actions through reconstruction between consecutive frames, which are not inherently interaction-centric and can be dominated by nuisance variations, such as ego-camera motion. Moreover, although human interactions and robot actions share interaction semantics, they often exhibit substantially different temporal dynamics, making direct transfer to robot policies challenging. To address these challenges, we propose **WING** (**W**orld Action Learning via **IN**teraction-Centric Spectral Latent **G**uidance), a framework for transferring interaction knowledge from egocentric videos to robot policies. First, we introduce an interaction–motion decoupling mechanism that separates interaction-relevant signals from task-irrelevant motion and selectively distills the interaction-centric components into latent actions. Second, motivated by the observation that cross-embodiment task semantics are primarily encoded in slowly varying temporal structures, we identify shared components between egocentric latent actions and robot behaviors in the spectral domain and use them as guidance for action generation at inference time. With these designs, WING achieves average success rates of 99.20% on LIBERO, 93.80% on RoboTwin 2.0, and 57.7% on RoboCasa–GR1, while also demonstrating strong performance across four real-world tasks under diverse generalization settings. These results demonstrate that WING can effectively distill embodied interaction knowledge from large-scale egocentric videos and transfer it to robot control, providing a scalable pathway for acquiring physical interaction knowledge from human experience.

1 INTRODUCTION

Building generalist robot models that can reliably operate across diverse real-world environments requires large-scale and diverse training data (Intelligence et al., 2025; Kim et al., 2024; Li et al., 2026b; Ye et al., 2026b; Shou et al., 2026; Hu et al., 2026). Directly collecting such data through robot teleoperation remains costly and difficult to scale. Human videos (Deng & Zhou, 2026; Goyal et al., 2017; Wang et al., 2023; Punamiya et al., 2026; Grauman et al., 2022), in contrast, contain abundant embodied interactions across diverse tasks and environments, offering a potentially scalable source of experience for robot learning. *Egocentric videos* (Hoque et al., 2026; Punamiya et al., 2026; Zheng et al., 2026b; Grauman et al., 2022; Wang et al., 2023; Damen et al., 2021) are particularly appealing in this context, as they capture rich human-object interactions from a first-person perspective that closely resembles the visual observations available to robots.

Transferring human experiences to robot control remains challenging because egocentric videos lack direct and generalizable action supervision. Explicit annotations such as hand poses are often tightly coupled to human morphology and kinematics, limiting their ability to provide transferable supervision for robot action learning. *Latent action learning* (Bu et al., 2025; Ye et al., 2025; Wei et al., 2026; Gao et al., 2026) offers a more flexible alternative by inferring action-like representations directly from visual transitions and encoding them in a latent space, without requiring explicit

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

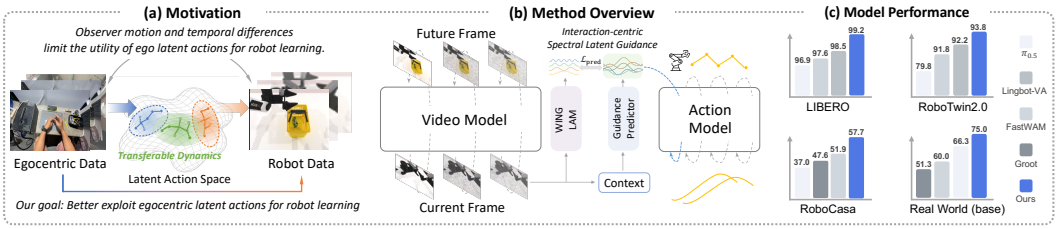


Figure 1: (a) **Motivation:** Egocentric videos provide rich interaction experience, but transferring this knowledge to robot policies remains challenging. We aim to extract transferable latent actions from egocentric data for robot learning. (b) **Method:** WING learns interaction-centric latent actions, extracts low-frequency temporal structure in the frequency domain, and uses it as guidance for robot action generation. (c) **Performance:** WING consistently achieves strong performance among the compared methods across simulation and real-world benchmarks.

action labels. However, extracting *transferable interaction knowledge* from visual transitions is fundamentally different from merely explaining how images change over time. Existing latent action models are typically trained to reconstruct future observations, encouraging them to capture any visually predictive variation rather than selectively modeling interaction-relevant dynamics. This leads to a central question: ***How can latent action learning extract interaction-centric knowledge from large-scale egocentric videos for transferable robot learning?***

We first examine the sources of nuisance factors in latent actions extracted from egocentric videos. We observe that a major source stems from **task-irrelevant observer motion**. Unlike robot manipulation settings where the head camera is typically mounted on the robot and remains stable, human-captured egocentric videos often exhibit substantial viewpoint changes caused by rapid head and body movements. These camera-induced variations can become entangled with interaction-related dynamics and even dominate the learned latent representations (Xu et al., 2026; Lee et al., 2026). However, transferable information primarily lies in the underlying hand-object interactions rather than task-irrelevant motion. This motivates us to design a latent action model that can disentangle interaction dynamics from observer-induced variations (Wei et al., 2026; Lin et al., 2026).

Moreover, we also need to account for differences in **execution dynamics** between human and robot interactions. Even when meaningful interaction structures are captured, human and robot behaviors can differ in how the same interaction is executed due to differences in embodiment, execution speed, and control granularity. Such discrepancies are particularly pronounced in fine-grained temporal patterns, which can vary substantially even when the underlying interaction semantics are shared. We therefore analyze the temporal structures of human egocentric interactions and robot actions in the frequency domain. Our analysis reveals substantially stronger correspondence in the low-frequency components of semantically matched human-robot interaction pairs. Motivated by this observation, we treat the low-frequency components of latent actions as transferable interaction information and use them as guidance for robot policy learning and action generation.

Motivated by these observations, we introduce **WING** (**W**orld Action Learning via **IN**teraction-Centric Spectral Latent **G**uidance), a framework for extracting transferable interaction dynamics from large-scale egocentric videos and leveraging them for robot learning. WING first learns interaction-centric latent actions by suppressing sensitivity to observer motion while preserving interaction-relevant changes. Specifically, we decompose visual transitions into global observer-induced motion and local interaction-related motion, and selectively distill the latter into the latent action representation. WING then decomposes latent action trajectories in the frequency domain using the discrete cosine transform (DCT) (Ahmed et al., 1974) and retains low-frequency components that exhibit more consistent structure across human and robot executions. At inference time, a lightweight predictor estimates this spectral guidance from the current observation and task context, which is then used to condition robot action generation.

We conduct extensive evaluations of both the representations learned by WING-LAM and the downstream performance of the complete WING framework. Our analysis shows that WING-LAM is substantially less sensitive to observer motion while preserving interaction-related structure, and that low-frequency latent dynamics exhibit stronger correspondence between semantically matched human and robot executions. We further evaluate WING across multiple simulated benchmarks and real-world manipulation tasks. In simulation, WING achieves average success rates of 99.2% on

LIBERO (Liu et al., 2023) and 93.80% on RoboTwin 2.0 (Chen et al., 2025), and reaches 57.7% on RoboCasa–GR1 (Nasiriany et al., 2024) after pre-training on egocentric video data, outperforming strong baselines including $\pi_{0.5}$ (Intelligence et al., 2025) and LingBot-VA (Li et al., 2026b). In real-world experiments, we evaluate WING on four bimanual manipulation tasks under both standard and diverse generalization settings, where it maintains strong performance across variations. Ablation studies further confirm that both interaction-centric latent action learning and low-frequency spectral guidance contribute consistently to downstream policy performance.

Overall, our contributions are threefold:

- We show that observer-induced variations can become entangled with interaction-relevant dynamics in egocentric latent actions, and further reveal that low-frequency latent dynamics exhibit stronger correspondence across human and robot interactions.
- Motivated by these findings, we introduce **WING**, which combines interaction-centric latent action learning with spectral latent guidance to suppress observer-induced variations and exploit temporal structures that are more consistently shared across embodiments.
- We validate WING through extensive representation analyses and closed-loop experiments across simulated and real-world manipulation tasks. WING consistently outperforms representative robot policies and alternative egocentric pretraining strategies.

2 Related Work

Recent vision-language-action (VLA) and world-action models (WAMs) have demonstrated strong generalization by combining large-scale vision-language or video pretraining with robot demonstrations (Brohan et al., 2023; Kim et al., 2024; Black et al., 2024; Intelligence et al., 2025; NVIDIA et al., 2025; Ye et al., 2026b; Yuan et al., 2026b; Li et al., 2026b). However, their performance still relies heavily on robot data, which remains expensive to collect and difficult to scale across diverse tasks and environments. To alleviate this limitation, recent studies have explored egocentric human videos as a scalable source of interaction experience. Prior works have explored this direction through representation learning, behavior transfer, and latent-action modeling (Nair et al., 2022; Ma et al., 2022; Kareer et al., 2025; Zheng et al., 2026b; Ye et al., 2025; Bu et al., 2025; Wei et al., 2026; Xu et al., 2026). However, ego-centric transitions entangle interaction dynamics with observer-induced camera motion, making it difficult to identify transferable manipulation structure. Existing approaches either model full visual transitions or reduce action-irrelevant variation, while other works exploit camera motion as an active-perception cue (Wei et al., 2026; Xu et al., 2026; Lin et al., 2026), without explicitly separating observer-specific motion from interaction dynamics.

WING addresses this gap by learning interaction-centric latent action guidance for ego-to-robot transfer. It disentangles observer motion from interaction dynamics and extracts low-frequency temporal structure that is more consistently shared between human and robot executions. A more comprehensive discussion of related work is provided in Appendix B.

3 Methods

In this section, we present the technical details of **WING**, including interaction-centric latent action learning, the spectral latent guidance module, and the overall training pipeline. Together, these components extract transferable interaction structure from large-scale egocentric videos and use it as structured latent guidance for robot action learning.

3.1 Overview

A robot policy $\pi_\theta(\mathbf{a}_{t:t+m} \mid \mathbf{l}, \mathbf{o}_t, \mathbf{s}_t)$ maps a language instruction $\mathbf{l}$, visual observation $\mathbf{o}_t$, and robot proprioceptive state $\mathbf{s}_t$ to an executable action chunk $\mathbf{a}_{t:t+m}$. Our goal is to learn generalizable action knowledge from large-scale egocentric videos through latent action modeling and transfer it to downstream robot manipulation policies. As discussed above, this transfer faces two key challenges: (1) reconstruction-based latent action models entangle interaction-relevant dynamics with task-irrelevant motion, particularly ego-camera motion; and (2) discrepancies in temporal dynamics between human interactions and robot actions hinder direct cross-embodiment transfer.

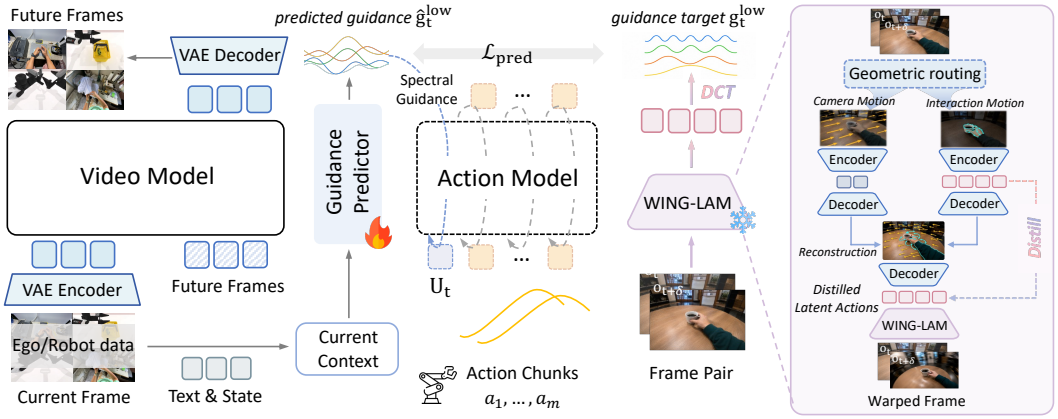


Figure 2: **Overview of WING.** WING first learns interaction-centric latent actions that reduce observer-induced variation while preserving interaction dynamics. It then extracts low-frequency temporal structure with stronger correspondence across human and robot executions and predicts this structure from the current context as guidance for robot action generation.

To address these challenges, **WING** comprises two key components. First, **WING-LAM** is an interaction-centric latent action model that encodes egocentric visual transitions $(\mathbf{o}_t^{\text{ego}}, \mathbf{o}_{t+\delta}^{\text{ego}})$ into latent actions $\mathbf{z}_t = E_\phi^{\text{WING-LAM}}(\mathbf{o}_t^{\text{ego}}, \mathbf{o}_{t+\delta}^{\text{ego}})$ while suppressing task-irrelevant motion, thereby encouraging the latent representation to capture interaction-relevant dynamics. Second, the **Spectral Latent Guidance Module** decomposes latent action trajectories $\mathbf{z}_{t:t+H}$ in the spectral domain and retains the first K low-frequency components, $\mathbf{g}_t^{\text{low}} = [\text{DCT}(\mathbf{z}_{t:t+H})]_{0:K-1}$, as transferable latent guidance. By suppressing rapidly varying high-frequency components, this module isolates slowly varying temporal structures that are more consistently shared across human and robot interactions.

At inference time, a lightweight predictor P_ψ estimates the spectral guidance $\hat{\mathbf{g}}_t^{\text{low}} = P_\psi(\mathbf{l}, \mathbf{o}_t, \mathbf{s}_t)$, which conditions the world-action model $\pi_\theta(\mathbf{a}_{t:t+m} \mid \mathbf{l}, \mathbf{o}_t, \mathbf{s}_t, \hat{\mathbf{g}}_t^{\text{low}})$ for fine-grained robot action generation. In this way, WING transfers knowledge learned from egocentric human videos as a high-level action prior, while allowing the robot policy to generate embodiment-specific fine-grained controls, thereby bridging large-scale human interaction and downstream robot manipulation.

3.2 Learning Interaction-Centric Latent Actions

Egocentric videos contain both observer-induced motion and hand-object interactions. Consequently, a latent action learned to reconstruct the full transition can encode both sources of change (Zhang et al., 2025). However, for robot guidance, this calls for the latent to retain interaction dynamics while reducing its sensitivity to observer motion. **WING-LAM** therefore uses an explicit motion-routing design that isolates interaction dynamics from observer motion and distills the resulting interaction latent into an RGB-only student.

Camera–Interaction Motion Decomposition. Observer motion and hand–object interaction exhibit different spatial motion patterns in egocentric videos: observer motion induces coherent image-wide changes, while interaction motion is more localized. We exploit this distinction to construct motion-aware supervision. Specifically, we decompose each frame transition into a global observer-motion component and a local interaction-motion component. Given two video frames $\mathbf{o}_t$ and $\mathbf{o}_{t+\delta}$, an offline point tracker estimates point correspondences: $\mathbf{p}_{t,i}, \mathbf{p}_{t+\delta,i} \in \mathbb{R}^2$, where i identifies the same tracked point across frames. Training-time region masks separate background tracks from hand–object interaction tracks. Using the background tracks, we estimate an invertible global image warp $W_{t\to t+\delta}$ as the observer-motion target, capturing the dominant coherent transformation between the two frames. After compensating this global motion, the remaining displacement of each point represents motion unexplained by the observer movement:

$$\mathbf{r}_{t,i} = \underbrace{W_{t\to t+\delta}^{-1}(\mathbf{p}_{t+\delta,i})}_{\text{camera-compensated location}} - \mathbf{p}_{t,i}, \tag{1}$$

where $\mathbf{r}_{t,i} \in \mathbb{R}^2$ denotes the residual displacement expressed in the first frame's coordinates, which serves as the interaction-motion target.

Motion-Routed Latent Learning. These decomposed motion signals provide privileged supervision for learning a factorized latent representation. We then train a two-branch teacher to encode visual features and tracked point coordinates, producing camera and interaction latents, $\mathbf{z}^T_{t,\mathrm{cam}} \in \mathbb{R}^{D_c}$ and $\mathbf{z}^T_{t,\mathrm{int}} \in \mathbb{R}^{D_i}$, where the superscript T denotes the teacher and D_c, D_i are the respective latent dimensions. The camera branch predicts the global warp, while the interaction branch predicts residual displacements, and their decoded motions jointly reconstruct the tracked locations in the second frame. Although motion routing separates the two sources, observer-specific cues may still leak into the interaction latent; we therefore apply camera interventions that preserve interaction dynamics and enforce latent consistency.

Interaction Distillation. Although the teacher learns interaction-centric representations using privileged motion cues, these cues are unavailable during deployment. We therefore distill the teacher's interaction latent into an RGB-only student S_ϕ that directly extracts latent actions from frame pairs. For each pair $(\mathbf{o}^{\mathrm{ego}}_t, \mathbf{o}^{\mathrm{ego}}_{t+\delta})$, we construct a camera-perturbed view by applying a camera-like global warp to the second frame, yielding $\tilde{\mathbf{o}}^{\mathrm{ego}}_{t+\delta}$. The two pairs share the same interaction target $\mathbf{z}^T_{t,\mathrm{int}}$ while differing only in observer-induced motion. The student then encodes the original and augmented pairs as: $\mathbf{z}^{\mathrm{nat}}_t = S_\phi(\mathbf{o}^{\mathrm{ego}}_t, \mathbf{o}^{\mathrm{ego}}_{t+\delta})$ and $\mathbf{z}^{\mathrm{aug}}_t = S_\phi(\mathbf{o}^{\mathrm{ego}}_t, \tilde{\mathbf{o}}^{\mathrm{ego}}_{t+\delta})$, where both outputs lie in $\mathbb{R}^{D_i}$, and nat and aug denote the original and augmented inputs, respectively. We optimize the student using:

$$\mathcal{L}_{\mathrm{distill}} = \frac{1}{2D_i} \sum_{v \in \{\mathrm{nat},\mathrm{aug}\}} \left\| \mathbf{z}^v_t - \mathbf{z}^T_{t,\mathrm{int}} \right\|_2^2 + \frac{\lambda_{\mathrm{view}}}{D_i} \left\| \mathbf{z}^{\mathrm{nat}}_t - \mathbf{z}^{\mathrm{aug}}_t \right\|_2^2, \tag{2}$$

where v indexes the two inputs and λ_{view} weights their prediction consistency. The first term transfers the teacher's interaction representation; the second encourages the student to retain that representation under camera-like image warps. Details of implementation are provided in Appendix C.1.

Overall, the above design separates observer-induced variation from interaction dynamics, yielding interaction-centric latent actions that capture transferable manipulation structure and serve as the foundation for the subsequent spectral guidance module.

3.3 Spectral Latent Action Guidance

Although WING-LAM extracts interaction-centric latent actions, semantically matched human and robot executions can differ in temporal dynamics. We find that low-frequency latent components exhibit stronger cross-embodiment correspondence than full trajectories, and therefore retain these components as spectral guidance for robot action generation.

Spectral Latent Representation. Given visual observations sampled at a fixed interval Δt, WING-LAM encodes H successive transitions into $\mathbf{Z}_t = [\mathbf{z}_t, \ldots, \mathbf{z}_{t+H-1}]^\top \in \mathbb{R}^{H \times D_i}$, where $\mathbf{z}_{t+h} = S_\phi(\mathbf{o}_{t+h}, \mathbf{o}_{t+h+1})$. An orthonormal DCT is then applied along the temporal dimension, and the lowest K components are retained as the guidance target $\mathbf{g}^{\mathrm{low}}_t$. This truncation affects only the latent guidance, not the robot actions.

Predicting Low-Frequency Guidance. Since $\mathbf{g}^{\mathrm{low}}_t$ requires future observations, we train a guidance predictor P_ψ to estimate it from the current observation and instruction, together with proprioception when available. Writing these inputs as $\mathbf{c}_t$, we minimize the mean squared prediction error:

$$\widehat{\mathbf{g}}^{\mathrm{low}}_t = P_\psi(\mathbf{c}_t), \qquad \mathcal{L}_{\mathrm{pred}} = \frac{1}{KD_i} \left\| \widehat{\mathbf{g}}^{\mathrm{low}}_t - \mathbf{g}^{\mathrm{low}}_t \right\|_F^2, \tag{3}$$

where $\| \cdot \|_F$ denotes the Frobenius norm. During robot training, we jointly optimize this loss and the action-generation objective. Future frames are only used to construct training targets.

Conditioning Robot Actions. We convert the predicted guidance $\widehat{\mathbf{g}}^{\mathrm{low}}_t$ into $\mathbf{U}_t \in \mathbb{R}^{K \times D_{\mathrm{hid}}}$, where D_{hid} is the action model's hidden dimension. Each frequency coefficient vector is projected to this dimension, normalized with LayerNorm, and given a learned embedding identifying its frequency mode. The action model attends to these guidance tokens through a gated residual update:

$$\mathbf{H}^{(l)}_a \leftarrow \mathbf{H}^{(l)}_a + \gamma_l \, \mathrm{Attn}\Big(Q = \mathrm{LayerNorm}(\mathbf{H}^{(l)}_a), K = \mathbf{U}_t, V = \mathbf{U}_t\Big), \tag{4}$$

where $\mathbf{H}^{(l)}_a$ denotes the action hidden sequence at layer l, and the learned scalar γ_l controls the magnitude of latent guidance.

3.4 Training Strategy

We train the world action model (WAM) in two stages: pretraining on egocentric videos, followed by robot policy learning from robot demonstrations.

Egocentric Pretraining. Using our curated egocentric data, we adopt Wan2.2 (Wan et al., 2025) and train the video expert for future-frame generation and the guidance predictor P_ψ for low-frequency spectral guidance prediction. The two pathways learn visual and interaction dynamics through the generation objective and latent guidance prediction, respectively. Full details of egocentric pretraining are provided in Appendix C.2.

Robot Policy Learning. After pretraining, we transfer the pretrained visual backbone and guidance predictor to the robot policy and jointly train them with the action expert on robot demonstrations. The guidance predictor is supervised on robot data, while the WING-LAM encoder remains frozen. The predicted guidance conditions downstream robot action generation during both training and inference. This stage adapts the interaction dynamics learned from egocentric videos to robot control, with robot demonstrations providing direct supervision for executable actions.

4 Experiments

In this section, we evaluate WING from both representation and policy perspectives. Specifically, we aim to answer the following four questions:

- **(Q1)** What structures in egocentric latent actions reveal transferable interaction knowledge?
- **(Q2)** Can WING effectively transfer latent action knowledge to improve robot performance?
- **(Q3)** How do interaction-centric latent actions and spectral latent action guidance contribute?
- **(Q4)** How effective is WING's pretraining strategy for downstream robot policy learning?

4.1 Experimental Setup

We first evaluate whether WING-LAM reduces observer-related variation while retaining interaction-relevant semantics, and then investigate which temporal structures in latent trajectories exhibit stronger cross-embodiment correspondence between human and robot executions. We compare WING-LAM with representative LAM baselines using motion readout, camera perturbations, and action classification, and analyze semantic similarity between matched and mismatched human–robot trajectories across frequency bands. Evaluation details are provided in Appendix D.1 and Appendix D.2. We further evaluate WING on LIBERO (Liu et al., 2023), RoboTwin 2.0 (Chen et al., 2025), RoboCasa–GR1 (Nasiriany et al., 2024), and a real-world dual-arm platform. We compare against representative vision-language-action, latent-action, and world-action modeling methods, reporting success rate in simulation and both task completion and progress in the real world. Full evaluation protocols are provided in Appendix D.

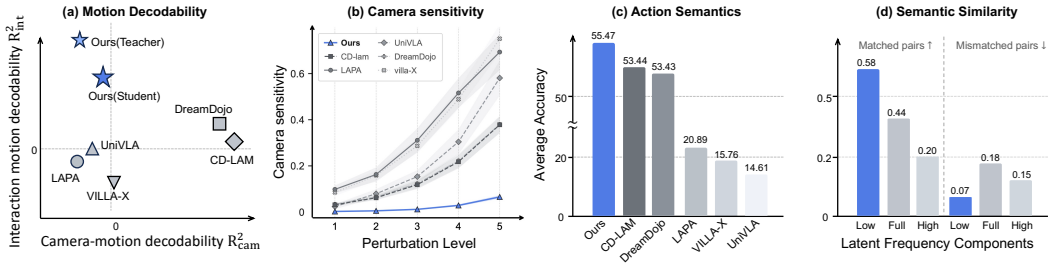


Figure 3: **Characterizing interaction structure in egocentric latent actions.** **(a)** Camera-motion and interaction-motion decodability measured by MLP-probe R^2. **(b)** Normalized latent sensitivity under progressively stronger camera-motion perturbations. **(c)** Average action-category classification accuracy using frozen latent representations across the LARYBench (Nie et al., 2026) egocentric datasets. **(d)** Cross-embodiment semantic similarity of low-frequency, high-frequency, and full latent-action trajectories for semantically matched and mismatched human–robot pairs.

Table 1: **Simulation results on LIBERO and RoboTwin 2.0.** Success rate (%). The best and second-best results in each column are shown in **bold** and underline, respectively.

(a) LIBERO

Method	Spatial	Object	Goal	Long	Avg.
VLAs					
π_0 (Black et al., 2024)	98.00	96.80	94.40	88.40	94.40
$\pi_{0.5}$ (Intelligence et al., 2025)	98.80	98.20	98.00	92.40	96.90
Latent-Action Based Methods					
LAPA (Ye et al., 2025)	73.80	74.60	58.80	55.40	65.70
UniVLA (Bu et al., 2025)	96.50	96.80	95.60	92.00	95.20
LaWAM (Chen et al., 2026a)	**99.40**	99.60	98.40	97.00	98.60
VLA-JEPA (Sun et al., 2026)	96.20	99.60	97.20	95.80	97.20
WAMs					
Motus (Bi et al., 2026)	96.80	99.80	96.60	97.60	97.70
LingBot-VA (Li et al., 2026b)	98.50	99.60	97.20	98.50	98.50
Fast-WAM (Yuan et al., 2026b)	98.20	**100.00**	97.00	95.20	97.60
Ours	99.00	**100.00**	**98.80**	**99.00**	**99.20**

(b) RoboTwin 2.0

Method	Clean	Rand.	Avg.
VLAs			
π_0 (Black et al., 2024)	65.92	58.40	62.16
$\pi_{0.5}$ (Intelligence et al., 2025)	82.74	76.76	79.75
X-VLA (Zheng et al., 2026a)	72.90	72.80	72.85
LingBot-VLA (Wu et al., 2026)	88.56	86.68	87.62
Latent-Action Based Methods			
LaWAM (Chen et al., 2026a)	92.64	89.80	91.22
WAMs			
Motus (Bi et al., 2026)	88.66	87.02	87.84
Fast-WAM (Yuan et al., 2026b)	91.88	91.78	91.83
GigaWorld-Policy (Ye et al., 2026a)	86.36	85.04	85.70
LingBot-VA (Li et al., 2026b)	92.90	91.50	92.20
Ours	**94.56**	**93.04**	**93.80**

4.2 **(Q1):** Characterizing Egocentric Latent Structure For Robot Learning.

We first examine whether WING-LAM suppresses observer-related variation while preserving interaction semantics. As shown in Figure 3(a), both its teacher and student retain strong interaction-motion information but substantially less camera-motion information than DreamDojo (Gao et al., 2026). Notably, although CD-LAM (Wei et al., 2026) is explicitly designed to reduce action-irrelevant variation, it still retains more camera-motion information under our evaluation protocol. This selectivity persists under direct perturbation: as shown in Figure 3(b), the student exhibits the lowest latent sensitivity under increasing camera motion. Importantly, this robustness does not come at the expense of action semantics, as WING-LAM achieves the highest average action-classification accuracy on LARYBench (Nie et al., 2026) (Figure 3(c)). We then ask which temporal components of these interaction-centric trajectories are most consistent across human and robot executions. As shown in Figure 3(d), low-frequency components yield a substantially larger separation between semantically matched and mismatched pairs than either the full trajectory or the high-frequency components, indicating that shared interaction structure is concentrated more strongly in low-frequency dynamics and motivating their use as guidance for action generation.

4.3 **(Q2):** Robot Policy Performance

Results on Simulation. Across single-arm, bimanual, and humanoid manipulation benchmarks, WING achieves average success rates of 99.20% on LIBERO (Table 1(a)), 93.80% on RoboTwin 2.0 (Table 1(b)), and 57.7% on RoboCasa–GR1 (Table 2), outperforming the strongest baselines on all three. Notably, its advantage extends to LIBERO's long-horizon suite and RoboTwin's randomized scenes, suggesting that the gains are not confined to shorter tasks or clean visual conditions. Together, these results demonstrate the effectiveness of WING for robot manipulation across diverse embodiments and task settings. Full results for RoboTwin 2.0 and RoboCasa are provided in Appendices E.4.2 and E.4.1, respectively.

Table 2: **RoboCasa–GR1 Tabletop.** Best in **bold**; second-best underlined. † denotes our reproduced results.

Method	Avg. SR
GR00T-N1.6	47.6
StarVLA-OFT	48.8
DiT4DiT	50.8
LDA-1B	55.4
Being-H0.7	49.2
FastWAM†	51.9
Ours	**57.7**

Results on the real-world robot. As illustrated in Figure 4, WING achieves the highest average success rate of 75.0% under the standard setting. Across both standard and generalization settings, WING consistently outperforms FastWAM and GR00T-N1.7 in average success rate and progress score. Under generalization settings, WING achieves comparable average success to the strong baseline $\pi_{0.5}$, while attaining a slightly higher average progress score. These results demonstrate the effectiveness of WING in real-world manipulation and its ability to maintain strong performance under diverse generalization settings.

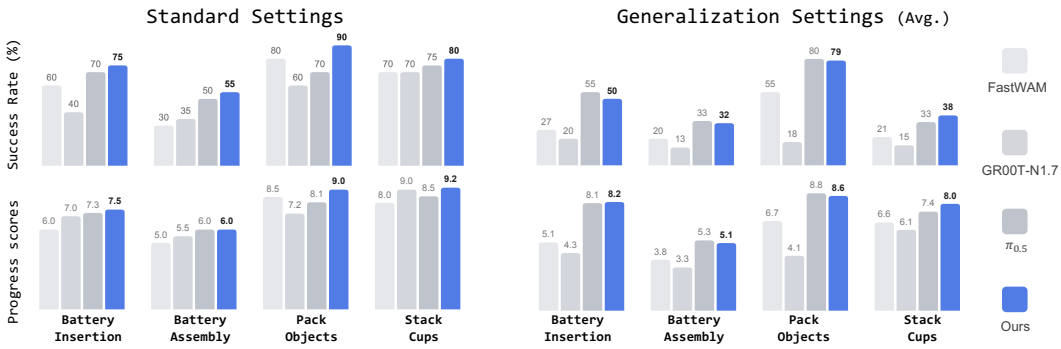


Figure 4: **Real-world task evaluation results.** We evaluate different robot foundation models on both standard and generalization settings across four real-world manipulation tasks. Our method achieves consistently strong performance, achieving the strongest standard-setting performance and remaining competitive under generalization.

4.4 **(Q3):** Ablation Studies on WING-LAM and Spectral Guidance.

Table 3: **Ablation on LIBERO and Real-World.** We report SR on these tasks.

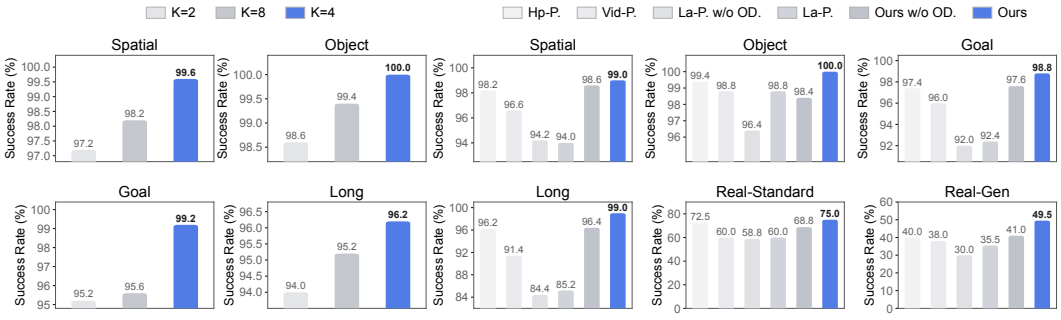
Method		LIBERO				Real-World	
WING-LAM	DCT	Spatial	Object	Goal	Long	Base	Gen.
✗	✗	99.2	97.2	96.2	92.6	65.0	36.0
✓	✗	98.0	99.2	95.6	94.2	67.5	40.3
✗	✓	99.2	98.2	98.0	96.2	71.3	43.0
✓	✓	**99.6**	**100.0**	**99.2**	**96.2**	**72.5**	**45.0**

Ablation on observer debiasing and spectral guidance. Table 3 evaluates the contribution of both components under a controlled setting *without ego-pretraining of the WAM*. When WING-LAM is ablated, we replace it with the pretrained DreamDojo LAM while keeping the spectral guidance unchanged. When DCT module variant is ablated, we retain the same latent encoder but bypass DCT and use the full time-domain latent trajectory directly as guidance. Removing either component reduces average performance across both LIBERO and real-world evaluations. The gains from the DCT variant show that frequency-domain decomposition and low-frequency retention yield more effective guidance by emphasizing slowly varying interaction dynamics. The additional gains from WING-LAM indicate that latent actions with reduced observer-induced variation provide more useful guidance for robot control. This advantage of WING-LAM becomes even clearer in the ego-pretraining analysis below, where it leads to larger improvements.

Ablation on the number of DCT components. We further investigate how much spectral information is required for effective latent guidance by varying the number of retained DCT coefficients $K \in \{2, 4, 8\}$, where $K = 8$ corresponds to the full spectrum. As shown in Figure 5(a), $K = 4$ consistently performs best across all suites. Retaining only two coefficients is overly restrictive and discards useful temporal dynamics, while the full spectrum introduces additional high-frequency components without improving downstream control. These results suggest that effective latent guidance requires a sufficient, but not excessive, spectral bandwidth, with $K = 4$ providing the best balance between preserving control-relevant temporal structure and filtering variation that is less consistently shared across human and robot executions.

4.5 **(Q4):** Analysis of Ego Pre-training Strategies.

Having established WING's downstream performance, we further examine whether its interaction-centric spectral guidance also provides an effective pre-training strategy for WAMs from egocentric videos. We compare pre-training methods under the same 200h budget which differ in how video-derived information supervises the WAM. In Figure 5(b), **Vid-P.** performs video-only pre-training, isolating the effect of exposure to egocentric videos themselves. **HP-P.** uses hand-pose labels as supervision. **LA-P.** follows LAPA (Ye et al., 2025) which uses latent actions as direct action-prediction targets. In contrast, **Ours** predicts spectral latent guidance to condition action generation. To isolate the effect of suppressing observer-induced variation, **w/o OD.** denotes variants in which WING-LAM is replaced with DreamDojo LAM. Implementation details are provided in Appendix C.2.



(a) Ablation on DCT components (K). (b) Pretraining ablation across six suites.

Figure 5: **Analysis of spectral guidance and egocentric pretraining strategies. (a)** Effect of the number of retained DCT components K on latent guidance. **(b)** Comparison of different egocentric pretraining strategies including video-only pretraining, hand-pose supervision, direct latent-action pretraining, and spectral latent guidance, with additional variants to evaluate the effect of interaction-centric debiasing, across LIBERO and real-world tasks.

As shown in Figure 5(b), video-only pre-training consistently underperforms our method, with particularly clear gaps on the real-world standard and generalization settings, indicating that the gains of egocentric pre-training cannot be explained by video exposure alone. Replacing WING-LAM with a non-debiased latent-action model further degrades performance in most settings, supporting the importance of suppressing observer-induced variation. Direct latent-action prediction also performs worse, possibly reflecting the gap between visual transition representations and executable robot actions. Together, these results suggest that WING is most effective when ego-derived latent actions serve as auxiliary guidance, rather than as direct action targets, while the robot policy retains its original action-prediction objective. We further analyze the effectiveness and efficiency of pre-training in Appendix E.3.3 and E.3.4 by examining downstream validation loss after different amounts of egocentric pretraining and downstream performance under varying amounts of robot training data. These analyses show that WING not only learns effective representations during pre-training, but also retains stronger downstream performance as robot supervision becomes more limited.

5 Limitations and Future Work

Although WING provides an effective way to exploit egocentric videos for robot learning, several limitations remain. First, constructing the WING-LAM teacher relies on external motion cues and an image-space camera–interaction decomposition. The dominant observer motion is approximated by a single global affine warp, which can become less accurate under strong parallax, large depth variation, or pronounced 3D camera motion, potentially introducing noise into the interaction supervision. Second, latent-action pretraining and spectral guidance add training cost and pipeline complexity. Future work could explore more self-contained, depth-aware motion modeling and more efficient guidance mechanisms.

6 Conclusion

In this work, we study how to effectively transfer interaction knowledge from egocentric human videos to robot learning. We show that transferable latent actions require suppressing observer-induced variation and capturing temporal structures that remain consistent across human and robot executions. Based on these findings, we propose **WING**, a framework that learns and transfers interaction-centric latent knowledge from egocentric videos. WING suppresses observer-induced variations through interaction-centric latent action learning and leverages low-frequency latent dynamics that are more consistent across embodiments through spectral latent guidance. Extensive experiments across simulation and real-world manipulation benchmarks demonstrate that WING effectively transfers embodied interaction knowledge from egocentric videos to robot control, achieving strong performance and generalization across diverse tasks. Our results suggest that scaling robot learning with human experience requires not simply modeling all observed dynamics, but identifying interaction structures that remain consistent across observers and embodiments.

AI USE STATEMENT

We used generative AI tools for language polishing, improving manuscript readability, discussing research methodology and experimental design, and assisting with minor code utilities and debugging. All AI-assisted outputs were reviewed and verified by the authors. The authors take full responsibility for all contents, claims, and artifacts in this work.

ETHICS STATEMENT

This work uses publicly available egocentric video datasets and established robot-learning benchmarks in accordance with their respective terms of use. No new human-subject study or personally identifiable dataset was collected as part of this work. Real-robot experiments were conducted in a controlled laboratory setting on manipulation tasks. As with other learned robot policies, the resulting models may exhibit unpredictable behavior outside the evaluated settings and should not be interpreted as validated for safety-critical or unrestricted deployment around humans.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the proposed method, training procedure, data processing and downstream policy learning in the main paper and Appendix. We plan to release the training, evaluation, and real-robot deployment code upon publication.

REFERENCES

Nasir Ahmed, T. Natarajan, and K. R. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, 1974.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. In *CVPR*, 2026.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.

Jialei Chen, Kai Wang, Kang Chen, Shuaihang Chen, Feng Gao, Wenhao Tang, Zhiyuan Li, Weilin Liu, Zhuyu Yao, Boxun Li, Yuanbo Xu, and Chao Yu. LaWAM: Latent world action models for efficient dynamics-aware robot policies. *arXiv preprint arXiv:2606.15768*, 2026a.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

Xiaoyu Chen, Hangxing Wei, Pushi Zhang, Chuheng Zhang, Kaixin Wang, Yanjiang Guo, Rushuai Yang, Yucen Wang, Xinquan Xiao, Li Zhao, et al. Villa-x: enhancing latent action modeling in vision-language-action models. In *ICLR*, 2026b.

StarVLA Community. Starvla: A lego-like codebase for vision-language-action model developing. *arXiv preprint arXiv:2604.05014*, 2026.

Weisheng Dai, Kai Lan, Jianyi Zhou, Bo Zhao, Xiu Su, Junwen Tong, Weili Guan, and Shuo Yang. Conla: Contrastive latent action learning from human videos for robotic manipulation. *arXiv preprint arXiv:2602.00557*, 2026.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. The epic-kitchens dataset: Collection, challenges and baselines. *TPAMI*, 2021.

Yufan Deng and Daquan Zhou. Humannet: Scaling human-centric video learning to one million hours. *arXiv preprint arXiv:2605.06747*, 2026.

Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981.

Zipeng Fu, Qingqing Zhao, Qi Wu, Gordon Wetzstein, and Chelsea Finn. Humanplus: Humanoid shadowing and imitation from humans. *arXiv preprint arXiv:2406.10454*, 2024.

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, et al. Dreamdojo: A generalist robot world model from large-scale human videos. *arXiv preprint arXiv:2602.06949*, 2026.

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

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022.

Adam W. Harley, Yang You, Xinglong Sun, Yang Zheng, Nikhil Raghuraman, Yunqi Gu, Sheldon Liang, Wen-Hsuan Chu, Achal Dave, Pavel Tokmakov, Suya You, Rares Ambrus, Katerina Fragkiadaki, and Leonidas J. Guibas. Alltracker: Efficient dense point tracking at high resolution. In *ICCV*, 2025.

Ryan Hoque, Peide Huang, David Yoon, Jian Zhang, et al. Egodex: Learning dexterous manipulation from large-scale egocentric video. In *ICLR*, 2026.

Bing Hu, Zaijing Li, Rui Shao, Junda Chen, April Hua Liu, Wei-Shi Zheng, and Liqiang Nie. From abstraction to instantiation: Learning behavioral representation for vision-language-action model. *arXiv preprint arXiv:2605.22671*, 2026.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. Egomimic: Scaling imitation learning via egocentric video. In *ICRA*, 2025.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, et al. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026.

Jung Min Lee, Taehyun Cho, Li Zhao, and Jungwoo Lee. Why latent actions fail, and how to prevent it. *arXiv preprint arXiv:2605.20223*, 2026.

Hao Li, Ganlong Zhao, Yufei Liu, Haotian Hou, Guoquan Ye, Tongyan Fang, Chunxiao Liu, Siyuan Huang, Jianbo Liu, Xiaogang Wang, et al. Ace-ego-0: Unifying egocentric human and robotic data for vla pretraining. *arXiv preprint arXiv:2606.17200*, 2026a.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, et al. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026b.

Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024.

Xingyao Lin, Guojin Zhong, Tianyi Lu, Ziyi Ye, Yichen Zhu, Zuxuan Wu, and Yu-Gang Jiang. Activemimic: Egocentric video pretraining with active perception. *arXiv preprint arXiv:2606.06194*, 2026.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *NeurIPS*, 2023.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. In *ICLR*, 2025.

Hao Luo, Wanpeng Zhang, Yicheng Feng, Sipeng Zheng, Haiweng Xu, Chaoyi Xu, Ziheng Xi, Yuhui Fu, and Zongqing Lu. Being-h0.7: A latent world-action model from egocentric videos. *arXiv preprint arXiv:2605.00078*, 2026.

Jiangran Lyu, Kai Liu, Xuheng Zhang, Haoran Liao, Yusen Feng, Wenxuan Zhu, Tingrui Shen, Jiayi Chen, Jiazhao Zhang, Yifei Dong, et al. Lda-1b: Scaling latent dynamics action model via universal embodied data ingestion. *arXiv preprint arXiv:2602.12215*, 2026.

Teli Ma, Jia Zheng, Zifan Wang, Chunli Jiang, Andy Cui, Junwei Liang, and Shuo Yang. DiT4DiT: Jointly modeling video dynamics and actions for generalizable robot control. *arXiv preprint arXiv:2603.10448*, 2026.

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022.

Yecheng Jason Ma, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. Liv: Language-image representations and rewards for robotic control. In *ICML*, 2023.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. RoboCasa: Large-scale simulation of everyday tasks for generalist robots. *arXiv preprint arXiv:2406.02523*, 2024.

Dujun Nie, Fengjiao Chen, Qi Lv, Jun Kuang, Xiaoyu Li, Xuezhi Cao, and Xunliang Cai. Lary: A latent action representation yielding benchmark for generalizable vision-to-action alignment. *arXiv preprint arXiv:2604.11689*, 2026.

Yichen Niu, Haoran Lv, Xinrui Zhang, Xueyao Wan, Shiyu Gao, Ying Ai, Hui Xu, Yongqi Hu, Hengyi Zhang, Yang Xie, et al. Egoaero: Learning dexterous manipulation from a single egocentric video without object assets. *arXiv preprint arXiv:2606.08057*, 2026.

NVIDIA, Johan Bjorck, Nikita Cherniadev Fernando Castañeda, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An open foundation model for generalist humanoid robots. In *ArXiv Preprint*, 2025.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models. In *ICRA*, 2024.

Ryan Punamiya, Simar Kareer, Zeyi Liu, Josh Citron, Ri-Zhao Qiu, Xiongyi Cai, Alexey Gavryushin, Jiaqi Chen, Davide Liconti, Lawrence Y Zhu, et al. Egoverse: An egocentric human dataset for robot learning from around the world. *arXiv preprint arXiv:2604.07607*, 2026.

Junyao Shi, Zhuolun Zhao, Tianyou Wang, Ian Pedroza, Amy Luo, Jie Wang, Jason Ma, and Dinesh Jayaraman. Zeromimic: Distilling robotic manipulation skills from web videos. In *ICRA*, 2025.

Quanxin Shou, Fangqi Zhu, Shawn Chen, Puxin Yan, Zhengyang Yan, Yikun Miao, Xiaoyi Pang, Zicong Hong, Ruikai Shi, Hao Huang, et al. Halo: A unified vision-language-action model for embodied multimodal chain-of-thought reasoning. *arXiv preprint arXiv:2602.21157*, 2026.

Gilbert Strang. The discrete cosine transform. *SIAM Review*, 1999.

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. Vla-jepa: Enhancing vision-language-action model with latent world model. *arXiv preprint arXiv:2602.10098*, 2026.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

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

Xin Wang, Taein Kwon, Mahdi Rad, Bowen Pan, Ishani Chakraborty, Sean Andrist, Dan Bohus, Ashley Feniello, Bugra Tekin, Felipe Vieira Frujeri, et al. Holoassist: an egocentric human interaction dataset for interactive ai assistants in the real world. In *ICCV*, 2023.

Ye Wang, Pei Lin, Xiong-Hui Chen, Haoqi Yuan, Zhixuan Liang, Yiyang Huang, Anzhe Chen, Zixing Lei, Jie Zhang, Tao Zhang, et al. Ego2robot: Scalable robot data synthesis from egocentric human data. *arXiv preprint arXiv:2608.02580*, 2026.

Yufan Wei, Kun Zhou, Lingjun Mao, Zijun Zhang, Ziming Xu, Ziqiao Xi, Shuang Liang, Ruobing Han, Yuchen Yan, Xinyue Wang, et al. Causally debiased latent action model for embodied action conditioned world models. *arXiv preprint arXiv:2607.09185*, 2026.

Wei Wu, Fan Lu, Yunnan Wang, Shuai Yang, Shi Liu, Fangjing Wang, Qian Zhu, He Sun, Yong Wang, Shuailei Ma, et al. A pragmatic vla foundation model. *arXiv preprint arXiv:2601.18692*, 2026.

Runze Xu, Yiluo Zhang, Jian Wang, Yu Wang, and Jincheng Yu. Motion-focused latent action enables cross-embodiment vla training from human egovideos. *arXiv preprint arXiv:2606.18955*, 2026.

Ge Yan, Jinghao Liu, Yuzhi Fan, Lei Cai, Minwen Liao, Jesse Zhang, and Dieter Fox. Flex-π: A multi-stream world-action model with compute flexibility. *arXiv preprint arXiv:2608.10860*, 2026.

Jiange Yang, Yansong Shi, Haoyi Zhu, Mingyu Liu, Kaijing Ma, Yating Wang, Gangshan Wu, Tong He, and Limin Wang. Como: Learning continuous latent motion from internet videos for scalable robot learning. In *CVPR*, 2026.

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

Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, Jie Li, Jindi Lv, Jingyu Liu, et al. Gigaworld-policy: An efficient action-centered world–action model. *arXiv preprint arXiv:2603.17240*, 2026a.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. In *ICLR*, 2025.

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

Haoqi Yuan, Zhixuan Liang, Anzhe Chen, Ye Wang, Haoyang Li, Pei Lin, Yiyang Huang, Zixing Lei, Tong Zhang, Jiazhao Zhang, et al. Qwen-robotmanip technical report: Alignment unlocks scale for robotic manipulation foundation models. *arXiv preprint arXiv:2606.17846*, 2026a.

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.

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

Lingzhi Zhang, Shenghao Zhou, Simon Stent, and Jianbo Shi. Fine-grained egocentric hand-object segmentation: Dataset, model, and applications. In *ECCV*, 2022.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *ICLR*, 2026a.

Ruijie Zheng, Dantong Niu, Yuqi Xie, Jing Wang, Mengda Xu, Yunfan Jiang, Fernando Castañeda, Fengyuan Hu, You Liang Tan, Letian Fu, et al. Egoscale: Scaling dexterous manipulation with diverse egocentric human data. *arXiv preprint arXiv:2602.16710*, 2026b.

A Diagnostic Analysis

A.1 Observer Variation in a Reconstruction Bottleneck

We analyze when reconstruction-based latent action learning retains observer-induced variation. Following the observation decomposition in (Lee et al., 2026) and the linear LAM analysis in (Zhang et al., 2025), we separate interaction and observer contributions to a visual transition.

Setup. Let $\mathbf{m}$ denote the hand–object interaction state and $\mathbf{c}$ the observer state. The observation map h generates visual representations $\mathbf{o} = h(\mathbf{m}, \mathbf{c})$ and $\mathbf{o}' = h(\mathbf{m}', \mathbf{c}')$, both in $\mathbb{R}^{d_o}$. Using the intermediate observation $h(\mathbf{m}', \mathbf{c})$, we decompose the observation change as

$$\begin{aligned} \Delta\mathbf{o}_{\text{int}} &= h(\mathbf{m}', \mathbf{c}) - h(\mathbf{m}, \mathbf{c}), \\ \Delta\mathbf{o}_{\text{obs}} &= h(\mathbf{m}', \mathbf{c}') - h(\mathbf{m}', \mathbf{c}), \\ \Delta\mathbf{o} &= \mathbf{o}' - \mathbf{o} = \Delta\mathbf{o}_{\text{int}} + \Delta\mathbf{o}_{\text{obs}}. \end{aligned} \tag{5}$$

The first component changes the interaction state at a fixed observer state; the second changes the observer state at the final interaction state. This identity holds for nonlinear h and does not require the two components to be independent.

For the linear analysis below, we subtract the respective means from all observation and transition variables and reuse the same notation. The additive identity above is preserved. All expectations are taken over the distribution of observation pairs, with finite second moments assumed.

Consider a linear LAM with a d_z-dimensional bottleneck, where $1 \le d_z < d_o$:

$$\mathbf{z} = \mathbf{C}\mathbf{o} + \mathbf{D}\mathbf{o}', \qquad \widehat{\mathbf{o}}' = \mathbf{A}\mathbf{o} + \mathbf{B}\mathbf{z}. \tag{6}$$

Here, $\mathbf{A} \in \mathbb{R}^{d_o \times d_o}$, $\mathbf{B} \in \mathbb{R}^{d_o \times d_z}$, and $\mathbf{C}, \mathbf{D} \in \mathbb{R}^{d_z \times d_o}$ are unconstrained trainable matrices. They minimize the expected squared reconstruction error,

$$\mathcal{L} = \mathbb{E}\left[\|\widehat{\mathbf{o}}' - \mathbf{o}'\|_2^2\right]. \tag{7}$$

Following (Zhang et al., 2025), we assume that the current observation and the total observation change are uncorrelated:

$$\mathbb{E}[\mathbf{o}\,\Delta\mathbf{o}^\top] = \mathbf{0}. \tag{8}$$

This is an additional condition of the linear analysis, not a consequence of centering.

Proposition 1 (Observer variation retained by reconstruction)**.** *For the linear LAM above under Eq. 8, let*

$$\mathbf{\Sigma} = \mathrm{Cov}(\Delta\mathbf{o}) = \mathbb{E}[\Delta\mathbf{o}\,\Delta\mathbf{o}^\top]$$

be the covariance matrix of the centered observation change. Let $\mathbf{u}_1, \ldots, \mathbf{u}_{d_o}$ be its orthonormal eigenvectors, with eigenvalues $\lambda_1 \ge \cdots \ge \lambda_{d_o} \ge 0$. Each eigenvalue is the variance along its corresponding direction: $\lambda_i = \mathbb{E}[(\mathbf{u}_i^\top \Delta\mathbf{o})^2]$.
Define $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_{d_z}]$ and $\mathbf{P} = \mathbf{U}\mathbf{U}^\top$, the orthogonal projector onto the d_z leading eigenvectors. One globally optimal reconstruction and its minimum loss are

$$\widehat{\mathbf{o}}' = \mathbf{o} + \mathbf{P}\Delta\mathbf{o}, \qquad \mathcal{L}^\star = \sum_{i=d_z+1}^{d_o} \lambda_i. \tag{9}$$

Let $\mathbf{\Sigma}_{\text{obs}} = \mathrm{Cov}(\Delta\mathbf{o}_{\text{obs}})$ be the covariance of the observer component. Its contribution to the reconstructed transition satisfies

$$\mathbb{E}\left[\|\mathbf{P}\Delta\mathbf{o}_{\text{obs}}\|_2^2\right] = \mathrm{tr}(\mathbf{P}\mathbf{\Sigma}_{\text{obs}}), \tag{10}$$

where tr *denotes the matrix trace. Thus observer variation is retained whenever it has nonzero variance in the selected principal subspace.*

Proof. Substituting $\mathbf{o}' = \mathbf{o} + \Delta\mathbf{o}$ into Eq. 6 gives

$$\widehat{\mathbf{o}}' - \mathbf{o}' = [\mathbf{A} + \mathbf{B}(\mathbf{C} + \mathbf{D}) - \mathbf{I}]\mathbf{o} + (\mathbf{B}\mathbf{D} - \mathbf{I})\Delta\mathbf{o},$$

where $\mathbf{I}$ is the $d_o \times d_o$ identity matrix. The expected cross term vanishes by Eq. 8, so

$$\begin{aligned}\mathcal{L} = \mathbb{E}\big[\|[\mathbf{A} + \mathbf{B}(\mathbf{C} + \mathbf{D}) - \mathbf{I}]\mathbf{o}\|_2^2\big] \\ + \mathbb{E}\big[\|(\mathbf{BD} - \mathbf{I})\Delta\mathbf{o}\|_2^2\big]\,.\end{aligned} \tag{11}$$

For any $\mathbf{B}, \mathbf{C}, \mathbf{D}$, choosing $\mathbf{A} = \mathbf{I} - \mathbf{B}(\mathbf{C} + \mathbf{D})$ eliminates the first term.

The product $\mathbf{BD}$ can represent any matrix of rank at most d_z. The remaining objective therefore seeks a linear reconstruction of $\Delta\mathbf{o}$ through at most d_z dimensions. By PCA optimality (Zhang et al., 2025), it is minimized by projecting onto the eigenvectors of $\mathbf{\Sigma}$ with the d_z largest eigenvalues.

To obtain the minimum loss, write $\mathbf{P} = \sum_{i=1}^{d_z} \mathbf{u}_i \mathbf{u}_i^\top$. The reconstruction discards the remaining orthogonal components:

$$\begin{aligned}\Delta\mathbf{o} - \mathbf{P}\Delta\mathbf{o} &= \sum_{i=d_z+1}^{d_o} (\mathbf{u}_i^\top \Delta\mathbf{o})\mathbf{u}_i, \\ \mathcal{L}^\star &= \mathbb{E}\big[\|\Delta\mathbf{o} - \mathbf{P}\Delta\mathbf{o}\|_2^2\big] \\ &= \sum_{i=d_z+1}^{d_o} \mathbb{E}[(\mathbf{u}_i^\top \Delta\mathbf{o})^2] = \sum_{i=d_z+1}^{d_o} \lambda_i.\end{aligned} \tag{12}$$

The squared components add because the eigenvectors are orthonormal. The minimum loss is therefore the total variance along the discarded directions. This optimum is attained in the original LAM by $\mathbf{A} = \mathbf{I}$, $\mathbf{B} = \mathbf{U}$, $\mathbf{C} = -\mathbf{U}^\top$, and $\mathbf{D} = \mathbf{U}^\top$.

Under this reconstruction, the predicted transition is $\mathbf{P}\Delta\mathbf{o}_{\text{int}} + \mathbf{P}\Delta\mathbf{o}_{\text{obs}}$. Since the observer component is centered,

$$\begin{aligned}\mathbb{E}\big[\|\mathbf{P}\Delta\mathbf{o}_{\text{obs}}\|_2^2\big] &= \operatorname{tr}(\mathbf{P}\mathbf{\Sigma}_{\text{obs}}\mathbf{P}^\top) \\ &= \operatorname{tr}(\mathbf{P}\mathbf{\Sigma}_{\text{obs}}) \\ &= \sum_{i=1}^{d_z} \mathbf{u}_i^\top \mathbf{\Sigma}_{\text{obs}} \mathbf{u}_i.\end{aligned} \tag{13}$$

The second equality uses $\mathbf{P}^\top = \mathbf{P}$, $\mathbf{P}^2 = \mathbf{P}$, and the cyclic property of the trace. Each term in the final sum is the observer variance along a selected direction and is nonnegative. The retained observer component is therefore nonzero in mean square exactly when at least one selected direction has positive observer variance. □

Interpretation and scope. The reconstruction objective selects directions according to total transition variance, without distinguishing interaction from observer motion. The result concerns a selected optimal reconstruction subspace; it does not establish statistical recoverability of observer state from the latent. When eigenvalues are tied at the cutoff, the optimal subspace may be nonunique.

A.2 Occurrence and Sensitivity of Observer Variation

The preceding analysis explains why reconstruction can retain observer variation. We now examine how observer motion affects a latent-action encoder. Following the occurrence–response decomposition in (Lee et al., 2026), we separate how often observer motion occurs from the latent response when it occurs.

Setup. Let $f : \mathbb{R}^{d_o} \times \mathbb{R}^{d_o} \to \mathbb{R}^{d_z}$ be a fixed latent-action encoder, which may be nonlinear. Using the observation map h defined above, fix an interaction transition $(\mathbf{m}, \mathbf{m}')$ and an initial observer state $\mathbf{c}$. For a final observer state $\mathbf{c}'$, define

$$\begin{aligned}\mathbf{z}(\mathbf{c}') &= f\big(h(\mathbf{m}, \mathbf{c}), h(\mathbf{m}', \mathbf{c}')\big), \\ \mathbf{z}(\mathbf{c}) &= f\big(h(\mathbf{m}, \mathbf{c}), h(\mathbf{m}', \mathbf{c})\big).\end{aligned} \tag{14}$$

The second expression keeps the observer fixed across the interaction transition. Their difference,

$$\Delta\mathbf{z} = \mathbf{z}(\mathbf{c}') - \mathbf{z}(\mathbf{c}), \tag{15}$$

therefore measures the latent response to observer motion while holding the interaction transition and current observation fixed.

All probabilities and expectations below are over $\mathbf{c}'$ conditional on the fixed $(\mathbf{m}, \mathbf{m}', \mathbf{c})$. Let $p = \Pr(\mathbf{c}' \neq \mathbf{c})$ be the probability of observer motion, and assume $\mathbb{E}[\|\Delta\mathbf{z}\|_2^2] < \infty$. No centering or linearity assumption is required.

Proposition 2 (Occurrence and response to observer motion)**.** *For the fixed interaction transition and initial observer state above, if $p > 0$, the expected squared latent change satisfies*

$$\mathbb{E}[\|\Delta\mathbf{z}\|_2^2] = p\,\mathbb{E}\big[\|\Delta\mathbf{z}\|_2^2 \mid \mathbf{c}' \neq \mathbf{c}\big]\,. \tag{16}$$

The two factors describe the occurrence probability of observer motion and the mean squared latent response when observer motion occurs. If $p = 0$, the latent change is zero almost surely.

Proof. When $\mathbf{c}' = \mathbf{c}$, the two encoder inputs in Eq. 14 coincide, so $\Delta\mathbf{z} = \mathbf{0}$. For $0 < p < 1$, the law of total expectation gives

$$\begin{aligned}\mathbb{E}[\|\Delta\mathbf{z}\|_2^2] = p\,\mathbb{E}\big[\|\Delta\mathbf{z}\|_2^2 \mid \mathbf{c}' \neq \mathbf{c}\big] \\ + (1-p)\,\mathbb{E}\big[\|\Delta\mathbf{z}\|_2^2 \mid \mathbf{c}' = \mathbf{c}\big]\,.\end{aligned}$$

The second term is zero, proving Eq. 16. For $p = 1$, conditioning on observer motion leaves the expectation unchanged; for $p = 0$, $\Delta\mathbf{z} = \mathbf{0}$ almost surely. □

Interpretation. For a fixed observer-motion distribution, the conditional expectation in Eq. 16 measures the encoder's average sensitivity to observer motion. It includes the effects of motion magnitude and direction. Comparisons between encoders therefore require the same perturbation distribution and a common latent scale. If observer motion occurs almost surely, $p = 1$ and the expected latent change is determined entirely by this response term.

WING-LAM aims to reduce the response of its interaction channel to observer motion while preserving interaction information.

A.3 Information Preservation and Frequency-Ordered Coordinates

We use the discrete cosine transform (DCT) to decompose latent-action trajectories into temporally ordered components. Here we briefly recall the standard properties needed for our analysis.

Consider $H \geq 2$ latent actions $\mathbf{z}_0, \ldots, \mathbf{z}_{H-1} \in \mathbb{R}^{D_i}$, sampled at a fixed temporal interval, and stack them as

$$\mathbf{Z} = [\mathbf{z}_0, \ldots, \mathbf{z}_{H-1}]^\top \in \mathbb{R}^{H \times D_i}.$$

Let $\mathbf{C} \in \mathbb{R}^{H \times H}$ denote the orthonormal DCT-II matrix,

$$C_{kh} = \alpha_k \cos\left[\frac{\pi k}{H}\left(h + \frac{1}{2}\right)\right], \qquad \alpha_k = \begin{cases} 1/\sqrt{H}, & k = 0, \\ \sqrt{2/H}, & k > 0, \end{cases}$$

where $h = 0, \ldots, H-1$ indexes time and $k = 0, \ldots, H-1$ indexes temporal frequency. The DCT coefficients are

$$\mathbf{Q} = \mathbf{C}\mathbf{Z},$$

with $\mathbf{q}_k^\top$ denoting the k-th row of $\mathbf{Q}$.

Because $\mathbf{C}$ is orthonormal, the full transform is invertible and preserves the trajectory energy:

$$\mathbf{Z} = \mathbf{C}^\top\mathbf{Q}, \qquad \|\mathbf{Z}\|_F^2 = \|\mathbf{Q}\|_F^2 = \sum_{k=0}^{H-1} \|\mathbf{q}_k\|_2^2. \tag{17}$$

The mode index also provides a natural ordering of temporal variation. Let $\mathbf{D} \in \mathbb{R}^{(H-1)\times H}$ be the first-difference operator,

$$(\mathbf{D}\mathbf{v})_h = v_{h+1} - v_h.$$

For the DCT-II basis, $\mathbf{C}$ diagonalizes $\mathbf{D}^\top\mathbf{D}$ (Strang, 1999), giving

$$\mathbf{C}\mathbf{D}^\top\mathbf{D}\mathbf{C}^\top = \mathrm{diag}(\lambda_0, \ldots, \lambda_{H-1}), \qquad \lambda_k = 4\sin^2\left(\frac{\pi k}{2H}\right).$$

Therefore,

$$\sum_{h=0}^{H-2} \|\mathbf{z}_{h+1} - \mathbf{z}_h\|_2^2 = \sum_{k=0}^{H-1} \lambda_k \|\mathbf{q}_k\|_2^2. \tag{18}$$

Since $0 = \lambda_0 < \lambda_1 < \cdots < \lambda_{H-1}$, higher-index modes correspond to progressively faster temporal variation.

These properties only establish that the DCT provides an information-preserving, frequency-ordered representation of the full trajectory. We further evaluate this property empirically through the cross-embodiment analyses in Appendix E.2.1.

B Related Work

B.1 Vision-Language-Action and World-Action Models

Vision-language-action (VLA) models have emerged as a dominant paradigm for generalist robot manipulation, leveraging large-scale vision-language pretraining and heterogeneous robot demonstrations to map visual observations and language instructions directly to actions (Brohan et al., 2023; Kim et al., 2024; Team et al., 2024; Black et al., 2024; Intelligence et al., 2025; NVIDIA et al., 2025; Hu et al., 2026). Recent works further scale model capacity, data diversity, and cross-embodiment training, leading to increasingly general robot policies across tasks and platforms (O'Neill et al., 2024; Liu et al., 2025; Li et al., 2024). Despite their strong semantic and visual priors, these policies primarily optimize action prediction and do not explicitly model how the observed world evolves under interaction.

World-action models (WAMs) provide a complementary direction by jointly modeling robot actions and future visual dynamics. Early approaches explored video prediction as a planning interface or as auxiliary supervision for control, while recent WAMs increasingly build on large-scale pretrained video generation models and learn actions together with future visual representations (Ye et al., 2026b; Yuan et al., 2026b; Li et al., 2026b; Yan et al., 2026; Ye et al., 2026a). Such joint prediction allows action learning to benefit from the spatiotemporal priors encoded by video models and has shown strong data efficiency and generalization in robot manipulation. Recent studies have further explored more efficient future prediction, action-centered world modeling, and richer geometric or semantic prediction targets (Yuan et al., 2026b; Yan et al., 2026; Ye et al., 2026a).

Our work builds on this world-action modeling paradigm but focuses on a different source of supervision. Rather than relying only on robot trajectories to learn interaction dynamics, we study how large-scale egocentric human videos can provide interaction priors for robot action generation. WING extracts these priors into a compact latent guidance signal, allowing human interaction dynamics to complement the visual world-modeling capability of a pretrained WAM without requiring human actions to be directly mapped to robot controls.

B.2 Learning from Egocentric Human Videos for Robot Manipulation

Egocentric human videos offer a scalable source of manipulation experience, covering diverse objects, environments, and interaction behaviors that are costly to collect through robot teleoperation. Large-scale datasets such as Ego4D, EPIC-KITCHENS, EgoDex, and EgoVerse have made such experience increasingly accessible (Grauman et al., 2022; Damen et al., 2021; Hoque et al., 2026; Punamiya et al., 2026). Early efforts primarily used human videos to learn transferable visual representations or reward functions for downstream robot learning (Nair et al., 2022; Ma et al., 2022; 2023). More recent approaches extract explicit motion cues, such as hand, wrist, or body trajectories, and convert them into robot-compatible supervision through retargeting, inverse kinematics, or embodiment-aware action representations (Li et al., 2026a; Kareer et al., 2025; Yang et al., 2025; Shi et al., 2025; Fu et al., 2024). These methods provide stronger behavioral supervision than visual pretraining, but generally require recovering human motion in a form that can be explicitly aligned with robot actions.

Latent-action learning provides a way to exploit human videos without recovering executable actions. These methods infer action-related representations from visual transitions and use them for policy or VLA pretraining (Ye et al., 2025; Bu et al., 2025; Dai et al., 2026; Yang et al., 2026). Because visual change is not equivalent to interaction dynamics, however, the resulting representations may also encode appearance, static scene content, and motion unrelated to manipulation. Recent approaches reduce such variation through task-conditioned representations, contrastive objectives, temporal cues, or foreground–background disentanglement (Bu et al., 2025; Dai et al., 2026; Yang et al., 2026; Xu et al., 2026). CD-LAM (Wei et al., 2026) further studies action-irrelevant bias in latent actions and improves their controllability through embodiment-centric reconstruction, action-centric contrastive learning, and latent-space calibration, including an evaluation of sensitivity to synthetic camera shifts. Together, these methods establish the importance of making latent actions more closely reflect embodied behavior, but primarily focus on action relevance or world-model controllability rather than the transferable temporal structure of egocentric interactions.

Observer motion makes this transfer problem particularly challenging. Head- and body-induced camera motion can dominate the visual transition even when the underlying hand–object interaction changes little. Prior work has recovered or compensated for camera motion when reconstructing human trajectories (Niu et al., 2026), while other methods treat observer motion as a useful component of active perception (Lin et al., 2026). These approaches clarify the role of observer motion in egocentric behavior, but pursue different objectives from learning latent interaction dynamics for robot policy guidance. Meanwhile, existing latent-action debiasing methods do not explicitly use observer and interaction motion as separate supervision channels. It therefore remains unclear how to reduce observer-induced variation in egocentric latent actions while preserving the interaction structure that can transfer to robot control.

WING addresses this problem in two stages. WING-LAM first uses geometric motion routing to separate dominant observer-induced image motion from local interaction residuals, and distills the resulting interaction representation into an RGB-only latent-action encoder. Reducing observer sensitivity alone, however, does not resolve differences in how humans and robots execute the same interaction, particularly in execution speed and fine-scale motion. WING therefore analyzes latent-action trajectories across temporal frequency bands, identifies low-frequency dynamics with stronger correspondence between semantically matched human and robot interactions, and uses these dynamics as guidance for robot action generation. In this way, WING connects observer-aware latent-action learning with the temporal structure used for ego-to-robot transfer.

C Implementation and Training Details

C.1 WING-LAM Architecture and Training Details

This section provides the implementation details of WING-LAM introduced in Section 3.2. WING-LAM is trained in two stages. We first construct structured motion supervision from egocentric frame pairs and train a motion-routed teacher that separates dominant observer-induced image motion from local interaction residuals. We then freeze the teacher and distill its interaction representation into an RGB-only student. The resulting student is frozen and used offline to construct latent-action targets for subsequent WING training. The optimization hyperparameters are summarized in Table 4.

C.1.1 Offline Motion Supervision Construction

Point tracks and reliability. Given a frame pair $(\mathbf{o}_t, \mathbf{o}_{t+\delta})$, we use AllTracker (Harley et al., 2025) to extract point correspondences:

$$(\mathbf{p}_{t,i}, \mathbf{p}_{t+\delta,i}), \qquad i = 1, \ldots, N_t,$$

where $\mathbf{p}_{t,i} \in \mathbb{R}^2$ denotes the image-plane location of the i-th tracked point in frame $\mathbf{o}_t$, i indexes tracked correspondences across the frame pair, and N_t is the number of correspondences returned for that pair. AllTracker additionally provides visibility and tracking-confidence estimates. We retain reliable correspondences and associate each retained track with a scalar reliability weight ρ_i derived from these estimates. EgoHOS (Zhang et al., 2022) provides an initial hand-region seed.

Global observer-motion estimation. We exclude tracks inside the initial hand region and fit an affine transformation to the remaining reliable background correspondences using deterministic RANSAC (Fischler & Bolles, 1981). The resulting warp $W_{t\to t+\delta}$ provides an image-space approximation of the dominant coherent motion induced by the moving observer. We also record a fitting confidence ρ_{cam} from the RANSAC fit and use it to weight the camera-motion supervision.

Interaction-region construction. For every reliable track, we compute its camera-compensated residual following Eq. 1:

$$\mathbf{r}_{t,i} = W^{-1}_{t\to t+\delta}(\mathbf{p}_{t+\delta,i}) - \mathbf{p}_{t,i}. \tag{19}$$

Starting from the hand-region seed, we expand the interaction region to spatially nearby tracks with sufficiently large compensated residual motion. The resulting binary indicator $m_i \in \{0,1\}$ distinguishes interaction tracks from background tracks. We denote

$$\mathcal{I}_t = \{i : m_i = 1\}, \qquad \mathcal{B}_t = \{i : m_i = 0\}, \tag{20}$$

and collect the privileged teacher inputs as

$$\mathcal{G}_t = \{\mathbf{p}_{t,i}, \mathbf{p}_{t+\delta,i}, m_i, \rho_i\}_{i=1}^{N_t}. \tag{21}$$

All quantities in $\mathcal{G}_t$ are used only during WING-LAM training.

C.1.2 Motion-Routed Teacher Architecture and Objectives

Teacher architecture. The teacher follows the two-frame spatiotemporal Transformer architecture of DreamDojo (Gao et al., 2026). The visual backbone encodes $(\mathbf{o}_t, \mathbf{o}_{t+\delta})$ into spatiotemporal visual features. For each reliable correspondence from $\mathbf{p}_{t,i}$ to $\mathbf{p}_{t+\delta,i}$, we sample the visual features at the corresponding locations in the two frames and concatenate them with the normalized point coordinates, displacement, and tracking confidence to obtain the track descriptor $\mathbf{d}_{t,i}$. Two gated set-pooling modules aggregate background and interaction descriptors separately:

$$\begin{aligned} \mathbf{z}^T_{t,\text{cam}} &= P_{\text{cam}}\left(\{\mathbf{d}_{t,i}\}_{i\in\mathcal{B}_t}\right), \\ \mathbf{z}^T_{t,\text{int}} &= P_{\text{int}}\left(\{\mathbf{d}_{t,i}\}_{i\in\mathcal{I}_t}\right), \end{aligned} \tag{22}$$

where $\mathbf{z}^T_{\text{cam}} \in \mathbb{R}^{D_c}$ and $\mathbf{z}^T_{\text{int}} \in \mathbb{R}^{D_i}$. T denotes the teacher. The two latent codes are decoded independently as:

$$\widehat{W}_{t\to t+\delta} = D_{\text{cam}}(\mathbf{z}^T_{t,\text{cam}}), \qquad \widehat{\mathbf{r}}_{t,i} = D_{\text{int}}(\mathbf{p}_{t,i}, \mathbf{z}^T_{t,\text{int}}). \tag{23}$$

For every reliable track, the decoded global and interaction motions are composed as:

$$\widehat{\mathbf{p}}_{t+\delta,i} = \widehat{W}_{t\to t+\delta}\left(\mathbf{p}_{t,i} + m_i\widehat{\mathbf{r}}_{t,i}\right). \tag{24}$$

Thus, background tracks are explained only by the global warp, whereas interaction tracks additionally receive the decoded residual motion.

Motion supervision. Let $h_\beta(\cdot,\cdot)$ denote the coordinate-wise Smooth-L_1 distance with $\beta = 0.01$, and let $\langle f_i\rangle_{w_i}$ denote the normalized weighted average of f_i with weights w_i. We supervise the global motion, canonical interaction residual, and their composition by:

$$\begin{aligned} \mathcal{L}_{\text{cam}} &= \rho_{cam}\left\langle h_\beta\left(\widehat{\mathbf{W}}_{t\to t+\delta}(\mathbf{p}_{t,i}), \mathbf{W}_{t\to t+\delta}(\mathbf{p}_{t,i})\right)\right\rangle_{\rho_i,\, i\in\mathcal{B}_t}, \\ \mathcal{L}_{\text{int}} &= \left\langle h_\beta\left(\widehat{\mathbf{r}}_{t,i}, \mathbf{r}_{t,i}\right)\right\rangle_{\rho_i,\, i\in\mathcal{I}_t}, \\ \mathcal{L}_{\text{trk}} &= \left\langle h_\beta\left(\widehat{\mathbf{p}}_{t+\delta,i}, \mathbf{p}_{t+\delta,i}\right)\right\rangle_{\rho_i}. \end{aligned} \tag{25}$$

Here, $\mathcal{L}_{\text{cam}}$ constrains the decoded observer-motion warp on background tracks, $\mathcal{L}_{\text{int}}$ supervises the camera-compensated interaction residuals, and $\mathcal{L}_{\text{trk}}$ ensures that the two components jointly explain the observed track endpoints.

C.1.3 Camera Intervention and Representation Regularization

Camera intervention. Motion routing reduces observer-induced motion in the interaction branch, but does not guarantee that the resulting representation is insensitive to observer-specific cues. We therefore construct a camera-intervened counterpart of each transition by perturbing only the global camera motion while preserving the underlying interaction motion. Encoding the natural and camera-intervened transitions yields teacher interaction representations $\mathbf{z}_{t,\text{int}}^{T,\text{nat}}$ and $\mathbf{z}_{t,\text{int}}^{T,\text{cam}} \in \mathbb{R}^{D_i}$, respectively. We encourage them to agree through:

$$\mathcal{L}_{\text{cons}} = \frac{1}{D_i} \sum_{d=1}^{D_i} h_\beta \left([\mathbf{z}_{t,\text{int}}^{T,\text{nat}}]_d, [\mathbf{z}_{t,\text{int}}^{T,\text{cam}}]_d \right), \tag{26}$$

where $[\mathbf{z}]_d$ denotes the d-th coordinate of $\mathbf{z} \in \mathbb{R}^{D_i}$. This consistency objective encourages the interaction representation to retain interaction information while reducing its sensitivity to camera motion.

Representation regularization. To prevent collapse and reduce redundancy, we apply variance and covariance regularization to the interaction representations from the natural and intervened views jointly. Let σ_d denote the batchwise standard deviation of latent dimension d. The variance regularizer is

$$\mathcal{L}_{\text{var}} = \frac{1}{D_i} \sum_{d=1}^{D_i} \max(0, \gamma - \sigma_d)^2, \qquad \gamma = 0.1. \tag{27}$$

Let $\mathbf{C} \in \mathbb{R}^{D_i \times D_i}$ denote the covariance matrix of the standardized latent dimensions, where C_{de} is the covariance between latent dimensions d and e. We further use

$$\mathcal{L}_{\text{cov}} = \frac{1}{D_i(D_i - 1)} \sum_{d \neq e} C_{de}^2. \tag{28}$$

The complete teacher objective is

$$\begin{aligned} \mathcal{L}_{\text{teacher}} = \lambda_{\text{trk}} \mathcal{L}_{\text{trk}} + \lambda_{\text{cam}} \mathcal{L}_{\text{cam}} + \lambda_{\text{int}} \mathcal{L}_{\text{int}} \\ + \lambda_{\text{cons}} \mathcal{L}_{\text{cons}} + \lambda_{\text{var}} \mathcal{L}_{\text{var}} + \lambda_{\text{cov}} \mathcal{L}_{\text{cov}}. \end{aligned} \tag{29}$$

The motion-supervision terms establish the camera–interaction routing, while the consistency and representation regularizers further suppress observer-motion leakage and stabilize the interaction latent space.

C.1.4 Interaction Distillation

The motion-routed teacher depends on privileged point tracks, interaction masks, and global-motion estimates, making it unsuitable for large-scale latent extraction. We therefore freeze the teacher and distill only its interaction representation into an RGB-only student S_ϕ.

The student is initialized from the teacher visual encoder but receives only the RGB frame pair. Its spatiotemporal visual tokens are mean-pooled and linearly projected into a D_i-dimensional interaction latent; no point tracks, masks, or motion parameters are provided. For each training transition, we use the frozen teacher's interaction representation from the natural view as the common distillation target:

$$\mathbf{z}_{t,\text{int}}^{T} := \mathbf{z}_{t,\text{int}}^{T,\text{nat}} = E_{T,\text{int}}(\mathbf{o}_t, \mathbf{o}_{t+\delta}, \mathcal{G}_t), \tag{30}$$

where $E_{T,\text{int}}$ denotes the interaction output of the teacher E_T. The student encodes the natural and camera-intervened frame pairs into $\mathbf{z}_t^{\text{nat}}$ and $\mathbf{z}_t^{\text{aug}}$, respectively. Both representations are trained to match $(\mathbf{z}_{t,\text{int}}^{T})$ and to agree with each other through Eq. 2. The teacher remains frozen, and gradients update only the student S_ϕ. After distillation, all privileged motion-processing components and the teacher are discarded. The frozen RGB-only student defines the WING-LAM latent action

$$\mathbf{z}_t = S_\phi \left(\mathbf{o}_t, \mathbf{o}_{t+\delta} \right), \tag{31}$$

which is subsequently used offline to construct the latent-action trajectories for WING.

Table 4: **WING-LAM training configuration.**

Setting	Value
Motion-Supervised Teacher Learning	
Latent dimensions	$D_{\mathrm{c}} = 16$; $D_{\mathrm{i}} = 32$
Loss weights	$\lambda_{\mathrm{trk}} = \lambda_{\mathrm{cam}} = \lambda_{\mathrm{int}} = \lambda_{\mathrm{var}} = 1$; $\lambda_{\mathrm{cons}} = 0.1$; $\lambda_{\mathrm{cov}} = 0.01$
Optimizer	AdamW
Training steps	60,000
Learning rate	5×10^{-6}
Weight decay	10^{-4}
Interaction Representation Distillation	
Latent dimensions	$D_{\mathrm{i}} = 32$
Loss weights	$\lambda_{\mathrm{view}} = 1$
Optimizer	AdamW
Training steps	60,000
Learning rate	10^{-5}
Weight decay	0.05

C.2 Egocentric-Video Pretraining Details

C.2.1 Pre-training Data and Preprocessing.

We pre-train our model on three large-scale egocentric video datasets: Ego4D, EgoDex, and EgoVerse, totaling approximately 200 hours of ego-videos. Specifically, Ego4D contributes approximately 20 hours, EgoDex contributes about 85 hours, and EgoVerse contributes 95 hours. Language instructions are obtained from the original task descriptions of EgoDex and EgoVerse, while those for Ego4D are constructed from the verb–noun action labels in its official long-term anticipation annotations. All egocentric videos are standardized to 30 FPS. The sampled RGB frames are resized to 384×640 and encoded by the frozen Wan2.2 VAE.

Latent Action Preprocessing. To construct latent-action supervision, we sample egocentric videos at a fixed physical interval of $0.2\,\mathrm{s}$, corresponding to every 6 frames at 30 FPS. The sampled frames are center-cropped to a 4:3 aspect ratio, resized to 240×320, and consecutive sampled endpoints are encoded by the frozen WING-LAM into 32-dimensional latent transitions. Nine sampled endpoints yield $H = 8$ transitions, spanning a $1.6\,\mathrm{s}$ temporal window.

Hand Pose Preprocessing For hand-pose-based pretraining, we construct a 200-hour corpus by supplementing it with 10 hours of hand-annotated data from each of EgoDex and EgoVerse, replacing the Ego4D portion that lacks the required annotations. We then follow the Ego2Robot (Wang et al., 2026) pipeline in RobotManip (Yuan et al., 2026a) to extract hand end-effector (EEF) trajectories from these annotations. The resulting trajectories serve as action supervision during model pretraining.

C.2.2 Guidance Predictor Training

During egocentric-video pretraining, we train the guidance predictor to estimate interaction-centric spectral guidance from the current visual observation and language instruction. For each training sample, the frozen WING-LAM encoder extracts an 8×32 latent-action trajectory from the sampled video frames. We apply an orthonormal temporal DCT and retain the lowest four frequency components, producing a 4×32 guidance target.

The predictor is a lightweight, current-observation-conditioned query Transformer. It uses four learned queries that cross-attend to a context formed by concatenating linearly projected current-frame tokens and language-instruction tokens. A single Transformer block with a hidden dimension of 1,024, eight attention heads, and a fourfold-expanded feed-forward network produces four query features, which are projected to 32 dimensions to predict the 4×32 normalized guidance target. The prediction loss is weighted by 0.1. Future frames are used only to construct the training target and are never provided to the predictor.

C.2.3 Optimization Setup

We conduct egocentric-video pre-training for 40 epochs on 32 NVIDIA H100 GPUs. We use a per-GPU batch size of 16, resulting in an effective global batch size of 512. The learning rate is set to 1×10^{-4}. To reduce training overhead, both the Wan2.2 VAE latents and WING-LAM latent transitions are pre-computed and cached before training. The video backbone is initialized from Wan2.2-TI2V-5B (Wan et al., 2025).

C.3 Robot Policy Training Details

C.3.1 Training Framework

For our model, the guidance predictor and shared visual backbone are initialized from the egocentric-video pretrained checkpoint. During robot policy training, the guidance predictor is jointly optimized with the robot policy. Given the current visual observation, language instruction, and robot proprioceptive state, the predictor estimates the low-frequency spectral representation of the anticipated interaction dynamics. The predicted coefficients are projected into guidance tokens and injected into the action model through gated cross-attention, providing an additional conditioning signal for action prediction. The spectral prediction targets are constructed following the same procedure as in egocentric-video pretraining. The overall training objective is

$$\mathcal{L}_{\text{total}} = \lambda_{\text{video}}\mathcal{L}_{\text{video}} + \lambda_{\text{action}}\mathcal{L}_{\text{action}} + \lambda_{\text{pred}}\mathcal{L}_{\text{pred}}, \tag{32}$$

where $\lambda_{\text{video}} = \lambda_{\text{action}} = 1$, and $\lambda_{\text{pred}} = 0.1$.

C.3.2 Optimization Setup

Robot policy post-training is conducted on 8 NVIDIA H100 GPUs with a global batch size of 128. We use the AdamW optimizer with a peak learning rate of 2×10^{-4}. The learning rate is linearly warmed up and then decayed according to a cosine schedule.

D Evaluation Details

In this section, we present the details of our evaluation experiments.

D.1 Latent-Action Representation Evaluation Details

We evaluate frozen latent-action representations along three complementary dimensions: *(i) what camera- and interaction-motion information can be decoded from the latent representation, (ii) how sensitive the representation is to controlled changes in camera motion,* and *(iii) how well it preserves high-level action semantics.* This section describes the datasets, targets, and evaluation protocols used for these analyses.

D.1.1 Datasets and Frozen Representations

We evaluate WING-LAM together with DreamDojo (Gao et al., 2026), CD-LAM (Wei et al., 2026), LAPA (Ye et al., 2025), VILLA-X (Chen et al., 2026b), and UniVLA (Bu et al., 2025) using their frozen latent-action encoders. The motion decodability and camera-sensitivity analyses use EgoDex (Hoque et al., 2026), EgoVerse (Punamiya et al., 2026), and Ego4D (Grauman et al., 2022). EgoDex provides synchronized head-pose annotations, while EgoVerse and Ego4D provide additional diversity in natural observer and interaction motion. We split videos by source trajectory to avoid overlap between probe training and evaluation. The action-semantic evaluation follows the LARYBench protocol and is described separately in Appendix D.1.4.

D.1.2 Motion Decodability

We evaluate how much camera- and interaction-motion information can be decoded from each frozen latent representation. For camera motion, we use the relative head motion when ground-truth head pose is available, and otherwise use the estimated global image motion as a proxy. For interaction motion, we remove the estimated global camera motion from the tracked point motion

and aggregate the remaining local motion into a fixed-dimensional target. We evaluate each frozen latent representation using three probe families: a ridge linear probe, a two-layer MLP with GELU activations, and a random Fourier feature (RFF) probe followed by regularized linear regression. The MLP is trained with AdamW, while the ridge and RFF probes use regularized linear regression. In all cases, the latent encoder remains frozen and only the probe is optimized.

For each target, we train a probe on the frozen latent representation and report the coefficient of determination (R^2) on the held-out test set:

$$R^2 = 1 - \frac{\sum_{n=1}^{N} \|\mathbf{y}_n - \widehat{\mathbf{y}}_n\|_2^2}{\sum_{n=1}^{N} \|\mathbf{y}_n - \bar{\mathbf{y}}\|_2^2}, \qquad \bar{\mathbf{y}} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{y}_n, \tag{33}$$

where $\mathbf{y}_n$ and $\widehat{\mathbf{y}}_n$ denote the ground-truth and predicted motion targets for sample n, respectively. Higher R^2 indicates that the corresponding motion information is more readily decoded from the latent representation. An R^2 of zero corresponds to the constant-mean baseline, while negative values indicate lower predictive accuracy than this baseline.

D.1.3 Sensitivity to Camera Perturbations

We complement motion decodability with a controlled intervention that directly measures how much a frozen latent representation changes under additional camera motion. For each frame pair $(\mathbf{x}_t, \mathbf{x}_{t+\delta})$, we construct matched *dynamic* and *static* perturbations. Dynamic perturbations apply different global transformations to the two frames, introducing additional inter-frame camera motion, whereas static perturbations apply the same transformation to both frames. The transformation magnitude and image-processing pipeline are matched between the two conditions, so that their difference primarily reflects sensitivity to relative camera motion rather than interpolation or boundary artifacts. For the multi-model comparison in Fig. 3, we instantiate this protocol using two transformations A and B of equal magnitude and opposite direction. The pairs AB and BA form the dynamic conditions, while AA and BB serve as their matched static controls. Let $\mathbf{z}^{PQ}$ denote the latent representation obtained after applying transformation P to the first frame and Q to the second, and let $\mathbf{z}^0$ denote the representation of the unperturbed pair.

Because latent scales differ across models, each latent dimension is standardized using statistics computed from the corresponding training split. We measure the standardized deviation of each perturbed representation from the unperturbed pair as

$$d^{PQ} = \sqrt{\frac{1}{D_i}\sum_{d=1}^{D_i}\left(\frac{[\mathbf{z}^{PQ}]_d - [\mathbf{z}^0]_d}{\sigma_d}\right)^2}, \qquad PQ \in \{AB, BA, AA, BB\}, \tag{34}$$

where σ_d is the training-set standard deviation of latent dimension d and D_i denotes the latent action's dimension. We report the dynamic-minus-static deviation,

$$\frac{1}{2}\left[d^{AB} + d^{BA}\right] - \frac{1}{2}\left[d^{AA} + d^{BB}\right]. \tag{35}$$

A smaller value indicates that the representation changes less in response to additional inter-frame camera motion after accounting for matched image perturbations.

We evaluate five increasing perturbation levels, $(0.25\,\text{px}, 0.05^\circ)$, $(0.5\,\text{px}, 0.1^\circ)$, $(1\,\text{px}, 0.2^\circ)$, $(2\,\text{px}, 0.4^\circ)$, and $(4\,\text{px}, 0.8^\circ)$, where each pair specifies the translation and rotation magnitude.

D.1.4 Action-Semantic Preservation

We further evaluate whether reducing camera-motion sensitivity preserves interaction-relevant semantic information. Following the LARYBench (Nie et al., 2026) evaluation protocol, we evaluate frozen latent representations on action-category classification across Ego4D (Grauman et al., 2022), EgoDex (Hoque et al., 2026), EPIC-KITCHENS (Damen et al., 2021), and HoloAssist (Wang et al., 2023). For each method, the latent encoder remains frozen and the same classification protocol is applied to the representations.

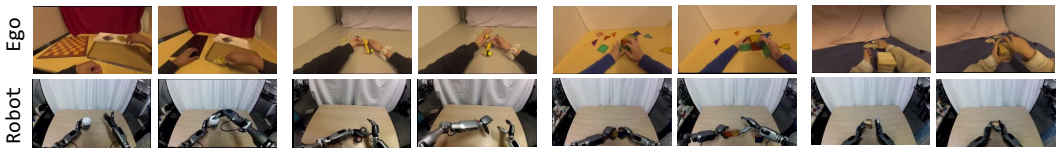


Figure 6: **Examples of matched ego–robot interaction pairs.** Each group shows two frames from an egocentric clip and two frames from its semantically matched robot clip.

D.2 Pair Construction for Spectral Similarity Analysis

To evaluate cross-embodiment correspondence, we construct matched and mismatched human–robot clip pairs through a hierarchical annotation protocol. We first retrieve candidate human and robot episodes from EgoDex and EgoDex-Eval according to their task annotations. These task labels are used only for coarse-grained episode-level filtering, since episodes assigned to the same task may still contain substantially different local motion patterns, interaction phases, or even opposite motion directions.

Within each candidate episode pool, human annotators further inspect the temporal content and select short clips according to their dominant action semantics. Specifically, a matched pair is formed by selecting a human clip and a robot clip that exhibit similar dominant interaction dynamics, rather than requiring frame-wise or trajectory-level alignment. Consequently, matched clips may differ substantially in object instance, scene appearance, viewpoint, operator, and precise end-effector trajectory, while still sharing the same underlying interaction pattern. This procedure therefore performs coarse task-level retrieval first, followed by fine-grained clip-level semantic matching within the retrieved episodes.

Mismatched pairs are constructed from the same task-conditioned candidate pools, but the selected human and robot clips exhibit inconsistent dominant motion semantics. This preserves a comparable task distribution while removing action-level correspondence. To reduce dependence among evaluation samples, each source episode is used at most once in the matched set and at most once in the mismatched set. The final evaluation set contains 100 matched and 100 mismatched pairs, covering 25 distinct tasks. The pair set and all match/mismatch labels were finalized before computing any latent trajectories or spectral similarities. To assess annotation reliability, two additional annotators independently reviewed the constructed pairs and showed high agreement with the original labels.

We use our frozen WING-LAM, the same latent-action encoder used to construct spectral guidance in WING, to encode both human and robot clips. WING-LAM is trained only on egocentric videos and is not adapted using robot data for this analysis. Therefore, this evaluation directly tests whether the interaction-centric latent space learned from egocentric data generalizes across embodiments, and whether semantically matched human and robot interactions exhibit consistent temporal structure in this shared representation.

For both egocentric and robot clips, we use the same physical sampling interval of 0.2s, corresponding to an effective sampling rate of 5 Hz. Each $H = 8$ latent-action trajectory is constructed from 9 sampled observation endpoints and spans 1.6s. We then apply DCT-II along the temporal dimension and compute cosine similarity between the corresponding low-, high-, and full-frequency representations. We report the average similarity over the 100 matched and 100 mismatched pairs in Figure 3.

D.3 Simulation Evaluation Details

D.3.1 Benchmark Protocols and Metrics

LIBERO. LIBERO contains four suites—Spatial, Object, Goal, and Long—with 10 tasks and 500 demonstrations per suite. We train one checkpoint across the four suites and evaluate each task for 50 rollouts, giving 2,000 rollouts in total. We use the benchmark episode budgets for each suite and report the mean task success rate for every suite together with their four-suite average.

RoboTwin 2.0. RoboTwin evaluates 50 bimanual tasks in clean and domain-randomized scenes. Following the standard multi-task protocol, training uses 2,500 clean demonstrations and 25,000

randomized demonstrations. Each task is evaluated for 100 rollouts under each scene condition. We report the 50-task clean mean, the randomized mean, and their arithmetic average.

RoboCasa–GR1. RoboCasa–GR1 contains 24 tabletop tasks executed by a Fourier GR1 humanoid with two 7-DoF arms, two 6-DoF hands, and a 3-DoF waist. We use the Unified 1K dataset with 1,000 teleoperated demonstrations per task and egocentric RGB observations. Each task is evaluated over 50 rollouts, and the main metric is the average of the 24 task success rates.

Evaluation. All simulation tables report success rate in percent. Notably, the FastWAM result on RoboCasa–GR1 is our reproduction trained to convergence. Complete RoboTwin and RoboCasa task-level tables are reported below; method-specific optimization details are given in Appendix C.3.

D.3.2 Baselines

We compare our method with recent robot manipulation policies spanning three representative paradigms: vision-language-action models, latent-action based policies, and world-action modeling approaches. Together, these baselines cover methods that directly map multimodal observations to robot actions, methods that introduce an intermediate latent action representation, and methods that explicitly exploit predicted future dynamics or world representations for policy learning.

Vision-language-action models. We first consider representative vision-language-action (VLA) policies that directly predict robot actions from visual observations and language instructions. These include π_0 (Black et al., 2024) and $\pi_{0.5}$ (Intelligence et al., 2025), X-VLA (Zheng et al., 2026a), LingBot-VLA (Wu et al., 2026), GR00T (N1.6 and N1.7) (NVIDIA et al., 2025), StarVLA-OFT (Community, 2026) and LDA-1B (Lyu et al., 2026). This group represents strong recent VLA systems that learn robot control directly in the action space without explicitly introducing a latent-action or future-dynamics interface.

Latent-action based methods. We additionally compare against approaches that introduce latent actions as an intermediate representation for learning manipulation behaviors, including LAPA (Ye et al., 2025), UniVLA (Bu et al., 2025), LaWAM (Chen et al., 2026a), Being-H0.7 (Luo et al., 2026), and VLA-JEPA (Sun et al., 2026). These methods are particularly relevant to our setting because they investigate whether interaction or action information can be represented in a latent space rather than being directly tied to embodiment-specific robot commands. In contrast, our method uses interaction-centric latent dynamics with reduced sensitivity to observer motion and emphasizes their low-frequency temporal structure as guidance for robot action learning.

World-action modeling methods. The third group contains methods that explicitly leverage predictive world representations, future observations, or future latent dynamics to facilitate action learning. We include Motus (Bi et al., 2026), LingBot-VA (Li et al., 2026b), FastWAM (Yuan et al., 2026b), GigaWorld-Policy (Ye et al., 2026a), LDA-1B (Lyu et al., 2026) and DiT4DiT (Ma et al., 2026). These baselines are the closest to our method in their use of anticipated future dynamics for control. Our method differs by constructing guidance from interaction-centric latent dynamics extracted from egocentric videos and using their low-frequency components, which exhibit stronger correspondence across human and robot executions, rather than conditioning the policy on the full latent trajectory.

Real-world baselines. For the real-world experiments, we focus on representative strong methods from the above families and compare against GR00T-N1.7 (NVIDIA et al., 2025), $\pi_{0.5}$ (Intelligence et al., 2025), and FastWAM (Yuan et al., 2026b). These baselines provide comparisons with a strong general-purpose VLA model, a recent large-scale robot policy, and a predictive world-action modeling approach, respectively, under the same real-world evaluation protocol.

D.4 Real-World Evaluation Protocols

D.4.1 Tasks and Progress Metrics

The real-world suite contains four tasks. *Pack Objects* requires placing a two-brick LEGO assembly, a tape roll, and an inverted plastic cup in a container. *Stack Cups* requires arranging three cups into

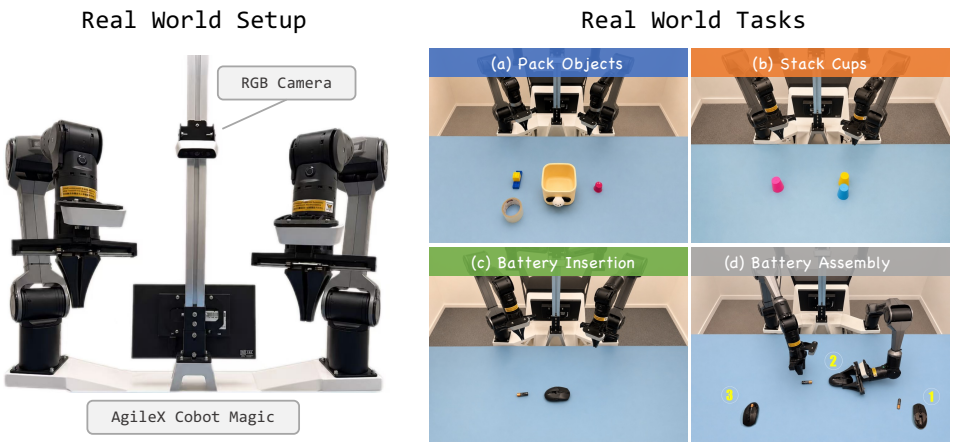


Figure 7: **Real-world tasks setup.** The left panel shows the robot platform and sensing setup; the remaining panels show base-setting executions of Pack Objects, Stack Cups, Battery Insertion, and Long-Horizon Battery Assembly.

Table 5: **Stage-wise scores distinguish partial progress from task completion.** Each task receives a maximum score of 10.

Task	Sub-goals	Score
Pack Objects	Pack one object	3.0
	Pack two objects	6.0
	Pack three objects	10.0
Stack Cups	Place the left base cup	3.0
	Place the right base cup	5.0
	Place the top cup	7.5
	Achieve a stable stack	10.0
Battery Insertion	Grasp the battery	3.0
	Align the battery with the slot and release it	5.0
	Press the battery into place	7.5
	Release the assembled mouse	10.0
Battery Assembly	Move the parts to the workspace	3.0
	Insert the battery	5.0
	Grasp the assembled mouse	7.5
	Move the mouse to the target place	10.0

a stable tower. *Battery Insertion* requires grasping a battery, aligning it with a mouse compartment, and seating it fully. *Long-Horizon Battery Assembly* additionally requires relocating the mouse and battery to the workspace before insertion and moving the assembled mouse to a target region. We collect 100 demonstrations each for Pack Objects and Stack Cups, and 150 demonstrations each for Battery Insertion and Long-Horizon Battery Assembly. For each task, we evaluate the policy over 20 trials. We report binary success and a stage-wise score. Success requires the final task condition to remain satisfied for at least three seconds. The score is the furthest valid milestone reached while all preceding milestones remain satisfied; Pack Objects is order-independent and scores the number of objects placed. Table 5 gives the complete rubric. The deployment code will be released.

Each rollout is autonomous: no human correction, object reset, or policy restart is permitted before termination. Hardware- or communication-only interruptions are discarded and repeated, whereas policy-induced safety terminations count as failures. The recorded protocol uses 20 trials per reported setting and the same predefined initial configurations, observation inputs, language instructions, action limits, termination rules, and low-level controller for all methods.

D.4.2 GENERALIZATION CONDITIONS

The generalization study changes one factor at a time: object appearance, task-irrelevant background, object geometry, or initial spatial layout. Camera placement, robot base pose, and all other factors are held fixed. Changed objects, colors, backgrounds, and poses are excluded from task-specific robot training demonstrations. Geometry is not evaluated for the battery tasks because changing the insertion dimensions would change the task mechanics. The task-specific generalization settings are defined as follows:

- *Pack Objects:*
  - **Appearance:** Replace the original cup and LEGO bricks with instances of different colors while preserving their shapes and sizes.
  - **Background:** Introduce task-irrelevant tabletop objects or vary the illumination without obstructing the target objects.
  - **Geometry:** Change the way the LEGO pieces are connected, while keeping its function in the task, and change the size of the cup.
  - **Spatial layout:** Relocate the tape roll and cup within predefined reachable regions while keeping the container fixed.
- *Stack Cups:*
  - **Appearance:** Replace the original cups with cups of different colors but identical geometry.
  - **Background:** Add task-irrelevant clutter to the tabletop while keeping the manipulation area reachable.
  - **Geometry:** Replace the original cups with a larger affordance-equivalent set that can still be arranged into the same tower structure.
  - **Spatial layout:** Randomize the initial positions of all three cups within predefined regions while preserving task feasibility.
- *Battery Insertion:*
  - **Appearance:** Replace the original battery with a color-different but dimension-matched instance.
  - **Background:** Add task-irrelevant tabletop clutter without occluding the mouse or battery.
  - **Spatial layout:** Randomize the initial positions and orientations of the mouse and battery within predefined reachable regions.
  - **Geometry:** Not evaluated, because changing the battery dimensions or compartment geometry would alter the insertion mechanics and therefore change the task itself.
- *Battery Assembly:*
  - **Appearance:** Replace the original battery with a color-different but dimension-matched instance.
  - **Background:** Introduce task-irrelevant tabletop clutter while preserving the visibility of all task objects.
  - **Spatial layout:** Vary the initial positions and orientations of the mouse and battery while retaining the required relocation, insertion, and final-placement stages.
  - **Geometry:** Not evaluated, because changing the insertion geometry would alter the underlying task requirements.

All shifted conditions use the same milestone rubric as the base setting. Multi-factor shifts, if evaluated, must be reported separately rather than mixed into the single-factor averages.

D.4.3 PLATFORM AND DEPLOYMENT

We conduct real-world experiments on a dual-arm Agilex Cobot Magic robot platform. The robot is equipped with three Orbbec DaBai DC1 cameras, including one front-facing camera and two wrist-

mounted cameras, which provide visual observations for policy inference. The robot action space consists of 14 dimensions, including the 6-DoF pose commands for each arm and additional gripper control signals. All real-world experiments use the same inference pipeline without additional task-specific or model-specific modifications. The deployment framework and corresponding code will be released to facilitate reproducibility.

E Additional Results

E.1 Additional Latent Action Results

E.1.1 Camera–Interaction Readout Results

Factor-selective routing across datasets. As shown in Table 6, across EgoDex, EgoVerse, and Ego4D, WING-LAM consistently exhibits complementary factor routing: $\mathbf{z}_{\mathrm{int}}$ preserves interaction-related information while suppressing observer-motion information, whereas $\mathbf{z}_{\mathrm{cam}}$ captures observer-motion variation with minimal interaction information. Notably, for EgoDex, observer-motion readout is evaluated against the recorded 6D head pose rather than the affine motion proxy used to supervise WING-LAM during training. The reduced head-motion decodability from $\mathbf{z}_{\mathrm{int}}$ therefore indicates that suppression learned from the affine proxy transfers, at least partially, to genuine egocentric head motion. The student model further inherits this routing behavior from the trajectory-assisted teacher, demonstrating that the interaction-centric representation can be obtained without privileged inputs at inference time.

Probe robustness. We further evaluate the readout results using probes with different capacities, including Ridge, MLP, and RFF. Although stronger probes improve absolute readout scores for some baselines, the qualitative routing behavior of WING-LAM remains unchanged, with complementary interaction and camera-motion selectivity preserved across probe settings.

E.1.2 Detailed LARYBench Results

Table 7 reports the detailed LARYBench (Nie et al., 2026) action-classification results across Ego4D, EPIC-KITCHENS, HoloAssist, and EgoDex. All latent-action encoders are frozen and evaluated under the same classification protocol. WING-LAM achieves the highest average accuracy across the four datasets, indicating that observer debiasing preserves action-related semantic information in the learned latent representation. These per-dataset results provide the details corresponding to the aggregate LARYBench result reported in the main text.

E.1.3 Qualitative Visualization of Interaction Motion

To further examine what motion patterns are captured by WING-LAM, we conduct a qualitative analysis by feeding its latent actions into the teacher decoder to reconstruct motion trajectories. Specifically, we extract trajectories from three consecutive clips and concatenate them sequentially to visualize longer-term motion patterns. The decoded trajectories are then visualized across diverse egocentric interactions in Figure 8. Across all examples, the decoded trajectories consistently concentrate on moving hands, forearms, manipulated objects, and their surrounding interaction regions. Their dominant directions broadly follow the local changes between the two frames across diverse activities, including object transfer, tool use, surface wiping, and food preparation. Comparatively few interaction responses appear in background regions, even when the egocentric viewpoint changes noticeably. These examples provide qualitative evidence that the WING-LAM latent preferentially preserves interaction-relevant motion rather than camera-induced global scene motion.

E.2 Additional Spectral Results

E.2.1 Cross-Embodiment Spectral Similarity Analysis

Experimental settings are detailed in Appendix D.2. As shown in Figure 3(d), low-frequency latents achieve the largest separation between semantically matched and mismatched human–robot interaction pairs, with a matched–mismatched gap of 0.51 (95% bootstrap CI: $[0.48, 0.53]$). In contrast, full-spectrum and high-frequency representations obtain substantially smaller gaps of 0.26



Figure 8: **Qualitative visualization of interaction motion decoded from WING-LAM.**

Table 6: **WING-LAM provides complementary camera- and interaction-factor readout across datasets.** We report held-out R^2 for camera and interaction targets using Ridge, MLP, and RFF probes. WING-LAM's representations exhibit weak camera readout while retaining strong interaction readout, whereas its camera representation exhibits the opposite pattern. Camera targets are ground-truth 6D head pose for EgoDex and image-space camera-motion proxies for EgoVerse and Ego4D. Each $\pm$ value denotes the half-width of a 95% grouped-bootstrap confidence interval. Among interaction representations, **bold** and underlined values mark the lowest and second-lowest R^2_{cam} and the highest and second-highest R^2_{int}, respectively, within each dataset and probe.

	Ridge		MLP		RFF	
Representation	R^2_{cam}	R^2_{int}	R^2_{cam}	R^2_{int}	R^2_{cam}	R^2_{int}
EgoDex						
DreamDojo	0.32 ± 0.04	0.12 ± 0.08	0.70 ± 0.05	0.25 ± 0.04	0.54 ± 0.08	0.17 ± 0.02
CD-LAM	0.54 ± 0.03	0.12 ± 0.00	0.71 ± 0.05	0.21 ± 0.02	0.56 ± 0.08	0.17 ± 0.03
LAPA	-0.23 ± 0.05	0.02 ± 0.07	**-0.27** ± 0.04	-0.03 ± 0.01	-0.23 ± 0.00	0.02 ± 0.00
VILLA-X	-0.03 ± 0.06	-0.41 ± 0.05	0.07 ± 0.02	-0.32 ± 0.06	-0.03 ± 0.01	-0.59 ± 0.01
UniVLA	**-0.56** ± 0.11	-0.05 ± 0.01	-0.03 ± 0.05	0.01 ± 0.06	**-0.56** ± 0.21	-0.05 ± 0.00
teacher $\mathbf{z}_{\text{int}}$	-0.06 ± 0.06	**0.82** ± 0.04	-0.03 ± 0.04	**0.74** ± 0.09	-0.06 ± 0.04	**0.82** ± 0.11
student $\mathbf{z}_{\text{int}}$	-0.02 ± 0.08	0.57 ± 0.02	0.00 ± 0.03	0.48 ± 0.03	-0.02 ± 0.03	0.54 ± 0.07
teacher $\mathbf{z}_{\text{cam}}$	0.93 ± 0.03	-0.04 ± 0.05	0.93 ± 0.06	-0.10 ± 0.06	0.92 ± 0.08	-0.04 ± 0.00
EgoVerse						
DreamDojo	0.26 ± 0.01	-0.22 ± 0.02	0.74 ± 0.08	0.45 ± 0.04	0.65 ± 0.05	-0.16 ± 0.03
CD-LAM	0.34 ± 0.08	-0.14 ± 0.02	0.75 ± 0.03	0.46 ± 0.02	0.63 ± 0.08	-0.06 ± 0.03
LAPA	-0.08 ± 0.06	-0.33 ± 0.00	-0.25 ± 0.01	-0.27 ± 0.07	-0.08 ± 0.00	-0.33 ± 0.00
VILLA-X	0.26 ± 0.04	-0.19 ± 0.07	0.07 ± 0.05	0.02 ± 0.02	0.26 ± 0.02	0.04 ± 0.03
UniVLA	**-0.52** ± 0.09	-0.14 ± 0.05	**-0.47** ± 0.15	-0.03 ± 0.01	-0.52 ± 0.01	-0.22 ± 0.08
teacher $\mathbf{z}_{\text{int}}$	-0.28 ± 0.06	**0.79** ± 0.06	-0.27 ± 0.03	**0.79** ± 0.04	-0.28 ± 0.00	**0.83** ± 0.06
student $\mathbf{z}_{\text{int}}$	-0.23 ± 0.07	0.74 ± 0.03	-0.27 ± 0.02	0.76 ± 0.07	**-0.61** ± 0.04	0.69 ± 0.04
teacher $\mathbf{z}_{\text{cam}}$	0.97 ± 0.02	-0.35 ± 0.05	0.96 ± 0.01	-0.09 ± 0.00	0.97 ± 0.01	-0.64 ± 0.16
Ego4D						
DreamDojo	0.47 ± 0.00	0.15 ± 0.04	0.72 ± 0.03	0.08 ± 0.07	0.59 ± 0.05	0.15 ± 0.00
CD-LAM	0.28 ± 0.00	0.15 ± 0.04	0.77 ± 0.04	0.09 ± 0.07	0.54 ± 0.06	0.15 ± 0.00
LAPA	**-0.03** ± 0.06	-0.11 ± 0.06	0.02 ± 0.05	-0.10 ± 0.03	**-0.02** ± 0.00	-0.12 ± 0.02
VILLA-X	0.34 ± 0.08	-0.06 ± 0.07	0.28 ± 0.06	-0.35 ± 0.05	0.34 ± 0.00	-0.08 ± 0.00
UniVLA	0.15 ± 0.07	-0.11 ± 0.07	0.16 ± 0.05	-0.06 ± 0.01	0.15 ± 0.00	-0.11 ± 0.00
teacher $\mathbf{z}_{\text{int}}$	-0.02 ± 0.07	**0.66** ± 0.00	**-0.03** ± 0.01	**0.64** ± 0.06	**-0.02** ± 0.02	**0.71** ± 0.05
student $\mathbf{z}_{\text{int}}$	0.07 ± 0.09	0.42 ± 0.08	0.04 ± 0.04	0.38 ± 0.05	0.07 ± 0.00	0.42 ± 0.02
teacher $\mathbf{z}_{\text{cam}}$	0.93 ± 0.03	-0.11 ± 0.01	0.95 ± 0.02	-0.09 ± 0.06	0.93 ± 0.02	-0.09 ± 0.07

Table 7: Comparison of different latent action models across four datasets.

Model	**WING-LAM**	**DreamDojo**	**CD-LAM**	**LAPA**	**UniVLA**	**VILLA-X**
Ego4D	64.13	60.92	60.67	21.02	15.22	14.80
EPIC-KITCHENS	40.57	40.65	41.19	17.58	9.95	10.35
HoloAssist	48.41	46.23	48.95	22.40	15.84	20.69
EgoDex	68.77	65.92	62.95	22.56	17.43	17.20
Avg.	**55.47**	53.43	53.44	20.89	14.61	15.76

($[0.22, 0.27]$) and 0.05 ($[0.03, 0.10]$), respectively. Importantly, repeating the same analysis with the pretrained DreamDojo LAM yields the same qualitative ordering, with low-frequency components exhibiting the largest matched–mismatched separation, suggesting that this frequency-selective correspondence is not specific to WING-LAM. To verify that this advantage is not merely driven by the dominant energy of the zeroth component, we further exclude q_0 and evaluate the remaining low-frequency dynamics. The $q_1 : q_3$ components retain a matched-pair similarity of 0.42, compared with only 0.08 for mismatched pairs. In contrast, extending the representation to $q_1 : q_7$ decreases the matched similarity to 0.31 while increasing the mismatched similarity to 0.14. This shows that the low-frequency correspondence persists even after removing the zeroth component,

and that incorporating higher-frequency components weakens the separation between semantically matched and mismatched interactions.

Overall, these results suggest that low-frequency latent components preserve interaction semantics with stronger cross-embodiment correspondence, whereas higher-frequency components contain variation that is less consistently shared across human and robot executions.

E.2.2 Spectral Bandwidth Gap Between Egocentric and Robot Data

We additionally examine the spectral distributions of latent-action trajectories in egocentric and robot data. For this dataset-level bandwidth analysis, we use the pretrained DreamDojo LAM rather than WING-LAM. DreamDojo has been trained on both egocentric and robot data, making it better suited for characterizing differences in the temporal frequency distributions of the two domains without strongly confounding the analysis with domain-specific encoder generalization. In contrast, WING-LAM is trained specifically on egocentric videos; applying it directly to robot observations could make the estimated spectral distribution partly reflect its out-of-domain generalization behavior rather than the intrinsic temporal characteristics of robot trajectories. Therefore, this analysis is intended to provide a more domain-balanced characterization of the temporal bandwidth gap between egocentric and robot interaction data, rather than to evaluate the cross-embodiment correspondence of the WING-LAM representation itself.

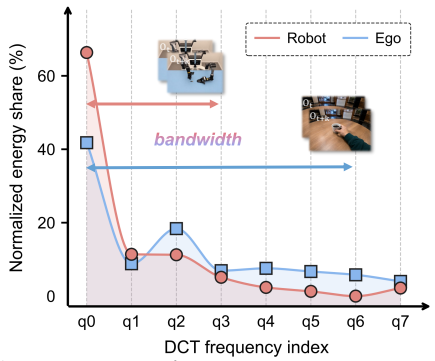


Figure 9: **Bandwidth gap between egocentric and robot data.**

Using the pretrained DreamDojo LAM, we encode ego and robot trajectories with the same physical temporal sampling interval of $0.2\,\mathrm{s}$, forming $H = 8$ latent-action sequences that span $1.6\,\mathrm{s}$. We then apply DCT-II along the temporal dimension. We measure the fraction of spectral energy retained by the lowest $K = 4$ frequency components. The spectral analysis reveals a noticeable difference in temporal bandwidth between egocentric and robot trajectories, with egocentric trajectories exhibiting broader frequency distributions than robot trajectories. Robot trajectories are substantially more concentrated in low-frequency components, whereas egocentric trajectories contain relatively more high-frequency variation. We observe the same qualitative trend when repeating the analysis with WING-LAM, indicating that this bandwidth difference is not specific to the DreamDojo representation. This observation provides additional motivation for examining cross-embodiment correspondence separately across frequency bands.

E.2.3 Action Semantics Across Frequency Bands

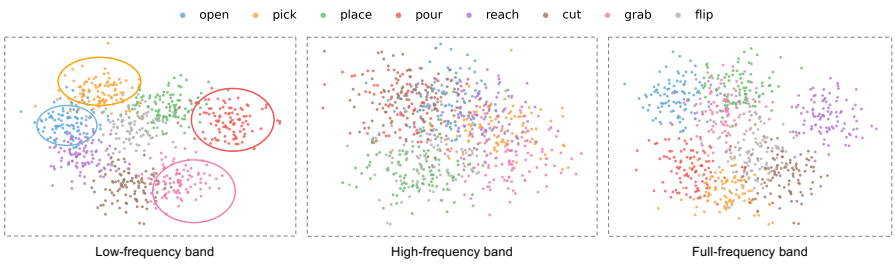


Figure 10: **Action-semantic organization across temporal frequency bands.**

We examine how action-semantic information is organized across temporal frequency bands on the official Ego4D validation split of LARYBench (Nie et al., 2026). For each video, we apply temporal DCT to its 8-step WING-LAM latent trajectory and construct low-frequency ($q_{0:3}$), high-frequency ($q_{4:7}$), and full-spectrum ($q_{0:7}$) representations. Each representation is reduced to 32 dimensions using PCA and evaluated with the same clustering and visualization procedures. We measure the agreement between the resulting clusters and the official action labels using normalized

mutual information (NMI). The low-frequency representation achieves substantially higher NMI (0.55 ± 0.02) than the full-spectrum (0.31 ± 0.01) and high-frequency (0.17 ± 0.05) representations, indicating that action semantics are more clearly organized in the low-frequency latent subspace.

E.3 Additional Policy Analyses

E.3.1 Latency Measurement

We measure policy inference latency on a single NVIDIA RTX 4090 GPU and report both the median and 95th-percentile latency. For evaluation, we follow the real-world deployment setting, using an action chunk size of 32 and 10 denoising steps. As shown in Table 8, our model incurs higher latency than lightweight policy baselines such as GR00T and $\pi_{0.5}$, due to the additional computation introduced by the dense world-modeling architecture. Cosmos-Policy (Kim et al., 2026) and Motus (Bi et al., 2026) require 845.6 ms and 1532.8 ms per policy inference, respectively. Thus, although inference latency remains a limitation of the current implementation, our model remains computationally competitive with generative policies of comparable complexity.

Table 8: **Policy inference latency on a single NVIDIA RTX 4090 GPU.** We report the median and P95 latency over repeated runs. All values are in milliseconds.

Method	Median	P95
$\pi_{0.5}$	90.6	93.8
GR00T	63.4	67.3
FastWAM	229.0	234.2
FastWAM-Joint	510.1	515.4
Cosmos-Policy	845.6	852.3
Motus	1532.8	1551.6
Ours	**500.6**	**517.1**

E.3.2 Ablation on Denoising Steps

The number of denoising steps N controls the amount of iterative refinement at inference time. We conduct this ablation on RoboTwin 2.0 and fix the action chunk size to 32 for all settings. As shown in Table 9, single-step inference achieves an average success rate of only 80.46%, whereas increasing N to 2 substantially improves it to 93.20%, with performance remaining relatively stable for larger N. Among the evaluated settings, $N = 12$ achieves the highest average success rate, with 94.56% and 93.04% on the Clean and Randomized settings, respectively, and is therefore used for the RoboTwin 2.0 result reported in the main table. Notably, nearby settings perform similarly (e.g., 93.55% success rate at $N = 10$), indicating that the overall performance is not sensitive to the precise choice of N once sufficient iterative refinement is used.

Table 9: **Denoising steps.** RoboTwin success rate (%).

N_{denoise}	**Clean**	**Rand.**
1	80.40	80.52
2	93.58	92.82
4	93.78	92.30
6	93.78	92.82
8	94.06	92.56
10	94.14	92.96
12	**94.56**	**93.04**
14	94.28	92.92
16	94.34	92.92

E.3.3 Egocentric Pretraining Efficiency

We investigate how the extent of egocentric pretraining affects subsequent robot policy optimization. All pre-training checkpoints are trained on the same 200h egocentric video corpus, and we select checkpoints obtained after 15k, 30k, 45k, and 60k pretraining steps. Each checkpoint is then used to initialize the robot policy, which is trained on the *Pack Objects* task under the same optimization setting. We use a held-out split of the *Pack Objects* data as the validation set and report the total validation loss of the full model throughout robot training. As shown in Fig. 11, longer egocentric pretraining consistently leads to lower validation loss and faster convergence, indicating that Stage-I pretraining provides a stronger initialization and improves the optimization efficiency of downstream robot policy learning.

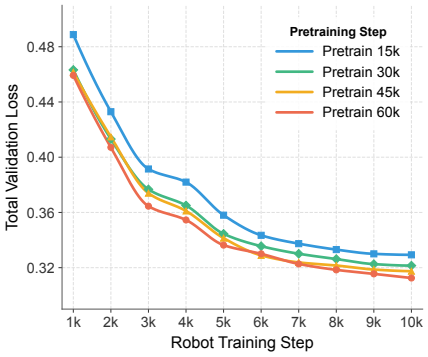


Figure 11: **Egocentric pretraining improves downstream optimization efficiency.**

E.3.4 Downstream Data Efficiency

To further evaluate the effectiveness of egocentric pretraining, we study how performance changes as the amount of downstream robot training data is reduced. We compare models initialized with and without our pretraining under 100%, 50%, and 25% of the downstream training data on LIBERO and the real-world standard setting.

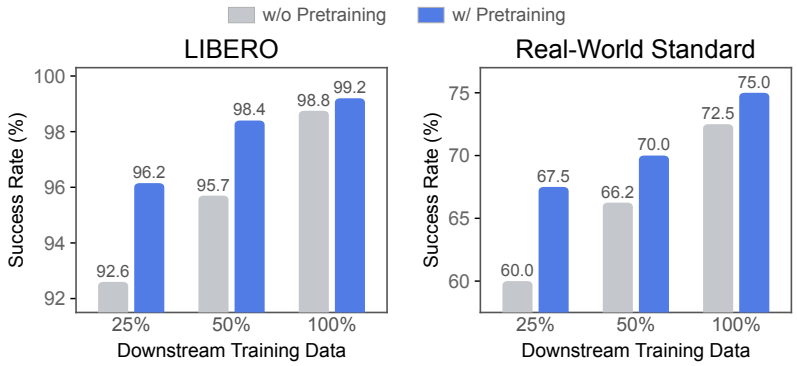


Figure 12: **Downstream data efficiency.** We compare downstream performance with and without egocentric pretraining using different fractions of robot training data. Pretraining consistently improves success rates on both LIBERO and the real-world standard setting, with larger gains as the amount of downstream data decreases.

As shown in Figure 12, egocentric pretraining consistently improves downstream performance on both LIBERO and real-world manipulation. More importantly, the advantage becomes increasingly pronounced as the amount of downstream robot data decreases, with the largest improvement observed in the 25% real-world data setting. These results demonstrate that our pretraining provides transferable knowledge that improves robot-data efficiency, particularly in low-data regimes.

E.4 Full Simulation Results

E.4.1 RoboCasa–GR1 Per-Task Results

Table 10 reports all 24 tabletop tasks. The FastWAM aggregate is our own reproduction.

Table 10: **Full RoboCasa–GR1 tabletop results.** Per-task success rate (%). Published per-task baseline entries are transcribed from the corresponding method papers. † denotes our FastWAM reproduction.

Task	GR00T-N1.6	StarVLA-OFT	DiT4DiT	LDA-1B	FastWAM†	Ours
BottleToCabinetClose	51.5	30.0	48.0	76.0	76.0	66.0
CanToDrawerClose	13.0	76.0	74.0	71.0	78.0	90.0
CupToDrawerClose	8.5	44.0	52.0	41.0	22.0	44.0
MilkToMicrowaveClose	14.0	44.0	50.0	52.0	68.0	58.0
PotatoToMicrowaveClose	41.5	32.0	36.0	41.0	46.0	38.0
WineToCabinetClose	16.5	36.0	42.0	57.0	70.0	62.0
FromCuttingboardToBasket	58.0	50.0	52.0	65.0	38.0	52.0
FromCuttingboardToCardboardbox	46.5	40.0	48.0	69.0	48.0	48.0
FromCuttingboardToPan	68.5	70.0	76.0	75.0	82.0	74.0
FromCuttingboardToPot	65.0	54.0	62.0	61.0	74.0	68.0
FromCuttingboardToTieredbasket	46.5	38.0	50.0	51.0	30.0	44.0
FromPlacematToBasket	58.5	32.0	50.0	53.0	44.0	68.0
FromPlacematToBowl	57.5	58.0	56.0	55.0	44.0	60.0
FromPlacematToPlate	63.0	52.0	32.0	59.0	52.0	52.0
FromPlacematToTieredshelf	28.5	24.0	18.0	24.0	36.0	36.0
FromPlateToBowl	57.0	60.0	56.0	53.0	44.0	68.0
FromPlateToCardboardbox	43.5	50.0	58.0	43.0	40.0	42.0
FromPlateToPan	51.0	66.0	68.0	55.0	42.0	50.0
FromPlateToPlate	78.7	68.0	58.0	61.0	72.0	78.0
FromTrayToCardboardbox	51.5	44.0	38.0	65.0	40.0	70.0
FromTrayToPlate	71.0	56.0	56.0	63.0	68.0	62.0
FromTrayToPot	64.5	62.0	54.0	55.0	50.0	64.0
FromTrayToTieredbasket	57.0	54.0	46.0	51.0	56.0	44.0
FromTrayToTieredshelf	31.5	30.0	38.0	33.0	26.0	46.0
Average	47.6	48.8	50.8	55.4	51.9	57.7

E.4.2 RoboTwin 2.0 Per-Task Results

Table 11 provides the complete 50-task clean/randomized breakdown.

E.5 Full Real-World Generalization Results

Here in this section, we report the full real-world generalization results. All aggregate metrics are computed from unrounded underlying values; rounding is applied only to the final reported number.

Table 11: **RoboTwin 2.0 per-task success rates.** Clean and randomized success rates (%) across 50 tasks, evaluated over 100 trials per task under each setting.

	$\pi_{0.5}$		FastWAM		GigaWorld-Policy		LingBot-VA		Motus		LaWAM		Ours	
Task	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.
Adjust Bottle	100	99	100	100	100	100	90	94	89	93	100	100	100	100
Beat Block Hammer	96	93	99	97	86	86	96	98	95	88	90	93	99	98
Blocks Ranking RGB	92	85	100	100	92	96	99	98	99	97	97	100	99	99
Blocks Ranking Size	49	26	94	98	44	48	94	96	75	63	93	89	87	82
Click Alarmclock	98	89	100	100	100	100	99	100	100	100	100	100	100	100
Click Bell	99	66	100	100	100	100	100	100	100	100	100	100	100	100
Dump Bin Bigbin	92	97	97	96	92	100	89	96	95	91	97	95	95	98
Grab Roller	100	100	100	100	100	100	100	100	100	100	100	100	100	99
Handover Block	66	57	95	81	80	80	99	78	86	73	96	87	92	87
Handover Mic	98	97	99	100	72	72	94	96	78	63	93	98	100	99
Hanging Mug	18	17	58	62	16	12	40	28	38	38	51	43	68	60
Lift Pot	96	85	100	100	98	98	100	99	96	99	100	99	100	100
Move Can Pot	51	55	90	88	76	78	94	97	34	74	98	93	98	98
Move Pillbottle Pad	84	61	100	99	90	90	99	99	93	96	97	90	100	100
Move Playingcard Away	96	84	100	100	78	72	100	99	100	96	100	100	100	100
Move Stapler Pad	56	42	77	64	92	82	91	79	83	85	94	87	91	76
Open Laptop	90	96	98	100	96	98	92	94	95	91	100	100	100	100
Open Microwave	34	77	62	45	74	66	82	86	95	91	41	43	91	76
Pick Diverse Bottles	81	71	80	85	82	70	89	82	90	91	91	88	86	80
Pick Dual Bottles	93	63	100	96	86	86	100	99	96	90	100	95	100	99
Place A2B Left	87	82	95	93	94	88	97	93	88	79	98	91	98	95
Place A2B Right	87	84	93	99	90	92	97	95	91	87	89	94	97	93
Place Bread Basket	77	64	91	93	82	82	97	95	91	94	92	85	100	95
Place Bread Skillet	85	66	90	93	94	90	95	90	86	83	90	83	92	89
Place Burger Fries	94	87	96	99	98	96	97	95	98	98	93	96	97	97
Place Can Basket	62	62	71	69	78	74	81	84	81	76	92	65	79	67
Place Cans Plasticbox	94	84	99	96	100	100	100	99	98	94	100	95	96	99
Place Container Plate	99	95	96	100	98	96	99	97	98	99	100	100	98	99
Place Dual Shoes	75	75	94	88	96	84	94	89	93	87	98	94	91	94
Place Empty Cup	100	99	100	100	90	90	100	100	99	98	99	100	99	99
Place Fan	87	85	96	96	92	94	99	93	91	87	92	93	94	98
Place Mouse Pad	60	39	83	89	88	90	93	96	66	68	91	84	96	93
Place Object Basket	80	76	89	88	90	92	91	88	81	87	92	90	87	84
Place Object Scale	86	80	90	97	88	80	96	95	88	85	95	88	96	96
Place Object Stand	91	85	90	94	100	98	99	96	98	97	92	93	99	96
Place Phone Stand	81	81	97	99	82	72	97	97	87	86	93	94	96	97
Place Shoe	92	93	96	99	98	96	98	98	99	97	100	100	94	97
Press Stapler	87	83	90	97	96	96	85	82	93	98	98	97	87	93
Put Bottles Dustbin	84	79	95	90	72	70	87	91	81	79	94	92	96	89
Put Object Cabinet	80	79	94	89	74	74	85	87	88	71	90	82	96	93
Rotate QRcode	89	87	93	89	90	84	96	91	89	73	94	89	94	90
Scan Object	72	65	89	92	60	64	96	91	67	66	96	90	95	92
Shake Bottle	99	97	100	100	100	100	100	97	100	97	100	100	100	99
Shake Bottle Horizontally	99	99	100	100	100	98	100	99	100	98	100	100	100	99
Stack Blocks Three	91	76	95	97	70	78	99	98	91	95	90	75	95	98
Stack Blocks Two	97	100	100	100	100	94	100	98	100	98	100	97	100	100
Stack Bowls Three	77	71	80	81	70	72	86	83	79	87	90	80	87	87
Stack Bowls Two	95	96	92	98	96	92	94	98	98	98	100	99	93	97
Stamp Seal	79	55	90	94	96	98	96	97	93	92	89	88	95	97
Turn Switch	62	54	61	59	82	84	44	45	84	78	47	56	75	79
Average	**82.74**	**76.76**	**91.88**	**91.78**	**86.36**	**85.04**	**92.90**	**91.50**	**88.66**	**87.02**	**92.64**	**89.80**	**94.56**	**93.04**

Table 12: Detailed real-world generalization results. We report success rate (SR, %) and average task score under different visual and spatial generalization settings.

Task	Generalization Setting	FastWAM		GR00T-N1.7		$\pi_{0.5}$		Ours	
		SR (%)	Score	SR (%)	Score	SR (%)	Score	SR (%)	Score
Battery Insertion	Background	50.0	7.30	40.0	6.00	70.0	8.75	60.0	8.50
	Appearance	10.0	3.60	5.0	3.00	35.0	7.00	40.0	8.00
	Spatial Layout	20.0	4.50	15.0	4.00	60.0	8.50	50.0	8.00
	Avg.	26.7	5.13	20.0	4.33	55.0	8.08	50.0	8.17
Battery Assembly	Background	10.0	3.00	5.0	2.50	10.0	4.60	10.0	5.00
	Appearance	30.0	4.00	15.0	3.00	40.0	5.25	40.0	5.00
	Spatial Layout	20.0	4.50	20.0	4.25	50.0	6.00	45.0	5.25
	Avg.	20.0	3.83	13.3	3.25	33.3	5.28	31.7	5.08
Pack Objects	Background	80.0	8.00	20.0	5.00	85.0	8.95	80.0	8.45
	Appearance	30.0	7.20	20.0	3.50	80.0	9.05	80.0	9.05
	Geometry	60.0	6.60	25.0	5.05	75.0	8.55	85.0	8.95
	Spatial Layout	50.0	5.15	5.0	2.90	80.0	8.75	70.0	8.05
	Avg.	55.0	6.74	17.5	4.11	80.0	8.83	78.8	8.63
Stack Cups	Background	10.0	6.00	5.0	3.50	0.0	5.00	10.0	7.00
	Appearance	25.0	8.00	15.0	7.50	30.0	8.13	40.0	8.50
	Geometry	30.0	7.00	0.0	4.80	40.0	8.15	50.0	8.63
	Spatial Layout	20.0	5.50	40.0	8.50	60.0	8.50	50.0	8.00
	Avg.	21.3	6.63	15.0	6.08	32.5	7.44	37.5	8.03

E.6 Case Study

In this section, we present qualitative real-world examples of WING under both the base and generalization settings.

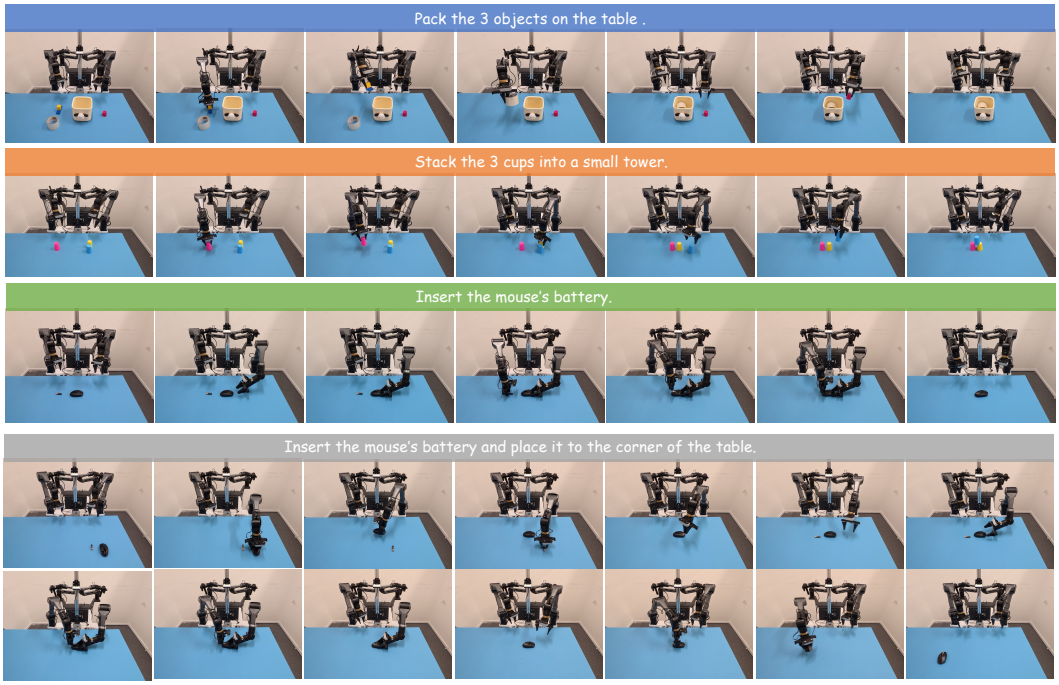


Figure 13: Qualitative results under the standard setting. From top to bottom, we illustrate four base tasks: Pack Objects, Stack Cups, Battery Insertion, and Battery Assembly.



Figure 14: Qualitative results for Pack Objects under generalization settings.

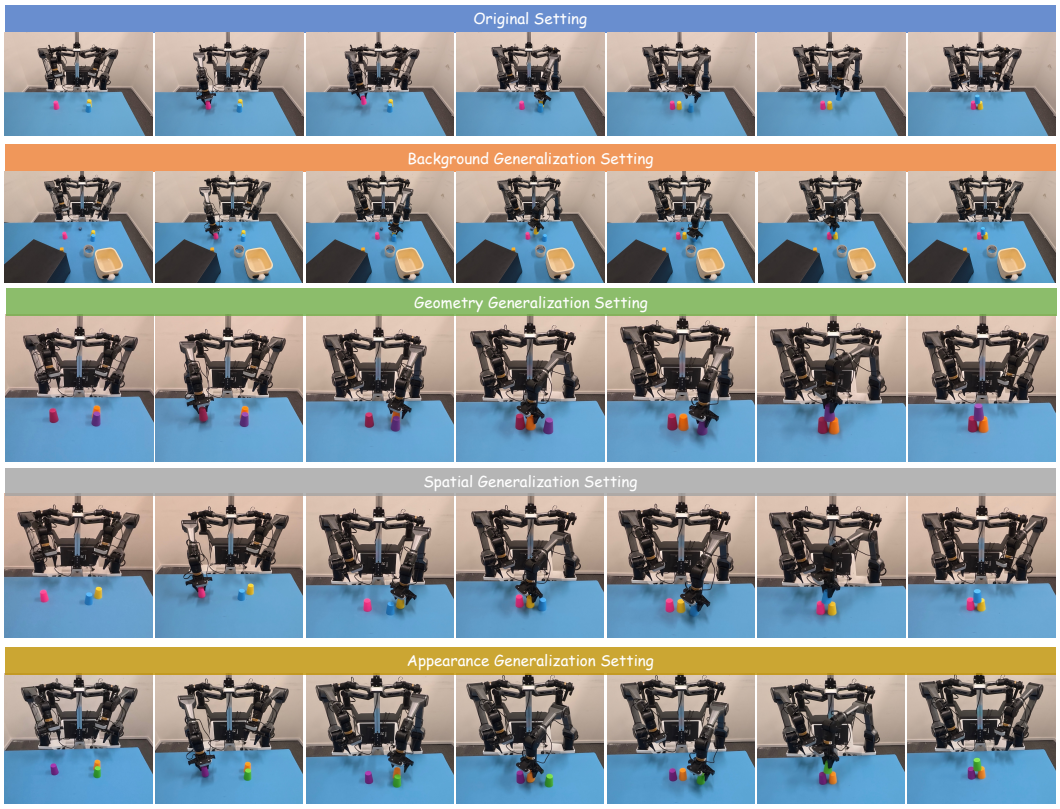


Figure 15: Qualitative results for Stack Cups under generalization settings.

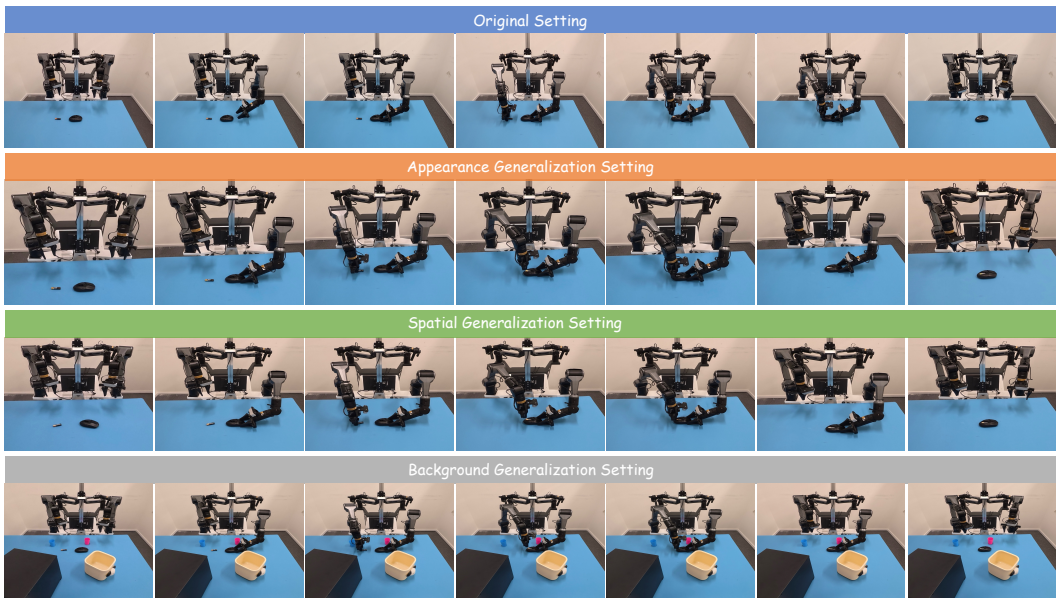


Figure 16: Qualitative results for Battery Insertion under generalization settings.



Figure 17: Qualitative results for Battery Assembly under generalization settings.