INTERNW0-Δ: AN EMBODIED WORLD MODEL BRIDGING PREDICTIVE DYNAMICS AND ACTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

World Action Models (WAMs) have emerged as a promising paradigm for generalist robot manipulation by jointly modeling visual dynamics and action generation. A central challenge is how to effectively integrate complementary priors from large-scale pretrained models—including visual dynamics, scene semantics, and geometric and motion understanding—into a unified framework. To this end, we introduce InternW0-Δ, a unified World Action Model that brings together pretrained visual dynamics, scene-level semantic understanding, 4D geometric and motion priors, and action generation within a directed Mixture-of-Transformers framework. Within the World–Action MoT, a pretrained video expert and an action expert interact under scene-grounded semantic guidance from a frozen VLM, while a pretrained 4D foundation model injects geometric and motion priors through training-only distillation. To further translate predictive visual dynamics into representations directly useful for action prediction, we introduce Causal Imprint, which learns future-relevant scene changes from training-only future supervision and makes these predictive representations directly available to the action expert without requiring future-video rollout at inference. To support large-scale joint training, we construct a heterogeneous corpus spanning robot demonstrations, UMI data, egocentric human demonstrations, and Ego2Robot data. We carefully curate and filter these diverse sources, resulting in over 20K hours of processed training data. We pretrain InternW0-Δ on this heterogeneous corpus and demonstrate strong performance across diverse simulation benchmarks and real-robot platforms. We will open-source the model code and weights, infrastructure, and data-processing pipeline.

1 INTRODUCTION

World Action Models (WAMs) offer a promising approach to generalist robot manipulation by jointly modeling visual dynamics and robot actions (Ye et al., 2026b; Bi et al., 2025). A key motivation behind this formulation is that large-scale video pretraining can provide strong visual and temporal knowledge about how scenes evolve over time. However, the ability to predict future observations does not directly translate into effective robot control. Action generation further requires identifying task-relevant changes, understanding object geometry and motion, and grounding these cues in the current instruction and scene. The central challenge is therefore to transfer predictive knowledge from visual dynamics modeling into representations that are directly useful for action generation, without requiring explicit future generation during online control (Yuan et al., 2026b).

To face this challenge, we introduce InternW0-Δ, a directed world-action model that combines predictive visual dynamics, temporal context, and task-conditioned scene semantics for action generation. A pretrained video expert and an action expert are coupled through a directed Mixture-of-Transformers architecture. The video expert processes a lightweight sparse memory of anchor, recent, and current observations, providing both episode-level context and recent interaction history, while a frozen vision-language model (VLM) supplies task-conditioned scene semantics to the action expert. To make predictive dynamics directly useful for control, Causal Imprint learns future-relevant scene changes from training-only future supervision and makes these representations available to the action expert. In parallel, training-only 4D-aware distillation from a Track4World (Lu et al., 2026) teacher injects geometric and motion priors into the video expert through auxiliary supervision. The directed information flow prevents Causal Imprint and action representations from

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.

accessing realized future observations, allowing InternW0-Δ to directly predict actions at inference without sampling future videos or invoking the distillation branch.

To support large-scale joint training, we curate a heterogeneous corpus spanning robot demonstrations, UMI data, egocentric human demonstrations, and Ego2Robot data, drawing primarily on public datasets. These sources differ substantially in robot embodiment, control space, camera configuration, and temporal convention. We therefore convert them into a canonical state-action representation and apply systematic quality filtering and temporal alignment, resulting in over 20K hours of processed training data.

We also develop complementary infrastructure to support efficient model iteration and online execution. We optimize training and inference through feature caching, compilation, and memory-efficient execution, enabling faster model iteration and asynchronous action-chunk execution. On a physical dexterous-hand system, our runtime achieves a controller-observed round-trip latency of 152.8 ms on a single NVIDIA RTX 5090, a $5.11\times$ speedup over the standard runtime.

We evaluate InternW0-Δ on LIBERO-Plus (Fei et al., 2026), RoboTwin 2.0 (Chen et al., 2026c), EBench (Gao et al., 2026), and RoboDojo (Chen et al., 2026b), spanning different embodiments, task demands, and distribution shifts. Post-trained only on unperturbed demonstrations, InternW0-Δ achieves the best success rates under distribution shift on both LIBERO-Plus (92.8%) and RoboTwin 2.0 Clean2Random (71.9%), while also leading on Clean2Clean (90.0%). On EBench, which targets mobile bimanual manipulation, it obtains the highest overall score of 66.0. On RoboDojo, whose memory, precision, and long-horizon tasks remain challenging for all methods, it achieves the best average success rate of 23.9%, nearly double that of the strongest prior WAM. We further demonstrate real-robot deployment on two gripper-based and two dexterous-hand platforms, adapting the same pretrained checkpoint to their respective control interfaces through post-training.

2 Related Work

Vision–language–action (VLA) models adapt pretrained vision–language backbones for robot control through tokenized actions (Brohan et al., 2023; Kim et al., 2024) or continuous action generation (Black et al., 2025b; NVIDIA et al., 2025). Subsequent efforts expand the breadth of policy learning, including heterogeneous-data learning (Black et al., 2025a; Yuan et al., 2026a), cross-embodiment adaptation (Wu et al., 2026a;b), and modular policy development (StarVLA Community, 2026). These works establish strong semantic and instruction-following foundations for generalist policies.

World action models (WAMs) additionally learn visual dynamics alongside actions. DreamZero (Ye et al., 2026b) leverages pretrained video priors for joint video–action prediction, while Motus (Bi et al., 2025) unifies understanding, video generation, and action modeling. LingBot-VA and its successor explore causal streaming control and native video–action pretraining, respectively (Li et al., 2026b; Zhang et al., 2026b). A central distinction is whether action inference requires future-video generation: Fast-WAM (Yuan et al., 2026b) retains future-video supervision during training while removing it from the action inference path. OpenWAM (Wang et al., 2026d) provides a unified framework for WAM development and deployment.

3 Method

As shown in Figure 1, InternW0-Δ integrates pretrained visual dynamics, task-conditioned scene semantics, and action generation within a directed world-action architecture. A pretrained video expert and an action expert are coupled through a directed Mixture-of-Transformers, while a frozen vision-language model provides scene-grounded task semantics to the action expert.

3.1 Multimodal Context Encoding

Multi-view visual encoding. At each time step τ, InternW0-Δ receives observations from up to K camera views. We denote the k-th view by $o_\tau^{(k)}$ and use a binary indicator $m_\tau^{(k)} \in \{0, 1\}$ to specify whether the corresponding view is available. To support heterogeneous camera configurations across embodiments, valid views are first resized and arranged into a unified visual canvas through an

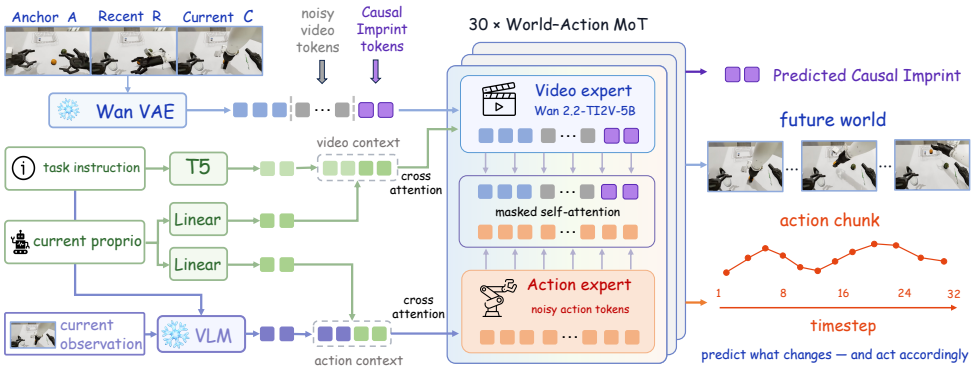


Figure 1: **Architecture overview of InternW0-Δ.** A frozen Wan VAE encodes sparse visual memory comprising anchor (A), recent (R), and current (C) observations. The pretrained Wan2.2-TI2V-5B video expert and the ActionDiT action expert are coupled through 30 directed Mixture-of-Transformers (MoT) blocks. T5 instruction embeddings condition the video expert, while a frozen VLM provides task-conditioned scene semantics to the action expert; current proprioception is independently projected for both experts. Causal Imprint tokens encode change-oriented predictive features from recent and current observations to guide action generation.

embodiment-dependent composition operator Π,

$$x_\tau = \Pi\left(\left\{o_\tau^{(k)}\right\}_{k=1}^{K}, \left\{m_\tau^{(k)}\right\}_{k=1}^{K}\right). \tag{1}$$

The composed observation is then encoded by the pretrained video VAE,

$$z_\tau = E_{\mathrm{VAE}}(x_\tau). \tag{2}$$

where z_τ serves as the visual latent input to the video expert.

Proprioceptive encoding. In addition to visual observations, InternW0-Δ conditions on the current proprioceptive state s_t, expressed in the canonical representation defined in Section A.3.1. The canonical state stores the current joint positions, absolute end-effector (EEF) pose, and gripper or hand configuration in fixed semantic slots. We independently project the canonical state into the embedding spaces of the video and action experts,

$$e_t^v = E_s^v(s_t), \qquad e_t^a = E_s^a(s_t), \tag{3}$$

where E_s^v and E_s^a are separately parameterized state encoders, each implemented as a single linear layer, and e_t^v and e_t^a denote the resulting proprioceptive embeddings for the video and action experts, respectively.

Action encoding. To support heterogeneous embodiments, robot-specific control signals are mapped into the canonical action representation defined in Section A.3.1, following the fixed semantic layout in Table 6. Joint, gripper, and hand actions specify absolute target configurations, whereas EEF actions specify motion relative to the current EEF pose. We denote an action chunk and its validity mask by

$$A_t \in \mathbb{R}^{H_a \times D_a}, \qquad M_A \in \{0,1\}^{H_a \times D_a}, \tag{4}$$

where H_a denotes the action horizon, $D_a = 80$ is the dimensionality of the canonical action space, and M_A indicates the valid timesteps and action dimensions for the active embodiment. The canonical action vectors are projected into the action expert's embedding space through an action encoder,

$$e_t^A = E_A(A_t), \tag{5}$$

where E_A is implemented as a single linear layer and e_t^A denotes the resulting action-token embeddings.

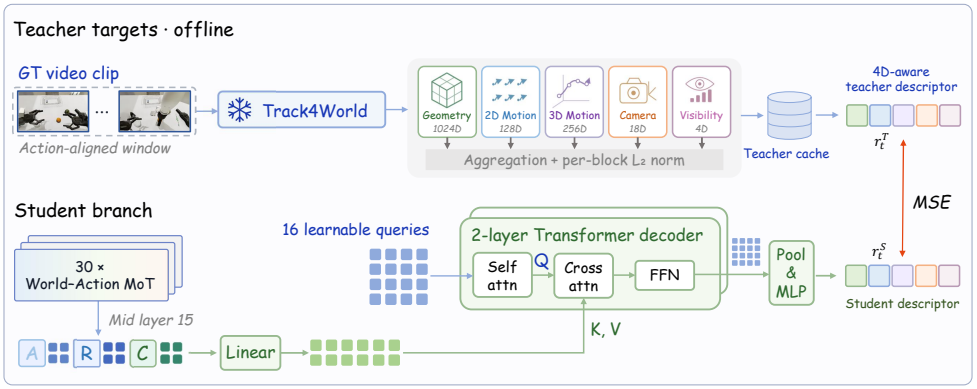


Figure 2: **4D-aware representation distillation.** A query-based student branch aggregates clean A/R/C features from the video expert and aligns them with cached Track4World descriptors via an auxiliary MSE loss, transferring geometric and motion priors without additional inference cost.

Task and scene semantic encoding. The pretrained video expert inherits the T5-based (Raffel et al., 2020) language conditioning pathway from Wan2.2-TI2V-5B (Wan Team, 2025). Given the instruction ℓ, we construct the video-side conditioning sequence as

$$C_t^v = [E_{\mathrm{T5}}(\ell); e_t^v]\,. \tag{6}$$

We introduce an additional frozen vision–language model to complement the T5 pathway with scene-level visual understanding. The VLM jointly processes all valid current views together with their view identities and the instruction,

$$H_t^{\mathrm{VLM}} = f_{\mathrm{VLM}}\left(\left\{\left(o_t^{(k)}, \mathrm{id}(k)\right) \,\middle|\, m_t^{(k)} = 1\right\}, \ell\right). \tag{7}$$

The resulting multimodal tokens provide the action expert with a richer understanding of the current scene while relating this visual context to the task instruction. We combine these features with the action-side proprioceptive embedding as

$$C_t^a = \left[H_t^{\mathrm{VLM}}; e_t^a\right]. \tag{8}$$

which is used to condition action prediction.

3.2 Sparse Memory Context

Motivated by the observation that nearby history is most informative for short-term motion and interaction continuity, whereas distant history mainly serves as a coarse global reference, we equip InternW0-Δ with a lightweight sparse memory that preserves both levels of temporal context. Specifically, InternW0-Δ maintains a sparse visual memory consisting of an episode-level global memory, a short-term memory from the previous action chunk, and the current observation,

$$\mathcal{X}_t = \{x_a,\, x_{t-H_c},\, x_t\}\,, \tag{9}$$

where x_a denotes the composed multi-view observation at the beginning of the episode, x_{t-H_c} provides a reference to the state before the previous action chunk, and x_t is the current observation. Their corresponding video latents, $\{z_a, z_{t-H_c}, z_t\}$, provide sparse episode-level and recent execution context to the video expert without maintaining a dense observation history.

3.3 Causal Imprint

Causal Imprint is motivated by the idea that supervision from future outcomes can leave a training-time imprint on representations formed from the observed context. Instead of exposing realized future observations to the action prediction path, we use them only as supervision, encouraging the model to encode future-relevant scene changes from observations available at inference time.

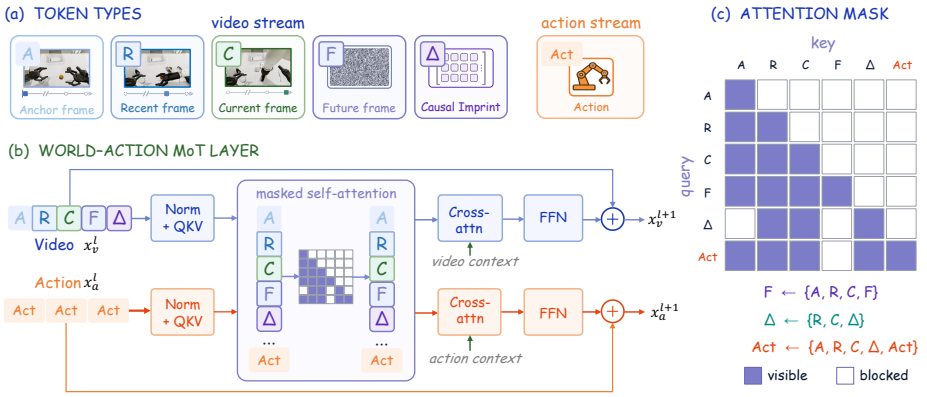


Figure 3: **World-Action MoT layer and attention mask.** (a) The video stream contains anchor (A), recent (R), current (C), future (F), and Causal Imprint (Δ) tokens, while action tokens (Act) form a separate stream. (b) The two experts compute separate query, key, and value projections for joint masked self-attention, followed by expert-specific cross-attention, feed-forward, and residual pathways. (c) Rows and columns index query and key token groups, respectively.

During training, we construct explicit change targets from the clean video latents before noise injection. Given the current latent z_t and future latent slices $\{z_{t+i\rho_v}\}_{i=1}^{T_v}$, we define the adjacent latent differences as

$$\Delta z_{t,i} = z_{t+i\rho_v} - z_{t+(i-1)\rho_v}, \qquad i = 1, \dots, T_v, \tag{10}$$

where $z_{t+0\rho_v} = z_t$. We stack these differences along the temporal dimension to form the supervision target ΔZ_t. These adjacent latent differences encourage Causal Imprint to encode visual changes along the future trajectory. The resulting Causal Imprint representations are made available to the action expert, while the clean future latents are used only to construct training targets. The complementary future-feature alignment is detailed in Section 3.6.

3.4 4D-Aware Representation Distillation

To enrich the video expert with geometric and motion-aware representations, we introduce a training-only distillation objective using a frozen Track4World (Lu et al., 2026) teacher (See Figure 2). For each training sample, the teacher processes a ground-truth video window aligned with the action horizon. We aggregate its geometry and 2D/3D motion features, together with camera-motion and visibility statistics, into a clip-level 4D-aware descriptor $r_t^{\mathrm{T}} \in \mathbb{R}^{1430}$ following the scheme of (Wang et al., 2026a). Each component is independently ℓ_2-normalized before concatenation.

The student branch uses only observations available to the policy at inference. Specifically, we extract the clean-condition hidden tokens corresponding to the anchor, recent, and current frames from the 15th VideoDiT block and linearly project them into the decoder embedding space. We then introduce 16 learnable queries and aggregate the projected visual tokens through a two-layer Transformer decoder. Each layer comprises query self-attention, cross-attention to the projected visual tokens, and a feed-forward network. Finally, the decoded queries are mean-pooled and projected to the teacher feature dimension, producing the student descriptor r_t^{S}.

We align the student descriptor with the cached teacher descriptor using an auxiliary MSE loss, jointly optimized with the original WAM objectives, to distill geometric and motion priors into the video expert. This distillation introduces no additional policy inference cost, as neither the teacher nor the student branch is required at inference.

3.5 Mixture-of-Transformers and Attention Mask

The video and action experts retain separate modality-specific parameters, while exchanging information through masked joint attention. The token groups and a coupled World-Action MoT layer are illustrated in Figure 3(a,b). At each coupled transformer block, the two experts independently

Table 1: Zero-shot robustness on LIBERO-Plus. All values are success rates (%).

Method	Camera	Robot	Language	Light	Background	Noise	Layout	Total ↑
VLA								
π_0 (Black et al., 2025b)	13.8	6.0	58.8	85.0	81.4	79.0	68.9	53.6
π_0-FAST (Pertsch et al., 2025)	65.1	21.6	61.0	73.2	73.2	74.4	68.8	61.6
RIPT-VLA (Tan et al., 2025)	55.2	31.2	77.6	88.4	91.6	73.5	74.2	68.4
OpenVLA-OFT (Kim et al., 2025)	56.4	31.9	79.5	88.7	93.3	75.8	74.2	69.6
StarVLA (StarVLA Community, 2026)	52.5	49.8	88.5	95.7	95.7	73.0	76.9	74.1
VLA-JEPA (Sun et al., 2026)	63.3	67.1	85.4	95.6	93.6	66.3	85.1	79.5
VLAct (Yang et al., 2026b)	73.9	68.4	81.5	96.7	96.7	86.0	83.3	82.6
$\pi_{0.5}$ (Black et al., 2025a)	78.4	73.6	80.8	96.2	94.1	89.0	84.5	84.4
InternVLA-A1.5 (Ma et al., 2026a)	83.1	55.1	86.9	96.4	98.2	<u>95.6</u>	85.2	84.8
Qwen-RobotManip-Context (Yuan et al., 2026a)	<u>89.9</u>	83.9	86.5	**98.6**	**99.9**	**97.9**	87.5	<u>91.4</u>
WAM								
Fast-WAM (Yuan et al., 2026b)	16.4	44.5	68.9	78.2	53.7	37.7	60.7	51.5
OpenWAM-α (Wang et al., 2026d)	33.8	76.1	88.0	97.0	87.1	39.8	77.5	69.2
4D-WAM (Yang et al., 2026a)	45.2	64.3	90.6	94.3	57.7	69.1	79.2	71.0
ST-WAM (Wang et al., 2026b)	55.4	60.1	79.3	93.0	74.2	79.5	74.3	72.8
Faster-WAM (Ma et al., 2026b)	67.9	49.0	<u>92.1</u>	94.3	57.0	82.3	82.7	75.0
JEPA-WAM (Lin et al., 2026)	79.2	59.2	68.2	93.3	94.6	83.6	76.1	79.2
Cosmos-Policy (Kim et al., 2026)	75.8	63.3	81.7	96.5	88.9	92.7	82.2	82.2
ImageWAM (Zhang et al., 2026d)	80.8	50.3	91.4	<u>98.1</u>	85.5	93.8	80.5	83.1
ABot-M0.5 (Chen et al., 2026a)	70.5	<u>87.4</u>	88.6	94.0	89.7	75.5	85.2	83.4
Being-H0.7 (Luo et al., 2026)	82.0	59.0	82.8	97.8	90.0	93.5	**88.5**	84.8
InternW0-Δ	**90.6**	**91.1**	**92.9**	95.8	<u>99.4</u>	94.0	<u>88.4</u>	**92.8**

compute their query, key, and value projections, which are concatenated for a single attention operation:

$$Y = \mathrm{Attn}\left(\begin{bmatrix}Q^v\\Q^a\end{bmatrix}, \begin{bmatrix}K^v\\K^a\end{bmatrix}, \begin{bmatrix}V^v\\V^a\end{bmatrix}; M\right), \qquad (Y^v, Y^a) = \mathrm{Split}(Y). \tag{11}$$

The resulting features are then routed back to their corresponding streams, where the video and action experts continue with their own cross-attention, feed-forward, and residual pathways. Thus, joint attention serves as the communication interface between the two experts without merging their modality-specific parameters. The cross-attention modules use the video-side context C_t^v and action-side context C_t^a defined in Equations (6) and (8), respectively. After the final MoT block, the action decoder uses a single linear layer to map the action expert's output features to D_a-dimensional flow-velocity predictions in the canonical action space.

As illustrated in Figure 3, the attention mask further controls the direction of information flow among the anchor, recent, current, future, Causal Imprint, and action tokens. Future-frame tokens can use the observed context for future prediction, while Causal Imprint and action tokens are prevented from directly accessing the realized future. Instead, the Causal Imprint tokens learn predictive change representations from the recent and current observations under future-derived supervision, and expose these representations to the action stream.

3.6 Training Objectives

We jointly optimize InternW0-Δ for video generation, action generation, Causal Imprint learning, and 4D-aware representation distillation.

Flow matching objectives. Following the flow-matching formulation of the pretrained video model, for a target variable y, we sample Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$ and a flow time $\sigma \in (0, 1)$, and construct

$$y^\sigma = (1-\sigma)y + \sigma\epsilon. \tag{12}$$

The corresponding flow-matching objective is

$$\mathcal{L}_{\mathrm{FM}}(y) = \mathbb{E}_{\epsilon,\sigma}\left[\|f_\theta(y^\sigma, \sigma) - (\epsilon - y)\|_2^2\right]. \tag{13}$$

We apply this objective to both future video generation and action prediction:

$$\mathcal{L}_{\mathrm{video}} = \mathcal{L}_{\mathrm{FM}}(Z_t^F), \qquad \mathcal{L}_{\mathrm{action}} = \mathcal{L}_{\mathrm{FM}}(A_t), \tag{14}$$

where Z_t^F denotes the future video latent sequence and A_t denotes the target action chunk.

Table 2: Evaluation results on RoboTwin2.0 (Chen et al., 2026c) Clean2Random under clean-only training. Success rates (%) are reported on Clean2Clean and Clean2Random.

Method	Clean2Clean ↑	Clean2Random ↑	Overall ↑
VLA			
GR00T-N1.7 (NVIDIA, 2026)	43.6	20.7	32.2
StarVLA (StarVLA Community, 2026)	58.1	10.6	34.4
X-VLA (Zheng et al., 2026)	68.0	20.9	44.5
Spatial Forcing (Li et al., 2026a)	77.2	26.7	52.0
ABot-M0 (Yang et al., 2026c)	70.7	36.0	53.4
$\pi_{0.5}$ (Black et al., 2025a)	73.1	47.9	60.5
GigaBrain-0.7 (GigaBrain Team et al., 2026)	66.8	67.9	67.4
Qwen-RobotManip-Context (Yuan et al., 2026a)	84.7	<u>69.4</u>	<u>77.1</u>
WAM			
AHA-WAM (Cai et al., 2026a)	64.3	3.2	33.8
Fast-WAM (Yuan et al., 2026b)	77.8	1.9	39.9
X-WAM (Guo et al., 2026)	70.0	25.8	47.9
4D-WAM (Yang et al., 2026a)	81.5	41.8	61.7
OpenWAM-α (Wang et al., 2026d)	<u>89.4</u>	48.7	69.0
InternW0-Δ	**90.0**	**71.9**	**81.0**

Causal Imprint objectives. We train Causal Imprint with the two complementary objectives. First, the final-layer Causal Imprint representation h_t^{Δ} is directly supervised by the adjacent clean-latent differences ΔZ_t defined in Equation (10). The corresponding objective is

$$\mathcal{L}_{\Delta} = \left\| h_t^{\Delta} - \Delta Z_t \right\|_2^2 . \tag{15}$$

This objective provides direct supervision for Causal Imprint, encouraging it to encode changes associated with the subsequent evolution of the scene. In addition, video diffusion transformers have been shown to develop rich semantic, temporal, and physical representations in their intermediate features (King et al., 2026; Esmati et al., 2026). We therefore introduce a representation-alignment objective to transfer such predictive information from the video expert to Causal Imprint. Specifically, each Causal Imprint token at block $\ell_s = 8$ is aligned with the spatially corresponding feature of the terminal future slice at block $\ell_t = 20$:

$$\mathcal{L}_{\text{align}} = 1 - \cos\left(h_{b,p}^{\Delta,\ell_s}, \text{sg}\left[h_{b,p}^{F,\ell_t} \right] \right), \tag{16}$$

where $\text{sg}[\cdot]$ denotes stop-gradient.

4D-aware representation distillation objective. We align the student descriptor r_t^{S} with the cached teacher 4D-aware descriptor r_t^{T} using an auxiliary mean squared error loss:

$$\mathcal{L}_{\text{4D}} = \frac{1}{D} \left\| r_t^{\text{S}} - \text{sg}\left[r_t^{\text{T}} \right] \right\|_2^2 , \tag{17}$$

where $D = 1430$ is the descriptor dimension and $\text{sg}[\cdot]$ denotes stop-gradient. The loss is averaged over training samples with valid teacher descriptors. This auxiliary supervision transfers geometric and motion priors to the video expert by backpropagating through the student branch and the selected VideoDiT hidden tokens, while the teacher remains frozen.

Overall objective. The complete training objective is

$$\mathcal{L} = \lambda_v \mathcal{L}_{\text{video}} + \lambda_a \mathcal{L}_{\text{action}} + \lambda_{\Delta} \mathcal{L}_{\Delta} + \lambda_{\text{align}} \mathcal{L}_{\text{align}} + \lambda_{\text{4D}} \mathcal{L}_{\text{4D}}. \tag{18}$$

4 Experiments

4.1 Training and Evaluation Setup

We first pretrain InternW0-Δ on heterogeneous embodied data and then post-train separate copies of the resulting checkpoint for each target benchmark or robot platform. The pretraining mixture comprises robot demonstrations (80%), Ego2Robot data (10%), UMI demonstrations (8%), and egocentric demonstrations (2%), with all sources mapped to a canonical 80-dimensional state–action representation. These percentages specify mixture sampling probabilities rather than proportions of recorded hours. Data processing and training configurations are provided in Sections A.3 and A.8.

Table 3: Evaluation results on EBench (Gao et al., 2026)

Method	Table Top		Simple PnP		Long Horizon		Overall	
	SR ↑	Score ↑	SR ↑	Score ↑	SR ↑	Score ↑	SR ↑	Score ↑
VLA								
StarVLA-OFT (StarVLA Community, 2026)	–	–	–	–	–	–	0.0	0.2
π_0 (Black et al., 2025b)	15.7	30.0	35.0	39.0	17.0	41.0	23.6	37.0
X-VLA (Zheng et al., 2026)	8.6	24.0	50.0	54.0	6.2	25.0	23.7	36.0
InternVLA-A1 (Cai et al., 2026b)	4.3	11.0	43.0	47.0	17.9	46.0	23.9	36.0
$\pi_{0.5}$ (Black et al., 2025a)	12.9	32.0	45.0	50.0	18.1	39.0	27.1	41.0
GigaBrain-0.7 (GigaBrain Team et al., 2026)	–	–	–	–	–	–	33.3	46.0
Qwen-RobotManip (Yuan et al., 2026a)	**50.0**	**70.0**	56.5	60.0	29.9	55.0	45.6	60.0
WAM								
Fast-WAM (Yuan et al., 2026b)	–	–	–	–	–	–	4.7	7.6
OpenWAM-α (Wang et al., 2026d)	30.0	44.2	**67.5**	**72.0**	<u>44.3</u>	<u>72.6</u>	**49.4**	<u>64.7</u>
InternW0-Δ	<u>33.6</u>	<u>55.2</u>	<u>60.0</u>	<u>64.3</u>	**49.4**	**76.5**	<u>49.2</u>	**66.0**

4.2 Main Results

We evaluate on LIBERO-Plus (Fei et al., 2026), RoboTwin 2.0 (Chen et al., 2026c), EBench (Gao et al., 2026), and RoboDojo (Chen et al., 2026b). LIBERO-Plus evaluates transfer from LIBERO training to perturbed environments, while RoboTwin evaluates clean-only training under Clean2Clean and Clean2Random settings. Benchmark-specific post-training details are provided in Section A.8.2. Baseline values are taken from the corresponding papers or public benchmark reports.

LIBERO-Plus. After post-training on LIBERO (Liu et al., 2023), we evaluate zero-shot robustness on LIBERO-Plus (Fei et al., 2026). As shown in Table 1, InternW0-Δ achieves the highest overall success rate among the compared methods at 92.8%, exceeding Qwen-RobotManip-Context by 1.4 percentage points. It leads the Camera, Robot, and Language categories, with the largest margin over the best category-wise baseline under robot perturbations. A detailed perturbation-wise analysis is provided in Section A.10.2.

RoboTwin 2.0. Under clean-only training, InternW0-Δ achieves 90.0% on Clean2Clean and 71.9% on Clean2Random in RoboTwin 2.0 (Chen et al., 2026c) (Table 2). The Clean2Random result exceeds Qwen-RobotManip-Context and OpenWAM-α by 2.5 and 23.2 percentage points, respectively, demonstrating strong transfer from clean demonstrations to randomized evaluation conditions. The relationship between clean performance and robustness is further examined in Section A.10.2.

EBench. On EBench (Gao et al., 2026), which evaluates mobile bimanual manipulation across Table Top, Simple PnP, and Long Horizon tasks, InternW0-Δ obtains the highest overall Score of 66.0 with a success rate of 49.2% (Table 3). Its strength is most evident on Long Horizon tasks, where both metrics lead the compared methods, although its overall success rate remains slightly below OpenWAM-α (49.4%). Category-level differences and the distinction between Score and success rate are discussed in Section A.10.2.

RoboDojo. On RoboDojo (Chen et al., 2026b), InternW0-Δ achieves the highest average success rate and Score among the compared methods, reaching 23.91% and 30.77, respectively (Table 4). The average success rate exceeds the strongest compared WAM, OpenWAM-α, by 11.99 percentage points. The results highlight strengths in Gen-Std and Precision, while Memory and Open tasks remain challenging relative to the leading methods in these categories. A capability-wise analysis is provided in Section A.10.2.

4.3 Ablation Study

We conduct cumulative component ablations on LIBERO-Plus to investigate the contribution of the main components in InternW0-Δ. Each configuration in Table 5 is trained independently under the same data, optimization, and evaluation protocol, with component changes applied cumulatively rather than through continued training. Starting from the baseline, we progressively introduce Sparse Memory Context (SMC), VLM, Causal Imprint (CI), representation alignment, and 4D-aware representation distillation.

Table 4: Evaluation results on RoboDojo (Chen et al., 2026b) across six evaluation categories.

Method	Gen-Std		Gen-Rand		Precision		Long-Horizon		Memory		Open		Avg	
	SR ↑	Score ↑	SR ↑	Score ↑	SR ↑	Score ↑	SR ↑	Score ↑	SR ↑	Score ↑	SR ↑	Score ↑	SR ↑	Score ↑
VLA														
StarVLA-α (StarVLA Community, 2026)	5.00	7.54	0.00	0.33	4.33	9.90	6.50	14.15	2.44	3.34	0.58	0.68	3.24	6.40
X-VLA (Zheng et al., 2026)	12.00	17.90	1.00	3.04	12.00	18.32	9.75	16.53	3.56	4.76	0.50	0.55	6.52	10.13
$\pi_{0.5}$ (Black et al., 2025a)	15.00	20.93	1.00	5.82	5.50	12.40	14.67	23.54	4.56	5.78	1.67	1.98	6.91	11.41
Spatial Forcing (Li et al., 2026a)	15.00	21.25	4.00	6.98	10.58	17.33	14.58	23.26	4.11	5.43	1.58	1.78	8.04	12.38
Hy-Embodied-0.5-VLA (Zhang et al., 2026a)	17.00	21.98	0.00	1.57	8.00	13.81	14.92	25.74	12.11	13.37	0.58	0.65	8.80	13.07
Xiaomi-Robotics-1 (Team et al., 2026)	28.00	35.65	6.00	11.44	18.83	26.69	23.67	38.39	6.56	7.81	3.58	3.94	13.93	20.07
Galaxea G0.5 (Liu et al., 2026)	20.00	26.74	6.00	11.16	20.42	28.25	**32.25**	44.12	7.33	8.61	1.58	1.73	14.88	20.23
DM0.5 (Dexmal, 2026)	18.00	23.49	4.00	8.06	16.75	24.82	19.50	33.70	**47.44**	**47.74**	2.08	2.43	19.34	24.90
LLM as Policy														
GPT-6 Astra (Zhang et al., 2026c)	32.67	35.32	**28.33**	**31.40**	4.00	12.65	8.25	21.45	38.67	43.04	**31.00**	**34.36**	22.48	28.97
WAM														
Fast-WAM (Yuan et al., 2026b)	2.00	4.33	0.00	0.34	0.00	1.96	5.17	9.14	3.44	3.55	0.42	0.42	2.03	3.48
AHA-WAM (Cai et al., 2026a)	6.00	10.32	0.00	1.26	2.42	5.86	2.67	8.61	2.78	2.97	0.83	0.88	2.39	4.82
GigaWorld-Policy (Ye et al., 2026a)	6.00	10.28	0.00	0.41	1.83	6.15	8.92	15.51	2.22	3.46	0.50	0.54	3.27	6.20
X-WAM (Guo et al., 2026)	5.00	11.24	1.00	3.54	1.83	6.72	9.08	17.47	4.67	6.32	0.25	0.57	3.83	7.69
OpenWAM-α (Wang et al., 2026d)	25.56	33.16	4.11	8.26	9.25	18.45	25.33	34.93	9.11	10.41	1.08	1.41	11.92	17.18
InternW0-Δ	**33.78**	**40.98**	11.78	19.19	**23.25**	**31.98**	29.33	**46.29**	34.00	34.67	10.17	10.84	**23.91**	**30.77**

Table 5: Cumulative ablation on LIBERO-Plus (Fei et al., 2026). All variants are trained independently from scratch under the same settings, with SMC, vision-language conditioning, CI, $\mathcal{L}_{\text{align}}$, and $\mathcal{L}_{\text{4D}}$ added cumulatively. Values are overall success rates (%).

Method	Success rate (%) ↑
Baseline	49.59
Baseline + SMC	53.47
Baseline + SMC + Qwen3.5-2B	60.37
Baseline + SMC + RynnBrain1.1-2B	69.08
Baseline + SMC + RynnBrain1.1-2B + CI	70.80
Baseline + SMC + RynnBrain1.1-2B + CI + $\mathcal{L}_{\text{align}}$	76.45
Baseline + SMC + RynnBrain1.1-2B + CI + $\mathcal{L}_{\text{align}}$ + $\mathcal{L}_{\text{4D}}$	**78.37**

As shown in Table 5, adding Sparse Memory Context improves success from 49.59% to 53.47%, suggesting that sparse historical observations provide useful temporal context beyond the current observation. Introducing Qwen3.5-2B increases success to 60.37%, while replacing it with RynnBrain1.1-2B further raises it to 69.08%. This progression highlights the importance of the vision–language representations used for action conditioning.

Causal Imprint further improves success to 70.80% with latent-difference supervision and to 76.45% after adding future-feature alignment. The additional gain from alignment suggests that feature-level supervision provides useful predictive information beyond directly regressing latent changes. Finally, 4D-aware distillation raises success to 78.37%, suggesting that geometric and motion supervision remains beneficial after semantic conditioning and predictive alignment are introduced.

These improvements are measured cumulatively and are therefore conditional on the preceding components, rather than isolated estimates of each component's contribution. Detailed component-wise analysis is provided in Section A.12, with additional teacher and action-injection comparisons in Table 15.

5 Conclusion

We present InternW0-Δ, a world-action model that connects predictive visual dynamics with task-conditioned action generation. Causal Imprint learns future-relevant changes, while sparse memory and training-only 4D-aware distillation provide temporal, geometric, and motion context. Heterogeneous pretraining supports adaptation across simulation benchmarks and real-robot platforms with different control interfaces. The results show strong aggregate performance, while memory-intensive and open-ended tasks remain challenging. We plan to release the code, checkpoints, training recipes, and data-processing tools, subject to source-data permissions.

AI USE STATEMENT

AI models are part of the research pipeline. Pretrained video, language, and vision–language models supply the model initialization and conditioning described in Section 3. Large language and vision–language models are used for instruction screening and video–instruction consistency filtering in Section A.3. Learned segmentation and video-inpainting models are used in Ego2Robot synthesis. The additional selected-task policy experiment in Section A.12.2 uses GPT-6 Astra for execution-time corrections. During manuscript preparation, large language models (LLMs) were used for translation, language refinement and grammar checking.

ETHICS STATEMENT

The work combines heterogeneous robot and human egocentric data and includes human-to-robot visual synthesis. Such data can contain identifying imagery, and reuse or redistribution must respect source permissions and applicable dataset licenses; visual conversion does not remove these obligations. Automated data filtering may introduce coverage biases or retain errors, so the reported capabilities should not be interpreted as guarantees of equitable or reliable performance across people, environments, and embodiments. The real-robot experiments are research demonstrations, not certification for unsupervised physical deployment. Broader deployment requires application-specific risk assessment, appropriate human oversight, and safeguards for manipulation failures.

REPRODUCIBILITY STATEMENT

Sections 3 and 4.1 define the model and training recipe. Sections A.3 and A.13 document representation, filtering, data statistics, and retargeting. Sections A.8 and A.9 give optimization, adaptation, caching, and deployment settings. Sections A.9.2 and A.10 to A.12 provide deployment details, complete benchmark results, real-robot experiments, and additional ablations. Release plans include data-processing code, filtered-data indices, training and evaluation code, checkpoints, and deployment utilities, subject to source-data redistribution conditions. Planned releases are not asserted to be available with this manuscript.

REFERENCES

Arthur Allshire, Himanshu Gaurav Singh, Ritvik Singh, Adam Rashid, Hongsuk Choi, David McAllister, Justin Yu, Yiyuan Chen, Huang Huang, Pieter Abbeel, Xi Chen, Rocky Duan, Phillip Isola, Jitendra Malik, Fred Shentu, Guanya Shi, Philipp Wu, and Angjoo Kanazawa. Scalable behavior cloning with open data, training, and evaluation, 2026. URL https://arxiv.org/abs/2606.27375.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew J. Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-Supervised Video Models Enable Understanding, Prediction and Planning. *arXiv preprint arXiv:2506.09985*, 2025. URL https://arxiv.org/abs/2506.09985.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting Feature Prediction for Learning Visual Representations from Video. *arXiv preprint arXiv:2404.08471*, 2024. URL https://arxiv.org/abs/2404.08471.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A Unified Latent Action World Model. *arXiv preprint arXiv:2512.13030*, 2025. URL https://arxiv.org/abs/2512.13030.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy

Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 2025a. URL https://proceedings.mlr.press/v305/black25a.html.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Proceedings of Robotics: Science and Systems*, 2025b. doi: 10.15607/RSS.2025.XXI.010. URL https://www.roboticsproceedings.org/rss21/p010.html.

Kevin Black, Manuel Y. Galliker, and Sergey Levine. Real-time execution of action chunking flow policies, 2025c. URL https://arxiv.org/abs/2506.07339.

Kevin Black, Allen Z. Ren, Michael Equi, and Sergey Levine. Training-time action conditioning for efficient real-time chunking, 2025d. URL https://arxiv.org/abs/2512.05964.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023. URL https://arxiv.org/abs/2307.15818.

Remi Cadene, Simon Alibert, Francesco Capuano, Michel Aractingi, Adil Zouitine, Pepijn Kooijmans, Jade Choghari, Martino Russi, Caroline Pascal, Steven Palma, Mustafa Shukor, Jess Moss, Alexander Soare, Dana Aubakirova, Quentin Lhoest, Quentin Gallouédec, and Thomas Wolf. LeRobot: An Open-Source Library for End-to-End Robot Learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2602.22818.

Jisong Cai, Long Ling, Shiwei Chu, Zhongshan Liu, Jiayue Kang, Zhixuan Liang, Wenjie Xu, Yinan Mao, Weinan Zhang, Xiaokang Yang, Ru Ying, Ran Zheng, and Yao Mu. AHA-WAM: Asynchronous Horizon-Adaptive World-Action Modeling with Observation-Guided Context Routing. *arXiv preprint arXiv:2606.09811*, 2026a. URL https://arxiv.org/abs/2606.09811.

Junhao Cai, Zetao Cai, Jiafei Cao, Yilun Chen, Zeyu He, Lei Jiang, Hang Li, Hengjie Li, Yang Li, Yufei Liu, et al. InternVLA-A1: Unifying understanding, generation and action for robotic manipulation. *arXiv preprint arXiv:2601.02456*, 2026b. URL https://arxiv.org/abs/2601.02456.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. SAM 3: Segment Anything with Concepts, 2025. URL https://arxiv.org/abs/2511.16719. Version Number: 2.

Ronghan Chen, Yandan Yang, Zuojin Tang, Dongjie Huo, Tong Lin, Haoning Wu, Haoyun Liu, Yuzhi Chen, Lulu Zheng, Botai Yuan, Tianlun Li, Mingxin Wang, Dekang Qi, Bin Hu, Wei Mei, Yuze Xuan, Haolong Yang, Yanqing Zhu, Mu Xu, Zhiheng Ma, and Xinyuan Chang. ABot-M0.5: Unified Mobility-and-Manipulation World Action Model. *arXiv preprint arXiv:2607.00678*, 2026a. URL https://arxiv.org/abs/2607.00678.

Tianxing Chen, Yue Chen, Zixuan Li, Junyuan Tang, Kailun Su, Haoran Lu, Weijie Wan, Baijun Chen, Songling Liu, Haowen Yan, Honghao Su, Zhiyang Dou, Kaixuan Wang, Dandan Zhang, Yunze Liu, Yan Qin, Qiwei Liang, Qiwei Wu, Zijian Lin, Wenwei Lin, Yuran Wang, Minghua He, Tianshu Wu, Ruihai Wu, Jingquan Zhou, Kai-Chong Lei, Haibao Yu, Yuanfeng Ji, Weiyang Jin, Guanyu Lin, Xiaofan Li, Qi Xiong, Renjing Xu, Zhongyu Li, Wenhao Chai, Enze Xie, Ziwei Wang, Yao Mu, Hao Dong, Wojciech Matusik, Mingyu Ding, Wenbo Ding, Ping Luo, and Masayoshi Tomizuka. RoboDojo: A Unified Sim-and-Real Benchmark for Comprehensive Evaluation of Generalist Robot Manipulation Policies. *arXiv preprint arXiv:2607.04434*, 2026b. URL https://arxiv.org/abs/2607.04434.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. RoboTwin 2.0: A Scalable Data Generator and Benchmark with Strong Domain Randomization for Robust Bimanual Robotic Manipulation. In *Proceedings of the International Conference on Machine Learning*, 2026c. URL https://arxiv.org/abs/2506.18088.

AgiBot World Colosseum contributors. Agibot world colosseum. https://github.com/OpenDriveLab/AgiBot-World, 2024.

InternData-A1 contributors. Interndata-a1. https://github.com/InternRobotics/InternManip, 2025.

LiteGen contributors. Litegen. https://github.com/DeepLink-org/LiteGen, 2026.

Ronghao Dang, Jiayan Guo, Bohan Hou, Sicong Leng, Kehan Li, Xin Li, Jiangpin Liu, Yunxuan Mao, Zhikai Wang, Yuqian Yuan, Minghao Zhu, Xiao Lin, Yang Bai, Qian Jiang, Yaxi Zhao, Minghua Zeng, Junlong Gao, Yuming Jiang, Jun Cen, Siteng Huang, Liuyi Wang, Wenqiao Zhang, Chengju Liu, Jianfei Yang, Shijian Lu, and Deli Zhao. Rynnbrain: Open embodied foundation models, 2026. URL https://arxiv.org/abs/2602.14979.

Dexmal. DM0.5: From the Lab to the Open World. Technical blog and model release, 2026. URL https://www.dexmal.com/blog/dm0.5?lang=en-US. Released 2026-07-09. Accessed: 2026-09-18.

Parsa Esmati, Somjit Nath, Katja Hofmann, Derek Nowrouzezahrai, Samira Ebrahimi Kahou, and Majid Mirmehdi. The invisible hand of physics: When video diffusion models know more than they show, 2026.

Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Chenxi Wang, Junbo Wang, Haoyi Zhu, and Cewu Lu. Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 653–660. IEEE, 2024.

Haoquan Fang, Jiafei Duan, Donovan Clay, Sam Wang, Shuo Liu, Weikai Huang, Xiang Fan, Wei-Chuan Tsai, Shirui Chen, Yi Ru Wang, Shanli Xing, Jaemin Cho, Jae Sung Park, Ainaz Eftekhar, Peter Sushko, Karen Farley, Angad Wadhwa, Cole Harrison, Winson Han, Ying-Chun Lee, Eli VanderBilt, Rose Hendrix, Suveen Ellawela, Lucas Ngoo, Joyce Chai, Zhongzheng Ren, Ali Farhadi, Dieter Fox, and Ranjay Krishna. Molmoact2: Action reasoning models for real-world deployment, 2026. URL https://arxiv.org/abs/2605.02881.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: A Progressive Robustness Benchmark for Visual-Language-Action Models. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Fei_LIBERO-Plus_A_Progressive_Robustness_Benchmark_for_Visual-Language-Action_Models_CVPR_2026_paper.html.

Fourier ActionNet Team and Yao Mu. Actionnet: A dataset for dexterous bimanual manipulation, 2025. URL https://action-net.org/.

Ning Gao, Jinliang Zheng, Xing Gao, Haoxiang Ma, Hanqing Wang, Yukai Wang, Jiantong Chen, Zanxin Chen, Shujie Zhang, Mingda Jia, Xuekun Jiang, Zihou Zhu, Xinyu Li, Shuai Wang, Hao Li, Wenzhe Cai, Yuqiang Yang, Xudong Xu, Zhaoyang Lyu, Yao Mu, Tai Wang, Jiangmiao Pang, Jia Zeng, Weinan Zhang, and Chunhua Shen. EBench: Elemental Diagnosis of Generalist Mobile Manipulation Policies. *arXiv preprint arXiv:2606.18239*, 2026. URL https://arxiv.org/abs/2606.18239.

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

Jun Guo, Qiwei Li, Peiyan Li, Zilong Chen, Nan Sun, Yifei Su, Heyun Wang, Yuan Zhang, Xinghang Li, and Huaping Liu. Unified 4D World Action Modeling from Video Priors with Asynchronous Denoising. *arXiv preprint arXiv:2604.26694*, 2026. URL https://arxiv.org/abs/2604.26694.

Ryan Hoque, Peide Huang, David J. Yoon, Mouli Sivapurapu, and Jian Zhang. EgoDex: Learning dexterous manipulation from large-scale egocentric video, 2025. URL https://arxiv.org/abs/2505.11709.

Chengkai Hou, Kun Wu, Jiaming Liu, Zhengping Che, Di Wu, Fei Liao, Guangrun Li, Jingyang He, Qiuxuan Feng, Zhao Jin, Chenyang Gu, Zhuoyang Liu, Nuowei Han, Xiangju Mi, Yaoxu Lv, Yankai Fu, Gaole Dai, Langzhe Gu, Tao Li, Yuheng Zhang, Yixue Zhang, Xinhua Wang, Shichao Fan, Meng Li, Zhen Zhao, Ning Liu, Zhiyuan Xu, Pei Ren, Junjie Ji, Haonan Liu, Kuan Cheng, Shanghang Zhang, and Jian Tang. Robomind 2.0: A multimodal, bimanual mobile manipulation dataset for generalizable embodied intelligence, 2025. URL https://arxiv.org/abs/2512.24653.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video Prediction Policy: A Generalist Robot Policy with Predictive Visual Representations. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 24328–24346. PMLR, 2025. URL https://proceedings.mlr.press/v267/hu25g.html.

Boden Intelligence, Junpu Innovation Center, and Shanghai Jiao Tong University MINT Lab. Rw-rl dataset: Real-world reinforcement learning dataset. https://huggingface.co/datasets/MINT-SJTU/RW-RL-Dataset, 2026.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An Open-Source Vision-Language-Action Model. *arXiv preprint arXiv:2406.09246*, 2024. URL https://arxiv.org/abs/2406.09246.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-Tuning Vision-Language-Action Models: Optimizing Speed and Success. In *Proceedings of Robotics: Science and Systems*, 2025.

doi: 10.15607/RSS.2025.XXI.017. URL https://www.roboticsproceedings.org/rss21/p017.html.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos Policy: Fine-Tuning Video Models for Visuomotor Control and Planning. *arXiv preprint arXiv:2601.16163*, 2026. URL https://arxiv.org/abs/2601.16163.

Michael King, Aravindh Mahendran, Matthew Koichi Grimes, Fedor Kitashov, Adham Elarabawy, Pedro Velez, Maks Ovsjanikov, and Viorica Pătrăucean. Gen4u: Unifying video generation and understanding via diffusion, 2026.

Vikash Kumar, Rutav Shah, Gaoyue Zhou, Vincent Moens, Vittorio Caggiano, Jay Vakil, Abhishek Gupta, and Aravind Rajeswaran. Robohive – a unified framework for robot learning. In *NeurIPS: Conference on Neural Information Processing Systems*, 2023. URL https://sites.google.com/view/robohive.

Zihang Lai, Eldar Insafutdinov, Edgar Sucar, and Andrea Vedaldi. CoWTracker: Tracking by warping instead of correlation. *arXiv preprint arXiv:2602.04877*, 2026.

Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial Forcing: Implicit Spatial Representation Alignment for Vision-Language-Action Model. In *International Conference on Learning Representations*, 2026a. URL https://arxiv.org/abs/2510.12276.

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

Yihan Lin, Jiawei He, Shifeng Bao, Chen Zhao, Yang Li, Xiaobo Wang, Yan Wang, Cheng Chi, and Jing Zhang. JEPA-WAM: Learning Vision-Language-Action Policies with Joint-Embedding World Modeling. *arXiv preprint arXiv:2608.09381*, 2026. URL https://arxiv.org/abs/2608.09381.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8c3c666820ea055a77726d66fc7d447f-Abstract-Datasets_and_Benchmarks.html.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.

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

Jiahao Lu, Jiayi Xu, Wenbo Hu, Ruijie Zhu, Chengfeng Zhao, Sai-Kit Yeung, Ying Shan, and Yuan Liu. Track4world: Feedforward world-centric dense 3d tracking of all pixels. *arXiv preprint arXiv:2603.02573*, 2026.

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. URL https://arxiv.org/abs/2605.00078.

Haoxiang Ma, Junhao Cai, Xiaoxu Xu, Hao Li, Yuyin Yang, Yang Tian, Jiafei Cao, Hongrui Zhu, Zherui Qiu, Zhaxizhuoma, Yuqiang Yang, Jiaqi Peng, Xueyuan Wei, Yangkun Zhu, Jiahao Jiang,

Xing Gao, Hanqing Wang, Feng Yuan, Kailin Li, Xueyue Zhu, Tai Wang, Yan Ding, Jiangmiao Pang, Jia Zeng, Jingjing Zhang, Bowen Zhou, Yao Mu, Chunhua Shen, and Weinan Zhang. InternVLA-A1.5: Unifying Understanding, Latent Foresight, and Action for Compositional Generalization. *arXiv preprint arXiv:2607.04988*, 2026a. URL https://arxiv.org/abs/2607.04988.

Liheng Ma, Rui Heng Yang, Zhanguang Zhang, Mateo Clemente, Ziwen Hu, Tongtong Cao, and Yingxue Zhang. Faster-WAM: Do World Action Models Need Deep Action Modules? *arXiv preprint arXiv:2608.02365*, 2026b. URL https://arxiv.org/abs/2608.02365.

Motubrain Team. World action models in real time: An empirical study of smooth execution via asynchronous deployment, 2026. URL https://arxiv.org/abs/2608.01880.

NVIDIA. NVIDIA Isaac GR00T N1.7-3B. Hugging Face model card, 2026. URL https://huggingface.co/nvidia/GR00T-N1.7-3B. Model version N1.7. Accessed: 2026-09-18.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, 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. *arXiv preprint arXiv:2503.14734*, 2025. URL https://arxiv.org/abs/2503.14734.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient Action Tokenization for Vision-Language-Action Models. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.012. URL https://www.roboticsproceedings.org/rss21/p012.html.

Physical Intelligence. OpenPI: Open-Source Models and Packages for Robotics. GitHub repository, 2025. URL https://github.com/Physical-Intelligence/openpi.

Ryan Punamiya, Simar Kareer, Zeyi Liu, et al. EgoVerse: An egocentric human dataset for robot learning from around the world, 2026. URL https://arxiv.org/abs/2604.07607.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL https://www.jmlr.org/papers/v21/20-074.html.

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

Jaehwi Song, Suchae Jeong, Byeongguk Jeon, Sungdong Kim, Minjoon Seo, Hyungmok Son, and Kimin Lee. Habit: Human-aware behavior and interaction training dataset for robot manipulation. *arXiv preprint arXiv:2606.31682*, 2026.

StarVLA Community. StarVLA: A Lego-like Codebase for Vision-Language-Action Model Developing. *arXiv preprint arXiv:2604.05014*, 2026. URL https://arxiv.org/abs/2604.05014.

Jiayi Su, Yixin Zheng, Mi Yan, Li Yi, Zhizheng Zhang, and He Wang. GPT 6 Astra as an embodied policy. Technical report and code, 2026. URL https://github.com/anonymous-report-421/eval-of-gpt-6-astra-as-policy.

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. In *European Conference on Computer Vision*, 2026. URL https://arxiv.org/abs/2602.10098.

Shuhan Tan, Kairan Dou, Yue Zhao, and Philipp Krähenbühl. Interactive Post-Training for Vision-Language-Action Models. *arXiv preprint arXiv:2505.17016*, 2025. URL https://arxiv.org/abs/2505.17016.

Galaxea Team. Galaxea g0: Open-world dataset and dual-system vla model. *arXiv preprint arXiv:2509.00576*, 2025.

Xiaomi Robotics Team, Jun Guo, Piaopiao Jin, Jason Li, Peiyan Li, Yingyan Li, Futeng Liu, Wanli Peng, Optimus Qin, Yifei Su, et al. Xiaomi-robotics-1: Scaling vision-language-action models with over 100k hours of real-world trajectories. *arXiv preprint arXiv:2607.15330*, 2026.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

Wan Team. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. URL https://arxiv.org/abs/2503.20314.

Chenyi Wang, Xinkai Wang, Bokai Lin, Jialin Tian, Fucheng Zhang, Cewu Lu, and Lixin Yang. Track4Action: Distilling World-Centric 3D Tracker into Vision-Language-Action Policies. *arXiv preprint arXiv:2608.03727*, 2026a. URL https://arxiv.org/abs/2608.03727.

Mingxin Wang, Bin Hu, Bin Qian, Kaitao Jiang, Haoning Wu, Feng Yan, Bowen Jing, Ruiyang Hao, Enyi Wang, Kangning Niu, Yandan Yang, Mu Xu, Yan Wang, Houde Liu, and Tianlun Li. ST-WAM: Semantic-Temporal World Action Model for Robust Manipulation under Visual Distribution Shifts. *arXiv preprint arXiv:2607.28993*, 2026b. URL https://arxiv.org/abs/2607.28993.

Ye Wang, Pei Lin, Xiong-Hui Chen, Haoqi Yuan, Zhixuan Liang, Yiyang Huang, Anzhe Chen, Zixing Lei, Jie Zhang, Tao Zhang, Haoyang Li, Tong Zhang, Chenxi Xiao, Ziyuan Jiao, and Qin Jin. Ego2Robot: Scalable Robot Data Synthesis from Egocentric Human Data, 2026c. URL http://arxiv.org/abs/2608.02580. arXiv:2608.02580 [cs.RO].

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025.

Yuran Wang, Siqiao Huang, Mingleyang Li, Chenhao Zhang, Jiaqi Liang, Weiyang Jin, Yue Chen, Xuemin Chi, Donghao Zhou, Qize Yu, Yu-Kai Wang, Yuhan Rui, Shenzhe Yao, Zhen Yuan, Zhenhao Shen, Kefei Zhu, Zijie Zhu, Ning Gao, Xiaowei Chi, Guanqi He, Shanghang Zhang, Hao Dong, Lin Shao, and Hang Zhao. OpenWAM: An Open, Modular Exploration Towards Systematic World-Action Model Pretraining. *arXiv preprint arXiv:2609.07398*, 2026d. URL https://arxiv.org/abs/2609.07398.

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

Shihan Wu, Xuecheng Liu, Shaoxuan Xie, Pengwei Wang, Xinghang Li, Bowen Yang, Zhe Li, Kai Zhu, Hongyu Wu, Yiheng Liu, Zhaoye Long, Yue Wang, Chong Liu, Dihan Wang, Ziqiang Ni, Xiang Yang, You Liu, Ruoxuan Feng, Runtian Xu, Lei Zhang, Denghang Huang, Chenghao Jin, Anlan Yin, Xinlong Wang, Zhenguo Sun, Junkai Zhao, Mengfei Du, Mingyu Cao, Xiansheng Chen, Hongyang Cheng, Xiaojie Zhang, Yankai Fu, Ning Chen, Cheng Chi, Sixiang Chen, Huaihai Lyu, Xiaoshuai Hao, Yequan Wang, Bo Lei, Dong Liu, Xi Yang, Yance Jiao, Tengfei Pan, Yunyan Zhang, Songjing Wang, Ziqian Zhang, Xu Liu, Ji Zhang, Caowei Meng, Zhizheng Zhang, Jiyang Gao, Song Wang, Xiaokun Leng, Zhiqiang Xie, Zhenzhen Zhou, Peng Huang, Wu Yang, Yandong Guo, Yichao Zhu, Suibing Zheng, Hao Cheng, Xinmin Ding, Yang Yue, Huanqian Wang, Chi Chen, Jingrui Pang, YuXi Qian, Haoran Geng, Lianli Gao, Haiyuan Li, Bin Fang, Gao Huang, Yaodong Yang, Hao Dong, He Wang, Hang Zhao, Yadong Mu, Di Hu, Hao

Zhao, Tiejun Huang, Shanghang Zhang, Yonghua Lin, Zhongyuan Wang, and Guocai Yao. Robocoin: An open-sourced bimanual robotic data collection for integrated manipulation, 2025b. URL https://github.com/FlagOpen/RoboCOIN.

Wei Wu, Fan Lu, Yunnan Wang, Shuai Yang, Shi Liu, Fangjing Wang, Qian Zhu, He Sun, Yong Wang, Shuailei Ma, Yiyu Ren, Kejia Zhang, Hui Yu, Jingmei Zhao, Shuai Zhou, Zhenqi Qiu, Houlong Xiong, Ziyu Wang, Zechen Wang, Ran Cheng, Yong-Lu Li, Yongtao Huang, Xing Zhu, Yujun Shen, and Kecheng Zheng. A Pragmatic VLA Foundation Model. *arXiv preprint arXiv:2601.18692*, 2026a. URL https://arxiv.org/abs/2601.18692.

Wei Wu, Fangjing Wang, Fan Lu, He Sun, Shi Liu, Yunnan Wang, Yibin Yan, Yong Wang, Shuailei Ma, Xinyang Wang, Yibin Liu, Shuai Yang, Tianxiang Zhou, Kejia Zhang, Lei Zhou, Cheng Su, Nan Xue, Bin Tan, Han Zhang, Youchao Zhang, Fei Liao, Xing Zhu, Yujun Shen, and Kecheng Zheng. From Foundation to Application: Improving VLA Models in Practice. *arXiv preprint arXiv:2607.06403*, 2026b. URL https://arxiv.org/abs/2607.06403.

Lishan Yang, Wenxuan Song, Xi Wang, Pingyue Sheng, Zheng Fang, Ziyang Zhou, Junjie He, Haodong Yan, Jiayi Chen, Nan Sun, Qiao Sun, Pengwei Wang, Lingqiao Liu, Yan Wang, Yuxiang Gao, Feras Dayoub, and Haoang Li. 4D-WAM: Infusing Spatiotemporal Awareness into World Action Models through Trajectory Fields. *arXiv preprint arXiv:2608.08023*, 2026a. URL https://arxiv.org/abs/2608.08023.

Senqiao Yang, Chengyao Wang, Yuxin Chen, Zixuan Wang, Longxiang Tang, Haokun Gui, Jinhui Ye, Changsheng Lu, Xiaoyang Wu, Mingkang Zhu, Pengguang Chen, Shu Liu, Zhuotao Tian, Hengshuang Zhao, Bei Yu, and Jiaya Jia. Beyond Data Scaling: Representation-Centric Continued Pre-training for Vision-Language-Action Models. *arXiv preprint arXiv:2608.27550*, 2026b. URL https://arxiv.org/abs/2608.27550.

Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Dekang Qi, Junjin Xiao, Haoyun Liu, Ronghan Chen, Yuzhi Chen, Dongjie Huo, Feng Xiong, Xing Wei, Zhiheng Ma, and Mu Xu. ABot-M0: VLA Foundation Model for Robotic Manipulation with Action Manifold Learning. *arXiv preprint arXiv:2602.11236*, 2026c. URL https://arxiv.org/abs/2602.11236.

Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, Jie Li, Jindi Lv, Jingyu Liu, Min Cao, Peng Li, Qiuping Deng, Wenjun Mei, Xiaofeng Wang, Xinze Chen, Xinyu Zhou, Yang Wang, Yifan Chang, Yifan Li, Yukun Zhou, Yun Ye, Zhichao Liu, and Zheng Zhu. GigaWorld-Policy: An Efficient Action-Centered World–Action Model. *arXiv preprint arXiv:2603.17240*, 2026a. URL https://arxiv.org/abs/2603.17240.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi "Jim" Fan, and Joel Jang. World Action Models are Zero-shot Policies. *arXiv preprint arXiv:2602.15922*, 2026b. URL https://arxiv.org/abs/2602.15922.

Haoqi Yuan, Zhixuan Liang, Anzhe Chen, Ye Wang, Haoyang Li, Pei Lin, Yiyang Huang, Zixing Lei, Tong Zhang, Jiazhao Zhang, Jie Zhang, Jingyang Fan, Gengze Zhou, Qihang Peng, Chenxu Lv, Xiaoyue Chen, An Yang, Fei Huang, Junyang Lin, Dayiheng Liu, Jingren Zhou, Chenfei Wu, and Xiong-Hui Chen. Qwen-RobotManip Technical Report: Alignment Unlocks Scale for Robotic Manipulation Foundation Models. *arXiv preprint arXiv:2606.17846*, 2026a. URL https://arxiv.org/abs/2606.17846.

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

He Zhang, Lingzhu Xiang, Haitao Lin, Zeyu Huang, Minghui Wang, Dingyan Zhong, Yubo Dong, Yihao Wu, Yongming Rao, Dongsheng Zhang, et al. Hy-Embodied-0.5-VLA: From vision-language-action models to a real-world robot learning stack. *arXiv preprint arXiv:2606.14409*, 2026a.

Qihang Zhang, Lin Li, Luyao Zhang, Shuai Yang, Yiming Luo, Shuaiting Li, Ruilin Wang, Junke Wang, Jiahao Shao, Gangwei Xu, Jiaming Zhou, Yishu Shen, Yudong Jin, Fangyi Xu, Shuailei Ma, Jiaqi Liao, Guanxing Lu, Zifan Shi, Yongkun Wen, Yujie Zhao, Weixuan Tang, Xinyang Wang, Chaojian Li, Jiapeng Zhu, Ka Leong Cheng, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Native Video-Action Pretraining for Generalizable Robot Control. *arXiv preprint arXiv:2607.08639*, 2026b. URL https://arxiv.org/abs/2607.08639.

Wenbo Zhang, Kaixuan Wang, Yutao Ouyang, Xiaoyu Huang, Liyang Li, Kailun Su, Weiyang Jin, Wenhao Chai, Haotian Liang, Zhiyang Dou, Yue Chen, and Tianxing Chen. An unexpected robot policy: Early evaluations of gpt-6 astra on robodojo and beyond, 2026c. URL https://arxiv.org/abs/2609.24170.

Yuyang Zhang, Wenyao Zhang, Zekun Qi, He Zhang, Haitao Lin, Jingbo Zhang, Yao Mu, Xiaokang Yang, Wenjun Zeng, and Xin Jin. ImageWAM: Do World Action Models Really Need Video Generation, or Just Image Editing? *arXiv preprint arXiv:2606.19531*, 2026d. URL https://arxiv.org/abs/2606.19531.

Zongzheng Zhang, Jingrui Pang, Zhuo Yang, Kun Li, Minwen Liao, Saining Zhang, Guoxuan Chi, Jinbang Guo, Huan ang Gao, Modi Shi, Dongyun Ge, Yao Mu, Jiayuan Gu, Rui Chen, Hao Dong, Huazhe Xu, Li Yi, Yixin Zhu, Hang Zhao, Pengwei Wang, Shanghang Zhang, Guocai Yao, Jianyu Chen, Hongyang Li, and Hao Zhao. Dexora: Open-source vla for high-dof bimanual dexterity, 2026e. URL https://arxiv.org/abs/2605.18722.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Ya-Qin Zhang, Jiangmiao Pang, Jingjing Liu, Tai Wang, and Xianyuan Zhan. X-VLA: Soft-Prompted Transformer as Scalable Cross-Embodiment Vision-Language-Action Model. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.10274.

Shangchen Zhou, Chongyi Li, Kelvin C.K. Chan, and Chen Change. ProPainter: Improving Propagation and Transformer for Video Inpainting. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10443–10452, Paris, France, October 2023. IEEE. doi: 10.1109/ICCV51070.2023.00961. URL https://ieeexplore.ieee.org/document/10378438/.

A APPENDIX

The supplementary material preserves the detailed model, data-processing, training, infrastructure, and experimental descriptions. The core method and principal evidence are presented in the main text; the following sections provide implementation details and complete result tables.

Section	Contents
A.1	Extended related work
A.3	Canonical representation, datasets, and filtering
A.8	Pretraining and post-training configurations
A.9	Training systems and real-robot deployment
A.10	Complete experimental results and analyses
A.13	Human-to-robot inverse-kinematics details

A.1 EXTENDED RELATED WORK

A.1.1 VISUAL REPRESENTATIONS FOR ROBOT CONTROL

Beyond the policy architecture, the choice of representation determines which aspects of visual dynamics are made available to action generation. Video Prediction Policy (Hu et al., 2025) extracts features from a video diffusion model, while V-JEPA and V-JEPA 2 learn predictive embeddings and extend them to action-conditioned latent planning (Bardes et al., 2024; Assran et al., 2025). VLA-JEPA (Sun et al., 2026) and JEPA-WAM (Lin et al., 2026) incorporate future-state and spatial transition prediction into policy learning. Other approaches distill video foresight into policy queries (Ma et al., 2026a) or combine generative and semantic prediction targets with history retrieval (Wang et al., 2026b).

Geometric and motion supervision further enrich policy representations. Spatial Forcing (Li et al., 2026a) aligns VLA features with pretrained 3D representations, while LingBot-VLA 2.0 (Wu et al., 2026b) supervises queries with depth and causal video features. 4D-WAM (Yang et al., 2026a) transfers trajectory-field knowledge through temporal differences and correspondence. Track4Action (Wang et al., 2026a) predicts pooled Track4World (Lu et al., 2026) descriptors to condition action generation.

InternW0-Δ combines predictive and geometric supervision within the video expert, with distinct roles for the two representations. Causal Imprint predicts clean video-latent changes and aligns with future video-expert features; its hidden states directly inform the action expert. Separately, we use Track4World (Lu et al., 2026) as a training-only teacher to distill clip-level geometry and motion information into the video expert.

A.1.2 OPEN-SOURCE SYSTEMS FOR ROBOT LEARNING.

Open robot learning depends on reusable data interfaces, training implementations, and evaluation and deployment tools in addition to model checkpoints. LeRobot (Cadene et al., 2026) provides shared infrastructure for dataset handling, policy training, and robot interaction. OpenVLA (Kim et al., 2024) and openpi (Physical Intelligence, 2025) make pretrained policies accessible through public implementations and adaptation workflows, while StarVLA (StarVLA Community, 2026) modularizes VLA development to support interchangeable components and controlled experimentation. These systems lower the cost of reproducing and extending robot policies, although their supported models, data pipelines, and deployment settings differ.

Recent foundation-model efforts also expose larger-scale training recipes. LingBot-VLA (Wu et al., 2026a) releases checkpoints and a training and evaluation codebase. Qwen-RobotManip (Yuan et al., 2026a) documents the data-alignment pipeline underlying its manipulation policy. OpenWAM (Wang et al., 2026d) brings modular architectures, data processing, training, and deployment into a common framework for systematic WAM research. We share this emphasis on accessible research infrastructure. Alongside the policy, InternW0-Δ documents a unified data representation, multi-stage training, reusable frozen-encoder and teacher-feature caches, and compilation and memory optimizations. We will release these components together with the data-processing and

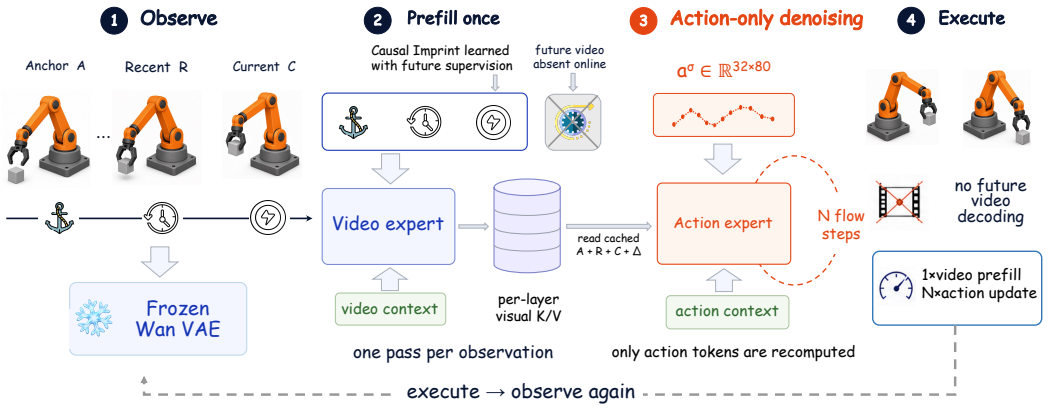


Figure 4: **Efficient inference with cached visual context.** (1) Anchor (A), recent (R), and current (C) observations are encoded by the frozen Wan VAE. (2) Conditioned on the video context, the video expert processes the observation tokens and Causal Imprint tokens once, caching their key/value (K/V) projections at each MoT layer. Future-video tokens are omitted. (3) Starting from Gaussian noise, the action expert generates an action chunk through N flow steps, reusing the cached A/R/C/Δ features and the action-side context. Only the action stream is recomputed during denoising. (4) The predicted actions are executed, and the process repeats with an updated cache when a new observation is acquired. Each observation-to-action cycle requires one video-expert prefill and N action-expert updates, without future-video sampling or decoding.

filtering code, versioned filtered-data indices, training and evaluation code, model checkpoints, and deployment utilities, so that subsequent work can investigate model and representation choices with a concrete, inspectable training recipe.

A.2 Efficient Inference

As illustrated in Figure 4, InternW0-Δ generates action chunks without sampling future video. Inference consists of a single video-expert prefill followed by iterative action-only denoising. The visual context is computed once for each new observation and reused throughout the action denoising process.

Visual prefill and caching. At each policy step, we encode the anchor, recent, and current observations defined in Equation (9) using the frozen Wan VAE. The resulting observation tokens, together with the learned Causal Imprint tokens, are processed by the video expert conditioned on C_t^v, without instantiating future-video tokens. We cache the self-attention key/value (K/V) projections of the A/R/C/Δ tokens at each MoT layer:

$$\mathcal{C}_t^{\mathrm{KV}} = \left\{ \left(K_t^{v,l}, V_t^{v,l} \right) \right\}_{l=1}^{L}, \tag{19}$$

where L is the number of coupled MoT layers. The directed mask in Figure 3(c) prevents observation and Causal Imprint tokens from attending to action tokens. With the video-side inputs and conditioning held fixed during action denoising, these cached features are independent of the evolving action tokens and can be reused at every flow step.

Action-only denoising and execution. We initialize a noisy action chunk $A_t^1 \sim \mathcal{N}(0, I)$ with shape $H_a \times D_a$, where $H_a = 32$ and $D_a = 80$. At each flow step, the action expert recomputes only the action stream, attending to the cached visual K/V and the action tokens, while using C_t^a through its cross-attention pathway. The predicted action flow velocity is

$$v_t^\sigma = f_\theta^a \left(A_t^\sigma, \sigma; \mathcal{C}_t^{\mathrm{KV}}, C_t^a \right), \tag{20}$$

where f_θ^a denotes the action-side flow predictor. Following the flow convention in Equation (13), we perform N integration steps from $\sigma = 1$ to $\sigma = 0$ to obtain the predicted action chunk $\hat{A}_t = A_t^0$.

No video-side token features are recomputed within this loop, and neither future-video sampling nor VAE decoding is required.

The predicted action chunk is used for robot execution. When a new observation is acquired, the sparse memory and conditioning are updated, and the visual cache is recomputed for the next cycle. Each cycle therefore consists of one video-expert prefill and N action-expert updates.

A.3 Data Representation and Processing

In this section, we introduce the data recipe of InternW0-Δ, which unifies heterogeneous embodied data under a common representation for joint training. We first describe the unified representation used across different data sources. We then describe the data sources, processing rules, and filtering statistics for robot manipulation data, egocentric and ego-to-robot data, and Universal Manipulation Interface (UMI) data.

A.3.1 Unified Representation

We train InternW0-Δ to use heterogeneous data sources that span diverse robot morphologies, degrees of freedom, control interfaces, coordinate conventions, and action dimensions. Directly mixing their native representations would assign inconsistent physical meanings to the same dimensions, hindering cross-embodiment learning. We therefore convert all trajectories into a canonical state–action representation with fixed semantic slots. In particular, all robot actions are mapped into a shared 80-dimensional action space, whose detailed layout is provided in Table 6. Each slot is associated with a predefined physical quantity rather than the original ordering used by the source dataset.

For the robot state, we retain absolute quantities that describe the current robot configuration. These include the current joint positions, the absolute end-effector (EEF) pose, and the current gripper or hand configuration. The EEF pose is represented by a 3D position together with a continuous 6D rotation representation. For parallel-jaw grippers, the hand state is represented by the gripper opening, while dexterous hands use the corresponding hand joint configuration. All EEF poses are first transformed into a consistent coordinate convention before being assigned to the canonical slots.

For the action representation, we distinguish between joint-space and task-space control signals. Joint actions are represented as absolute target joint positions, corresponding to the configuration that the low-level controller is expected to reach at the next or a future control step. EEF actions consist of a 3D translation increment and a 3D rotation vector. The latter parameterizes the relative orientation change in the local coordinate frame of the current EEF. Thus, each EEF uses a 9D absolute pose representation in the state and a 6D relative motion representation in the action, before being embedded into the canonical format. An action segment contains H_a consecutive canonical action vectors, each following the same fixed semantic layout.

A.4 Robot Data

A.4.1 Data Sources

Robot manipulation demonstrations serve as the main source of embodied supervision in our training corpus. We integrate a total of 15 datasets collected from both simulated and real-world environments, covering a wide range of manipulation scenarios, including single-arm and dual-arm tabletop tasks, dexterous manipulation, mobile manipulation, and humanoid interaction. The resulting collection spans diverse robot embodiments, task settings, sensing configurations, and control interfaces, providing broad coverage of manipulation behaviors for model training.

AgiBotWorld (contributors, 2024) AgiBotWorld is a large-scale real-world manipulation dataset collected using a fleet of dual-arm humanoid robots. Its full release contains over one million trajectories and approximately 3,000 hours of demonstrations, covering 217 tasks, 87 manipulation skills, more than 3,000 objects, and over 100 real-world scenes. The data includes dual-arm manipulation, dexterous-hand interaction, tool use, and mobile manipulation, with multimodal observations from multiple cameras and tactile sensors.

Table 6: Canonical layout of the 80-dimensional robot action representation. Slot ranges use zero-based, half-open indexing. Paired arm ranges are listed in left–right order.

Component	Slot range(s)	Dim.
Arm-specific slots		
Arm joints	$[0, 7),\ [40, 47)$	2×7
End-effector	$[7, 16),\ [47, 56)$	2×9
Gripper	$[16, 17),\ [56, 57)$	2×1
Hand	$[17, 29),\ [57, 69)$	2×12
Body and mobility slots		
Torso joints	$[29, 34)$	5
Independent lift	$[39, 40)$	1
Head	$[74, 77)$	3
Mobile base	$[77, 80)$	3
Reserved slots	$[34, 39),\ [69, 74)$	2×5
Total	$[0, 80)$	**80**

InternData-A1 (contributors, 2025) InternData-A1 provides large-scale manipulation data across both simulated and real-world settings. It contains more than 630K trajectories and 7,400 hours of interaction across four robot embodiments, 70 tasks, and 227 scenes, covering rigid, articulated, deformable, and fluid-object manipulation. It further includes long-horizon tasks, multi-arm collaboration, and human–robot interaction, providing complementary supervision beyond standard tabletop manipulation.

RoboMIND (Wu et al., 2025a) and RoboMIND 2.0 (Hou et al., 2025) RoboMIND contains 107K human-teleoperated trajectories across 479 tasks and four robot embodiments, ranging from single-arm manipulators to dual-arm and dexterous humanoid platforms. It also provides failure demonstrations in addition to successful trajectories. RoboMIND 2.0 further extends this collection to more than 310K real-world dual-arm trajectories across six embodiments and 739 tasks, together with tactile-enhanced and mobile-manipulation episodes, substantially increasing the coverage of contact-rich and spatially extended behaviors.

RW-RL Dataset (Intelligence et al., 2026) The RW-RL Dataset focuses on real-world interaction data for policy improvement beyond offline imitation learning. It contains over 1,000 hours of interaction across multiple robot families, scenario domains, and task templates, combining teleoperated demonstrations with autonomous rollouts, human interventions, reward signals, and termination labels. These data expose the model to both successful behaviors and states encountered during policy execution and recovery.

Dexora (Zhang et al., 2026e) Dexora targets high-DoF bimanual dexterous manipulation with dual robot arms and dual dexterous hands. It combines 100K embodiment-matched simulated trajectories with 10K real-world teleoperated episodes, covering both coarse arm motion and fine-grained finger control. This data provides dense supervision for manipulation behaviors that require coordinated arm and finger motion beyond parallel-jaw grippers.

ABC-130K (Allshire et al., 2026) ABC-130K contains more than 130K real-world bimanual teleoperation episodes, corresponding to over 3,500 hours of interaction across 195 manipulation tasks. Collected with dual-arm YAM platforms, it covers manipulation primitives including pick-and-place, folding, handover, insertion, tool use, and assembly. Its scale and task composition provide extensive supervision for coordinated bimanual manipulation.

Galaxea Open-World Dataset (Team, 2025) The Galaxea Open-World Dataset contains more than 500 hours of real-world mobile manipulation collected with a consistent robot embodiment. The demonstrations span residential, kitchen, retail, and office environments and include fine-grained subtask-level language annotations. Compared with fixed tabletop data, it provides ad-

ditional supervision for spatially extended and long-horizon manipulation that couples navigation with object interaction.

RoboCOIN (Wu et al., 2025b) RoboCOIN is a multi-embodiment bimanual manipulation dataset containing more than 180K demonstrations collected across 15 robot platforms. It covers 421 tasks in 16 real-world scenarios and organizes bimanual behaviors according to coordination patterns and object properties. Its hierarchical annotations provide supervision at trajectory, subtask, and frame levels, while its embodiment diversity complements datasets collected using a single robot platform.

RH20T (Fang et al., 2024) RH20T contains more than 110K contact-rich real-world manipulation sequences covering 147 tasks and seven robot configurations. In addition to RGB observations and robot proprioception, it records force, audio, depth, and, for part of the data, tactile information. Each robot trajectory is also associated with a human demonstration and language description, providing rich multimodal supervision for contact-sensitive manipulation.

RDT-1B (Liu et al., 2024) The robot-data corpus used by RDT-1B aggregates 46 datasets into more than one million episodes spanning multiple robot embodiments and manipulation settings. It is further complemented by more than 6K demonstrations collected on the ALOHA dual-arm platform. We incorporate this data to increase the diversity of robot morphologies and to strengthen the coverage of coordinated bimanual behaviors.

RoboSet (Kumar et al., 2023) RoboSet combines real-world teleoperation data with simulated human and expert-policy trajectories under a common data format. Its main real-world component contains approximately 31K kitchen manipulation trajectories across 40 tasks, accompanied by multi-view visual observations, actions, robot states, and rewards. The dataset therefore contributes both large-scale real-world tabletop interactions and additional simulated manipulation behaviors.

RealSource-World (RealSource, 2025) RealSource-World contains more than 14 million frames from over 11K real-world dual-arm manipulation episodes collected using the RS-02 humanoid robot. The dataset covers 35 tasks across household, kitchen, office, retail, and industrial environments, with synchronized head and wrist cameras and rich robot proprioception. It additionally provides atomic-skill segmentation and trajectory-quality annotations, making long-horizon demonstrations accessible at finer temporal granularity.

MolmoAct2-BimanualYAM (Fang et al., 2026) MolmoAct2-BimanualYAM provides more than 720 hours of real-world bimanual manipulation demonstrations collected using dual YAM arms. The dataset contains multi-view observations and language-annotated tasks covering behaviors such as garment manipulation, cable handling, grocery organization, packing, and table bussing. These demonstrations add substantial coverage of long-horizon bimanual tasks involving coordinated interactions between both arms.

HABIT (Song et al., 2026) HABIT focuses on robot manipulation in environments where humans are actively present. It contains more than 10K episodes and 160 hours of demonstrations across 60 tasks, organized around collaborative, shared-space, and human-directed interactions. This setting introduces behaviors such as human–robot synchronization, yielding, and gesture-conditioned manipulation that are largely absent from robot-only demonstration datasets.

ActionNet (Fourier ActionNet Team & Mu, 2025) ActionNet focuses on dexterous bimanual manipulation using humanoid robots. It contains more than 30K teleoperated trajectories, corresponding to approximately 140 hours of interaction, and covers both basic manipulation primitives and more complex two-hand behaviors using dexterous hands. The demonstrations are collected across multiple Fourier humanoid platforms and paired with manually verified language instructions, providing additional supervision for high-DoF humanoid manipulation.

A.4.2 Processing Rules

Our data curation builds on practices used in prior work, including Qwen-RobotManip (Yuan et al., 2026a). We apply signal anomaly and consistency filtering, static boundary trimming, visual quality

filtering, and an action magnitude guard to the robot demonstrations. We additionally perform automated checks of instruction correctness and video–instruction consistency. The resulting data are further checked through manual inspection of sampled episodes to verify robot motion and representation conventions across datasets and embodiments. We release the complete filtering pipeline and its implementation as an open-source resource to support reproducible data curation and facilitate community reuse and adaptation to additional datasets and embodiments.

Signal anomaly and consistency filtering. Following Qwen-RobotManip (Yuan et al., 2026a), we perform sudden-change detection and state–action consistency checks. Frames containing non-finite values or abrupt outliers, identified through deviations from smoothed trajectories and higher-order temporal changes, are removed while contiguous valid segments are retained. For physically comparable state–action dimensions with sufficient variation, we assess directional agreement after local cross-correlation alignment. We use a default agreement threshold of 0.65 and reject episodes if any required dimension falls below this threshold or if state changes systematically precede the associated actions. Comparisons involving delta actions are adapted to the corresponding control representation.

Static boundary trimming. We apply an adaptive trimming rule to remove prolonged inactivity at episode boundaries. Motion is estimated from normalized frame-to-frame state differences, with rotations represented continuously over time to avoid artificial discontinuities. An adaptive threshold derived from each episode's motion distribution identifies sustained active segments. We retain the interval spanning the first and last reliable active segments, together with a small context margin at both ends. Intermediate pauses are preserved to maintain the temporal structure of task execution. Short episodes and those without a clear separation between static and active motion are left unchanged.

Visual quality filtering. We perform frame-level visual quality checks to detect black frames, blur, compression artifacts, and other image-level anomalies, using frames resized to a unified resolution for all subsequent measurements. Black frames are identified from the mean luminance and the fraction of pixels below a dark-intensity threshold. For blur detection, Sobel gradients are first used to locate edge pixels, and the mean absolute Laplacian response is then computed only over sufficiently populated edge regions to avoid false positives in textureless manipulation scenes. Compression artifacts are quantified by comparing neighboring-pixel differences at periodic 8×8 block boundaries with those at non-boundary locations. Abrupt temporal changes are measured using frame differences normalized by the episode-level median and median absolute deviation. Extreme temporal changes, blockiness, and saturation anomalies are combined to identify suspected corruption, while decoding and shape errors are recorded separately.

Action magnitude guard. After the preceding signal and visual quality checks, we apply an action magnitude guard to remove residual large-amplitude commands. In some datasets, initialization and reset routines near episode boundaries introduce sustained large actions, such as rapid arm retraction. These motions can remain temporally smooth and consistent with the recorded states, allowing them to survive the preceding filters. When present over consecutive frames, they can also distort action statistics despite normalization based on the 1st and 99th percentiles (q_{01} and q_{99}). We therefore impose additional magnitude limits in physical units. For end-effector (EEF) actions, we discard samples whose single-step translation magnitude exceeds $0.2\,\mathrm{m}$ or whose single-step rotation magnitude exceeds $0.5\,\mathrm{rad}$.

Instruction correctness and video-instruction consistency filtering. We apply an LLM-based filter to assess the correctness, clarity, and executability of each natural-language instruction. Instructions that are empty, malformed, gibberish, incomplete, or lack a clear action, object, or target are removed before visual verification. For the remaining instructions, we perform task-level video-instruction consistency checking with a vision-language model. Using gripper-state signals recorded in the robot data, we detect opening and closing events and split each episode into action-centered video clips. The VLM first interprets these local clips individually to obtain descriptions of the corresponding manipulation actions. We then combine the clip-level descriptions with sparsely sampled global frames from the full episode and the original instruction, and ask the VLM to determine whether the observed behavior matches the instruction and whether the task is completed.

Table 7: Dataset-level statistics before and after filtering. FPS denotes the frame rate declared in the training data metadata; comma-separated values indicate subsets with different frame rates.

Dataset	FPS	Before filtering		After filtering		
		Hours	Episodes	Hours	Episodes	Frames (M)
Robot data						
InternData-A1 (contributors, 2025)	30	3,528.27	573,389	3,375.51	559,062	364.555
AgiBotWorld (contributors, 2024)	30	2,358.63	141,563	1,698.21	105,227	183.407
Galaxea (Team, 2025)	15, 62	681.22	29,994	327.42	17,703	17.684
RoboMIND (Wu et al., 2025a)	30	777.01	215,240	625.79	195,869	67.586
ABC-130K (Allshire et al., 2026)	30	3,533.30	128,996	2,587.94	94,727	279.497
RoboSet (Kumar et al., 2023)	≈ 5.12	21.64	9,500	20.05	9,382	0.370
MolmoAct2 (Fang et al., 2026)	30	68.68	3,331	33.28	1,927	3.594
RoboCOIN (Wu et al., 2025b)	30, 50	533.17	77,843	485.77	77,102	53.095
RDT-1B (Liu et al., 2024)	25	35.66	6,110	33.10	6,110	2.979
RH20T (Fang et al., 2024)	10	1,132.10	82,894	1,130.92	82,823	40.713
ActionNet (Fourier ActionNet Team & Mu, 2025)	30	157.72	32,121	157.22	32,120	16.980
Dexora (Zhang et al., 2026e)	20	26.30	7,867	26.30	7,867	1.894
HABIT (Song et al., 2026)	10	165.18	10,623	122.71	8,389	4.417
RealSource (RealSource, 2025)	30	345.46	26,671	177.92	25,504	19.216
RWRL (Intelligence et al., 2026)	15, 30	503.37	23,861	500.06	23,844	45.060
Total	–	**13,867.71**	**1,370,003**	**11,302.20**	**1,247,656**	**1,101.047**
UMI data						
Hy-UMI-10K (Zhang et al., 2026a)	30	2,161.74	250,135	2,075.42	416,053	224.146
Egocentric human data						
EgoDex (Hoque et al., 2025)	30	821.48	333,682	772.16	310,418	83.393
EgoVerse (Punamiya et al., 2026)	30	3,818.56	1,484,714	3,289.19	1,312,338	355.233
Total	–	**4,640.04**	**1,818,396**	**4,061.35**	**1,622,756**	**438.626**

Manual semantic verification. After all automated filtering stages, we manually inspect sampled episodes from every dataset–embodiment pair to verify the interpretation of robot states and actions. We replay the joint trajectories decoded by our data loaders using the corresponding URDF and visualize the EEF poses alongside the recorded videos from all available camera views. We assess whether the visualized robot motion, end-effector poses, and gripper behavior agree with the observations. Particular attention is paid to dataset-specific conventions, including gripper opening and closing definitions and coordinate-frame conventions. This inspection helps identify semantic mismatches and verify that our data loaders correctly interpret heterogeneous annotations and map them consistently into the unified representation.

We apply the filtering and verification stages in a fixed order. First, signal anomaly and consistency filtering removes invalid or inconsistent data while retaining contiguous valid segments. We then trim static episode boundaries, perform visual quality filtering, and apply the action magnitude guard. The remaining data undergo LLM-based instruction screening, followed by VLM-based checks of video–instruction consistency and task completion. Finally, we manually inspect sampled episodes from each dataset–embodiment pair to verify the decoded robot motion, its agreement with the recorded observations, and the mapping of dataset-specific conventions into the unified representation.

A.5 EGO AND EGO2ROBOT DATA

A.5.1 DATA SOURCES

EgoDex (Hoque et al., 2025) EgoDex contains 829 hours of egocentric human demonstrations collected using Apple Vision Pro across 194 tabletop manipulation tasks. It pairs 30 Hz RGB video with 3D head, upper-body, hand poses, camera intrinsics and poses, and language descriptions, providing fine-grained supervision for dexterous manipulation.

EgoVerse (Punamiya et al., 2026) EgoVerse is a continuously expanding collection of egocentric human demonstrations contributed by academic and industrial partners. We use an expanded EgoVerse snapshot of approximately 473K recordings, yielding 1.48M segmented episodes spanning

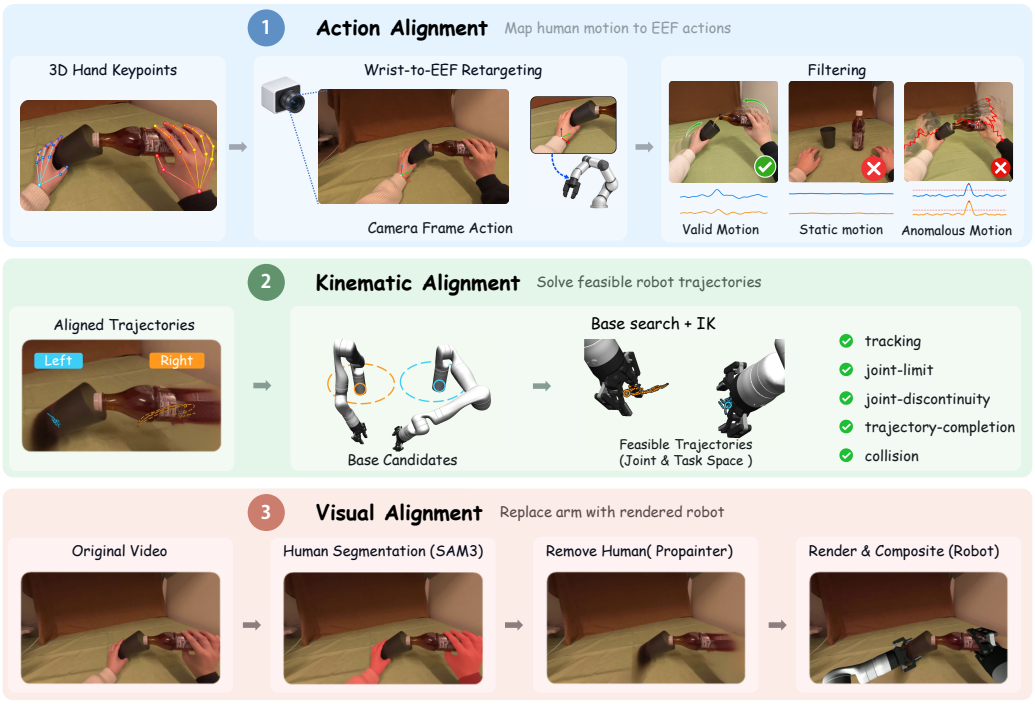


Figure 5: Ego and ego-to-robot data processing pipeline. Ego2Robot data undergo all stages, whereas egocentric data undergo only action alignment and action speed alignment. Action alignment maps hand motion to end-effector targets and gripper commands. Kinematic alignment couples base search with trajectory IK to obtain base poses and joint trajectories. Visual alignment combines the rendered robot with the human-removed scene using depth-aware compositing.

3,819 hours. The data pair egocentric video with 3D hand keypoints, 6-DoF head poses, and task descriptions, including subtask-level language annotations for industry-contributed recordings.

A.5.2 Processing Rules

The processing pipeline is organized into action alignment, kinematic alignment, and visual alignment, as shown in Figure 5. Our conversion procedure is inspired by Ego2Robot (Wang et al., 2026c) and Qwen-RobotManip (Yuan et al., 2026a). Unlike their representative-keyframe kinematic checks, we incorporate coarse-to-fine trajectory validation into base selection. Ego2Robot data undergo all stages, whereas egocentric data undergo only action alignment and action speed alignment.

Action alignment. Action alignment converts heterogeneous egocentric human data into a standardized representation of bimanual wrist states and actions. Specifically, we treat the wrist pose of each 3D hand as the corresponding end-effector pose. To obtain a unified camera-relative representation, we use the camera extrinsics to transform each end-effector pose from the source frame into the instantaneous camera frame and construct the corresponding state:

$$\mathbf{p}_{E_t}^{C_t} = R_s^{C_t}\mathbf{p}_{E_t}^{s} + \mathbf{t}_s^{C_t}, \quad R_{E_t}^{C_t} = R_s^{C_t}R_{E_t}^{s}, \quad \mathbf{s}_t = \left(\mathbf{p}_{E_t}^{C_t},\, R_{E_t}^{C_t},\, g_t\right). \tag{21}$$

Here, $\mathbf{p}_{E_t}^{s}$ and $R_{E_t}^{s}$ denote the end-effector position and orientation in the source frame s, respectively. $R_s^{C_t}$ and $\mathbf{t}_s^{C_t}$ specify the transformation from the source frame to the camera frame C_t, and g_t denotes gripper openness. Notably, We estimate gripper openness from human fingertip geometry. The canonical mapping uses the distance between the thumb tip and the midpoint of the index and middle fingertips:

$$d_t = \left\| \mathbf{p}_{\text{thumb},t} - \frac{\mathbf{p}_{\text{index},t} + \mathbf{p}_{\text{middle},t}}{2} \right\|_2, \qquad g_t = \text{clip}\left(\frac{d_t - 0.05}{0.02},\, 0,\, 1 \right). \tag{22}$$

All fingertip positions are measured in meters and expressed in a common coordinate frame at the same timestamp. Distances of $5\,\mathrm{cm}$ and $7\,\mathrm{cm}$ map to fully closed ($g_t = 0$) and fully open ($g_t = 1$) gripper targets, respectively. Following the action parameterization used for robot data, we define actions using camera-relative position deltas, EEF-local rotation deltas, and absolute next-frame gripper targets:

$$\Delta \mathbf{p}_t = \mathbf{p}_{E_{t+1}}^{C_{t+1}} - \mathbf{p}_{E_t}^{C_t}, \quad \Delta R_t = \left(R_{E_t}^{C_t}\right)^\top R_{E_{t+1}}^{C_{t+1}}, \quad \mathbf{a}_t = \left(\Delta \mathbf{p}_t,\, \Delta R_t,\, g_{t+1}\right). \tag{23}$$

After action alignment, we retain an episode only if at least one hand has valid pose observations spanning more than $1\,\mathrm{s}$ and covering more than 70% of the episode duration. We then discard episodes in which both hands are static, defined by each hand having position variance below $5 \times 10^{-4}\,\mathrm{m}^2$ and rotation variance below $0.1\,\mathrm{rad}^2$. For hand trajectories, we apply the trajectory outlier detection described in the robot data section and additionally reject episodes with excessive linear or angular speeds. We apply the same anomaly checks to the available camera trajectories. To limit the effects of substantial locomotion during manipulation, we also remove episodes with camera displacement greater than $1\,\mathrm{m}$.

Kinematic alignment. Kinematic alignment couples base selection with inverse kinematics (IK): the outer search proposes base configurations, while the inner solver holds each base fixed and computes joint trajectories to assess its suitability. Unlike the representative-keyframe checks described in Ego2Robot and Qwen-RobotManip, we incorporate coarse-to-fine trajectory validation into the search to detect tracking failures and discontinuous changes between IK solution branches.

Algorithm 1 summarizes the coupled base search and trajectory IK procedure for bimanual arms. The shared robot model $\mathcal{M}$ is loaded in MuJoCo (Todorov et al., 2012) from a robot description file and includes the kinematic structure, joint limits, collision geometry, and visual geometry required for IK, collision checking, and rendered occupancy evaluation. For arm a, $\mathcal{T}_a$ contains the end-effector targets and gripper commands, $\mathcal{W}_a$ is its workspace index, and $\mathcal{B}_a$ contains base candidates with their available joint trajectories and evaluation statistics. The stage index s takes the values coarse and medium in sequence, with progressively denser trajectory evaluation frames.

$\mathrm{SearchArmTrajectories}$ uses workspace seeds and prior trajectories for base-fixed IK, retaining a ranked shortlist subject to stage-specific tracking, joint-limit, joint-discontinuity, and trajectory-completion criteria and search limits. $\mathrm{PairBases}$ forms candidate pairs, rejects those violating minimum base separation or exhibiting medium-stage cross-arm collisions, and orders the survivors. Both functions use stage-specific priorities based on motion quality, rendered image occupancy, and layout preferences without relaxing acceptance criteria. Visual preferences favor less base and proximal-arm intrusion while also considering overall robot coverage. $\mathrm{SolveFullTrajectory}$ then solves full-frame trajectories with cached arm/base solutions reused when available and returns a pass flag v after trajectory-quality and cross-arm collision checks. Both IK routines use hierarchical updates combining position and approach tracking, contact-point and orientation refinement, and projected joint-space regularization, as detailed in Appendix A.13. The search returns the first passing pair in the scheduled order.

Visual alignment. Visual alignment uses SAM3 (Carion et al., 2025) to segment human regions and ProPainter (Zhou et al., 2023) to remove them from the source video. Using the selected base configurations and joint trajectories, the target robot is rendered from the original camera viewpoint and composited into the inpainted scene.

Depth-aware compositing calibrates estimated scene depth using metric hand keypoints and checks scale consistency near the manipulated objects. Object masks constrain occlusion decisions, and temporal hysteresis reduces unstable visibility changes across frames. The resulting videos are paired with the corresponding smoothed end-effector targets and gripper commands, with validity masks retained for downstream training, as shown in Figure 6.

Action speed alignment. To better match the motion speed of the robot data, we slow down EgoVerse and EgoDex trajectories by a factor of two. This doubles the trajectory duration while preserving the spatial path.

Algorithm 1 Coupled Base Search and Trajectory IK for Bimanual Arms

Require: Robot model $\mathcal{M}$; $\{\mathcal{T}_a, \mathcal{W}_a\}_{a\in\{L,R\}}$; acceptance criteria and search limits
Ensure: Base poses and joint trajectories $(b_L, b_R, \mathbf{Q}_L, \mathbf{Q}_R)$, or $\varnothing$

for $a \in \{L, R\}$ **do**
  $\mathcal{B}_a \leftarrow \mathrm{Candidates}(\mathcal{T}_a, \mathcal{W}_a)$
  for $s \in (\mathrm{coarse}, \mathrm{medium})$ **do**
    $\mathcal{B}_a \leftarrow \mathrm{SearchArmTrajectories}(\mathcal{B}_a, \mathcal{T}_a, \mathcal{W}_a; s)$
  end for
end for
$\mathcal{P} \leftarrow \mathrm{PairBases}(\mathcal{B}_L, \mathcal{B}_R)$
for $p = (b_L, b_R) \in \mathcal{P}$ in order **do**
  $(v, \mathbf{Q}_L, \mathbf{Q}_R) \leftarrow \mathrm{SolveFullTrajectory}(p, \mathcal{T}_L, \mathcal{T}_R)$
  if v **then**
    return $(b_L, b_R, \mathbf{Q}_L, \mathbf{Q}_R)$
  end if
end for
return $\varnothing$



Figure 6: Qualitative Ego2Robot synthesis results. After action and kinematic alignment, the robot is rendered from the original egocentric viewpoint and composited into the human-removed scene with depth-aware occlusion.

Table 8: Source-video coverage and robot training data produced by the Ego2Robot pipeline.

Dataset	Source data		Converted data		Final data		
	Hours	Episodes	Hours	Episodes	Robot-hours	Episodes	Frames (M)
EgoDex (Hoque et al., 2025)	772.16	310,418	310.03	149,753	1,702.33	1,041,346	183.852
EgoVerse (Punamiya et al., 2026)	3,289.19	1,312,338	791.51	133,697	3,931.43	2,689,167	424.410
Total	**4,061.35**	**1,622,756**	**1,101.55**	**283,450**	**5,633.77**	**3,730,513**	**608.262**

A.6 UMI Data

A.6.1 Data Sources

Hy-UMI-10K (Zhang et al., 2026a) The public release of Hy-UMI-10K contains approximately 250K episodes spanning 2,162 hours of bimanual manipulation. It pairs head- and wrist-camera RGB video with 6-DoF gripper poses, gripper openness, and task descriptions.

A.6.2 Filtering

Inspired by recent UMI data processing practices (Zhang et al., 2026a; Team et al., 2026), we apply the following filtering procedure. We first perform basic segment cleanup and reject any segment containing a gripper-pose quaternion q with $\|q\|_2 \leq 10^{-8}$. We then discard segments in which both hands are static, defined by each hand having position variance below $5 \times 10^{-4}\,\mathrm{m}^2$ and rotation variance below $0.1\,\mathrm{rad}^2$. We follow the trajectory anomaly filtering procedure described in section A.4.2 and additionally reject trajectories with excessive linear or angular speeds or abrupt position or orientation jumps between adjacent frames.

A.7 Data Filtering and Conversion Statistics

Table 7 reports dataset-level hours and episode counts before and after filtering, together with retained frame counts, for robot, egocentric human, and UMI data. Table 8 reports the ego2robot conversion of EgoDex and EgoVerse only: the filtered human videos used as input, the unique source coverage that completes synthesis, and the robot training data obtained from conversion.

Data filtering. After the processing rules above, the robot collection is reduced from 13,867.71 hours across 1,370,003 episodes to 11,302.20 hours across 1,247,656 episodes, comprising 1,101.047 million frames. EgoDex and EgoVerse jointly retain 4,061.35 of 4,640.04 hours and 1,622,756 of 1,818,396 episodes, totaling 438.626 million frames; the retained durations are 772.16 and 3,289.19 hours, respectively. Hy-UMI-10K is reduced from 2,161.74 hours across 250,135 episodes to 2,075.42 hours and 224.146 million frames. Retained UMI trajectories are segmented into multiple training episodes, so the episode count rises to 416,053.

Ego2Robot conversion. The source-data column of Table 8 is the filtered EgoDex and EgoVerse videos from Table 7. The converted-data column is successful source coverage: the union of intervals, across robot embodiments and processing batches, for which at least one embodiment completes synthesis. Each covered original source episode is counted once, even if only part of it is converted; only the successful intervals contribute to covered duration. The final-data column is the exported robot training set.

Completing synthesis requires a selected base configuration and a joint trajectory that satisfy the tracking, joint-limit, joint-discontinuity, and trajectory-completion criteria in Kinematic Alignment, together with base-separation and cross-arm collision checks for bimanual setups. An interval with no such configuration cannot be converted. Failures in subsequent synthesis stages, including visual alignment, likewise omit an interval from successful coverage, so unsuccessful coverage is not attributed solely to kinematic infeasibility.

Robot-hours sum usable durations across embodiments, reported before the factor-of-two slowdown in Action Speed Alignment, and robot episodes count the exported training segments of at least 32 action steps. One source episode can therefore yield multiple robot episodes, and the final training volume is larger than the unique source coverage. Across EgoDex and EgoVerse, the pipeline covers 1,101.55 hours from 283,450 unique source episodes and yields 5,633.77 robot-hours across 3,730,513 training episodes, totaling 608.262 million frames.

A.8 Training

We train InternW0-Δ in two stages. The first stage jointly optimizes a pretrained video expert and a randomly initialized action expert to a heterogeneous mixture of robot trajectories. The second stage specializes the resulting checkpoint to a target embodiment and task distribution.

A.8.1 Pre-training

Data and temporal sampling. Pre-training uses a heterogeneous hybrid dataset, including robot demonstrations (80%), ego-to-robot (Ego2Robot) data (10%), UMI-collected demonstrations (8%), and egocentric (Ego) demonstrations (2%). These data sources cover different robot forms, camera layouts, state conventions, and task vocabularies. Each data source is converted to a canonical 80-dimensional state and action space by its adapter before entering the hybrid dataset. The dataset streams are weighted according to the number of valid start frames, and a trajectory-level balancing mechanism prevents a small number of longer segments from dominating the hybrid dataset. We use fixed time windows for training instead of using complete segments. A training sample contains 33 video frames and a 32-step action block. The video stream and action stream are sampled at a fixed frequency ratio of 4; the most recent observation lags the current observation by 32 action steps, and the anchor point is taken from the beginning of the episode. Images are packed into a 384×256 training canvas used by the pre-training scheme.

We initially pretrain InternW0-Δ without 4D-aware representation distillation, and then continue pre-training with the auxiliary distillation objective described in Section 3.4 alongside the existing training objectives. During this final phase, each per-GPU mini-batch of 16 samples includes

Table 9: Pre-training settings.

Setting	Value
Canonical state/action dimension	80
Video resolution	384×256
Video frames	33
Action horizon	32
Video-to-action frequency ratio	4
Recent-frame offset	32 action steps
Batch size	16
Gradient accumulation	1
Optimizer	AdamW
Learning rate	5×10^{-5}
Weight decay	1×10^{-2}
Learning-rate schedule	5% warm-up followed by cosine decay
Mixed precision	bfloat16
Video expert	Wan2.2-TI2V-5B (Wan Team, 2025)
Action expert	ActionDiT with random initialization
VLM	RynnBrain1.1-2B (frozen) (Dang et al., 2026)
Video flow-matching loss weight	0.5
Action loss weight(Robot/UMI/Ego2Robot/Ego)	1.0/0.5/0.1/0.1
Causal Imprint loss weight	0.5
Future-feature alignment loss weight	0.1
Distillation loss weight	0.1

one sample paired with an offline-cached Track4World (Lu et al., 2026) descriptor. The auxiliary distillation loss is applied only to this sample.

Initialization. The video expert model is initialized using Wan2.2-TI2V-5B (Wan Team, 2025). The action expert model is randomly initialized. The visual language encoder is initialized using RynnBrain1.1-2B (Dang et al., 2026) and remains frozen during training. The two language pathways, T5 (Raffel et al., 2020) and VLM, assume different responsibilities during training. Instructions are converted into T5 embeddings for use by the video expert, thus preserving the pre-trained language-video interface of Wan2.2. T5 receives instructions but not the current scene, so its embeddings alone cannot determine which object is being referred to, its position in the camera view, or the progress of the task. To provide this missing observational basis, a valid current view and the same instructions are jointly passed to the RynnBrain1.1-2B visual language model. Its dense hidden states are appended with proprioceptive token and provided to the action expert.

Implementation and hyperparameters. We pre-train our model on 256 NVIDIA A800 GPUs using bfloat16 mixed precision. The per-GPU batch size is 16, with no gradient accumulation, resulting in an effective global batch size of 4,096. We use the AdamW optimizer with a learning rate of 5×10^{-5} and a weight decay of 1×10^{-2}. The learning-rate schedule consists of a 5% warm-up phase followed by cosine decay. The weights for the video flow-matching, Causal Imprint latent-difference, and future-feature alignment losses are 0.5, 0.5, and 0.1, respectively, as summarized in Table 9. We use data-source-dependent action loss weights: 1.0 for robot demonstrations, 0.5 for UMI-collected demonstrations, and 0.1 for both ego-to-robot (Ego2Robot) data and egocentric (Ego) demonstrations. Each weight scales only the action loss of samples from the corresponding data category. Pre-training consists of two consecutive phases. We first train for 235K optimization steps without 4D-aware representation distillation. We then continue training for an additional 10K steps, adding the distillation objective described in Section 3.4 with a weight of 0.1 while retaining the existing training objectives. The complete pre-training run comprises 245K optimization steps and takes approximately 14 days. The main hyperparameters are summarized in Table 9.

A.8.2 POST-TRAINING FOR SIMULATION

We post-train InternW0-Δ independently on four simulation benchmarks: EBench (Gao et al., 2026), RoboDojo (Chen et al., 2026b), LIBERO-Plus (Fei et al., 2026), and RoboTwin 2.0 (Chen et al., 2026c). Each experiment starts from **the same pretrained checkpoint** and uses a benchmark-specific data adapter to convert the native observation, state, and action interfaces into our unified representation. Native action dimensions are mapped into the canonical 80-dimensional action space, while dimensions that are not defined for a given embodiment are masked from the training objective.

Unless otherwise stated, we use a batch size of 16, eight data-loading workers per process, bfloat16 mixed precision, and a learning rate of 5×10^{-5}. Each sample is constructed from a 33-step trajectory window and contains a 32-step action chunk. Visual observations are sampled every four action steps, resulting in nine observation frames for each training sample. Multi-view observations are packed into a common 384×256 canvas across all four benchmarks. The benchmark-specific configurations are summarized in Table 10.

EBench (Gao et al., 2026). EBench is a mobile bimanual manipulation benchmark containing 26 tasks that cover tabletop manipulation, pick-and-place, and long-horizon interaction. The benchmark is designed to test a broad range of manipulation capabilities and generalization factors, including tasks that require coordinated dual-arm manipulation and mobile-base motion. We use the three RGB observations provided by the benchmark, consisting of one external view and two wrist views, and pack them into the common 384×256 input canvas.

The control interface contains end-effector targets for both arms, gripper commands, and planar mobile-base motion. We convert each arm pose into the unified end-effector representation, reduce the two finger coordinates of each gripper to a single gripper value, and retain the three-dimensional base motion command. These components are then mapped into their corresponding entries in the canonical action space. We use z-score normalization and post-train for 10 epochs.

RoboDojo (Chen et al., 2026b). RoboDojo provides 42 simulation tasks for bimanual manipulation and organizes them along five capability dimensions: generalization, memory, precision, long-horizon execution, and open-vocabulary instruction following. The tasks are designed to evaluate more than short-horizon pick-and-place behavior and include challenging multi-stage and precision-sensitive interactions. For post-training, we use the high camera together with the left and right wrist cameras and pack the three views into a 384×256 canvas.

The standard simulated embodiment uses a 14-dimensional bimanual joint-space interface, with six arm joints and one gripper coordinate for each arm. The benchmark-specific adapter maps the valid joint and gripper channels into the canonical action representation and masks dimensions that are not used by this embodiment. We use z-score normalization and post-train for 10 epochs.

LIBERO (Liu et al., 2023). LIBERO is a single-arm manipulation benchmark built around a Franka robot and contains four standard task suites: LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-10. We post-train jointly on demonstrations from these four suites. Each observation contains an external scene view and a wrist view, which are packed into the common 384×256 input representation.

The native action consists of a six-dimensional end-effector command and a scalar gripper command, which are mapped into the corresponding entries of the canonical action space. We use min–max normalization and post-train for 15 epochs.

We evaluate the resulting checkpoint on the LIBERO-Plus robustness setting. While standard LIBERO measures performance close to the post-training distribution, LIBERO-Plus preserves the underlying task semantics while introducing controlled distribution shifts along seven dimensions, including object layout, camera viewpoint, robot initialization, language instruction, lighting, background appearance, and sensor noise. This provides a direct measure of how well the post-trained policy generalizes when the visual, linguistic, and execution conditions deviate from those observed during training.

Table 10: Simulation post-training configurations. LIBERO-Plus is used only for evaluation. RoboTwin 2.0 is trained on the clean split and evaluated on the randomized.

Benchmark	Training split	Views	Resolution	Normalization	Epochs
EBench (Gao et al., 2026)	EBench train	3	384×256	Z-score	10
RoboDojo (Chen et al., 2026b)	RoboDojo train	3	384×256	Z-score	10
LIBERO-Plus (Fei et al., 2026)	LIBERO (Liu et al., 2023) train	2	384×256	Min–max	15
RoboTwin 2.0 (Chen et al., 2026c)	Clean	3	384×256	Z-score	10

Table 11: Real-robot datasets used for post-training and evaluation.

Task	Episodes	Frames	Duration (min)	Epochs
AC-One (gripper)				
Luminol reaction	162	433,715	240.95	10
Get a drink	137	244,440	135.80	10
Toast bread	304	311,631	173.13	10
MOF experiment	350	1,173,626	652.02	10
Arx5 (gripper)				
Magnetic stirrer	202	66,353	36.86	10
Place tube	2,664	555,702	308.72	10
Franka+XHand (dexterous hand)				
Pour water	50	44,704	24.84	100
Place fruit into box	48	39,253	21.81	100
Stack cups	54	48,264	26.81	100
TianJi Marvin+Wuji Hand (dexterous hand)				
Use dropper	101	107,703	59.84	50
Make a sandwich	101	106,731	59.30	50

Note. "Epochs" denotes task-wise post-training epochs.

RoboTwin 2.0 (Chen et al., 2026c). RoboTwin 2.0 is a bimanual manipulation benchmark containing 50 tasks for the ALOHA-AgileX embodiment. We use three RGB observations, consisting of one high camera and two wrist cameras, and pack them into a 384×256 training canvas. The native 14-dimensional action space contains six joint coordinates and one gripper coordinate for each arm, which are mapped to the corresponding bimanual entries of our canonical action representation.

Following the Clean2Random evaluation protocol, we post-train exclusively on demonstrations collected under the clean environment configuration and evaluate the same checkpoint under both clean and progressively randomized environments. The randomized evaluation introduces controlled changes in background appearance, lighting, scene clutter, and tabletop height, with the Hard setting combining multiple sources of variation. This setting directly measures whether the policy can retain its manipulation capability when the visual and physical environment deviates from the post-training distribution. We use z-score normalization and post-train for 10 epochs.

A.8.3 Post-Training for Real Robots

Setup. We post-train InternW0-Δ on 4 robot setups, including two **gripper-based** ones and **dexterous-hand** ones. For **gripper-based** setups, task-specific datasets are collected on the ARX AC-One real-robot platform and Arx5 single-arm robot. For **dexterous-hand** setups, we post-train the same pretrained checkpoint on the bimanual Franka+XHand platform and TianJi Marvin+Wuji Hand combination. We have collected 11 tasks across all 4 setups, whose details are summarized in Table 11. The post-training settings for real robots are included in Table 12. The runs share the model and optimizer configuration and start from the same pretrained checkpoint; task-wise training epochs differ as listed in Table 11.

Table 12: Post-training settings for real-robots.

Setting	Value
Canonical state/action dimension	80
Video resolution	384×256
Video frames	33
Action horizon	32
Video-to-action frequency ratio	4
Recent-frame offset	32 action steps
Batch size	16
Gradient accumulation	1
Optimizer	AdamW
Learning rate	5×10^{-5}
Weight decay	1×10^{-2}
Learning-rate schedule	5% warm-up followed by cosine decay
Mixed precision	bfloat16
Video expert	Wan2.2-TI2V-5B (Wan Team, 2025)
Action expert	ActionDiT
VLM	RynnBrain1.1-2B (frozen) (Dang et al., 2026)
Video loss weight	0.5
Action loss weight	1.0
Causal Imprint loss weight	0.5
Future-feature alignment loss weight	0.1
Distillation loss weight	0.1

RTC post-training. During real-robot post-training, we train the policy to predict 32-step action chunks. Following training-time RTC (Black et al., 2025d), we simulate inference delay by sampling a committed prefix length $d \sim \mathcal{U}\{0, \ldots, 16\}$ for each full-length training sample; for padded samples, the maximum prefix length is capped by the valid sequence length to retain at least one supervised suffix action. The first d ground-truth actions remain clean, with their per-token flow timesteps set to zero, while noise is applied to the remaining actions. The action loss is computed only over valid suffix tokens and action dimensions. This trains the policy to predict an action continuation conditioned on a known prefix. At deployment, the prefix is supplied by the previously generated plan, as described in Section A.9.2.

A.9 Training and Deployment Infrastructure

A.9.1 Training Infrastructure for Model Development

During model iteration, we identified two main sources of overhead: repeated extraction of visual features from samples revisited across epochs, and the compute and activation memory required by the MoT backbone. We address these costs at complementary stages of the training pipeline. A unified cache reuses the outputs of the frozen VAE and VLM, while layerwise compilation and checkpointing improve the execution of the MoT backbone. Together, these optimizations reduce redundant encoder work and allow the backbone to be configured for either throughput or activation memory.

Feature Caching For a training sample x_i, let

$$z_i = E_{\mathrm{VAE}}(x_i), \qquad h_i = E_{\mathrm{VLM}}(x_i, p_i, s_i), \tag{24}$$

where z_i is the video latent, h_i is the VLM context, and (p_i, s_i) denote the task prompt and state. Because the same samples are revisited during training, repeatedly evaluating the frozen encoders expends computation without changing these representations. Feature caching therefore aims to amortize encoding across training steps while preserving preprocessing, temporal alignment, camera ordering, and model semantics.

Figure 7 shows the Cache Manager between the data loader, feature encoders, and storage backends. Built on LiteGen (contributors, 2026), it performs lookups, validates artifact metadata, dispatches

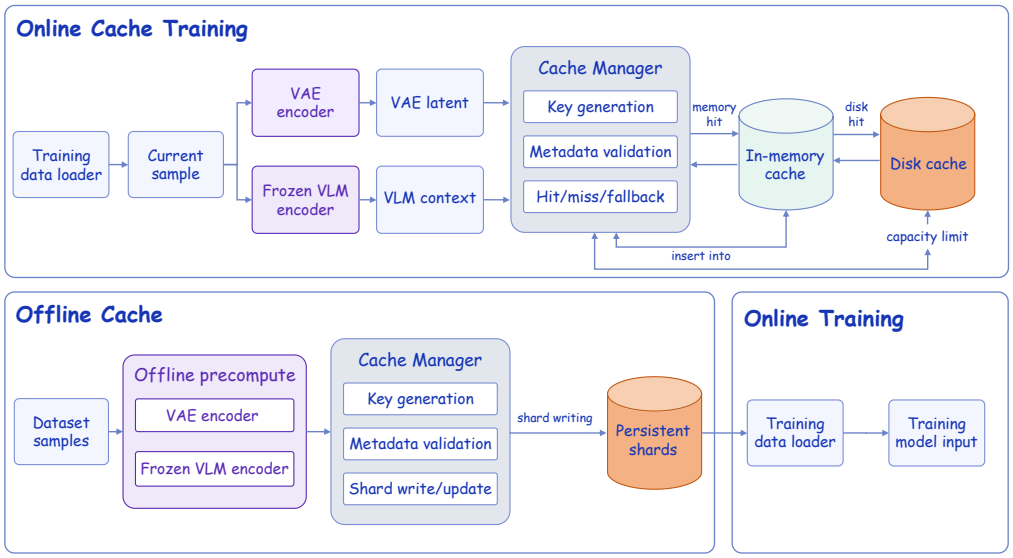


Figure 7: Feature caching for exploratory training. The Cache Manager serves VAE latents and VLM contexts to training through a common interface for online and offline materialization, reducing repeated feature extraction during model development.

reads and writes, and records hits, misses, and fallbacks. To prevent reuse across incompatible data or encoder configurations, each artifact is identified by its sample and representation provenance. A representative key is

$$\text{key} = \text{hash}(D, e, k, v, P, M), \tag{25}$$

where D, e, k, and v identify the dataset version, episode, temporal chunk, and camera view, and P and M identify the preprocessing and model versions. VLM artifacts must additionally account for any prompt or state inputs that affect h_i. Metadata validation rejects incompatible or corrupted artifacts; an online miss can then fall back to encoding rather than silently consuming an invalid feature.

The same artifact contract supports two materialization strategies. In *online caching*, the manager first checks memory and then disk for each sample. A miss invokes the corresponding VAE or frozen VLM encoder, after which the feature is inserted into memory and, when needed, persisted to disk. Entries are thus populated as the dataset is traversed: this avoids a separate preprocessing job, although the first pass still pays the encoding cost. Asynchronous persistence limits the impact of writes on training. In *offline caching*, batched encoder inference produces persistent feature shards before training begins. A manifest records artifact keys, coverage, tensor shapes and data types, preprocessing versions, and model fingerprints; a startup check validates the manifest and index. The one-time preprocessing cost can then be amortized over subsequent epochs and repeated experiments using the same data and encoder versions.

Layerwise MoT Optimization The MoT backbone contains 30 blocks. In each block, a video expert and an action expert update separate token streams while sharing mixed self-attention. The experimental configuration uses a 3,072-dimensional Wan video stream and a 1,024-dimensional action stream; the latter also attends to text, the current VLM context, and proprioceptive tokens. Our objective is to improve the throughput and memory efficiency of this backbone while preserving its attention routing and intended gradients.

As illustrated in Figure 8, we compile each MoT layer as an independent PyTorch Inductor graph and invoke the layers from an eager orchestrator. Layer boundaries constrain recompilation, allow per-layer profiling, and retain opportunities to overlap ZeRO communication with computation. We compile supported regions while leaving operations that Inductor cannot lower in eager execution. Mixed attention uses PyTorch FlexAttention with a 64-token block size; constructing the block mask

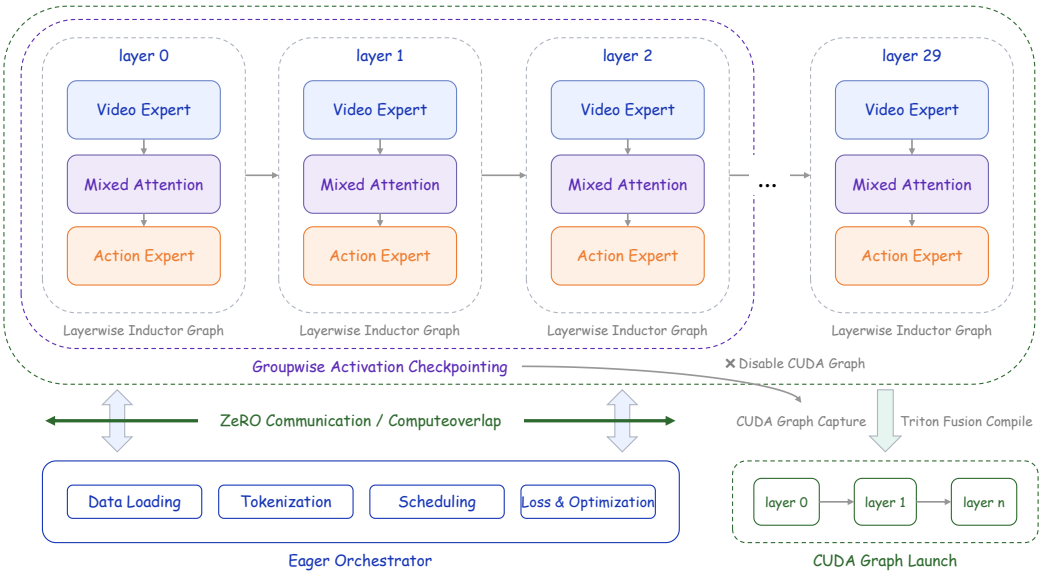


Figure 8: Layerwise compilation of the two-stream MoT backbone. An eager orchestrator invokes independently compiled layers and groups three consecutive layers for external non-reentrant checkpointing when memory reduction is required. VLM contexts are padded before action cross-attention.

outside the compiled layer keeps the causal and stream-specific routing rules consistent across execution modes. To stabilize the compiled action-layer shapes, we pad VLM contexts and validity masks to 640 tokens. The current LIBERO (Liu et al., 2023) cache reaches 614 tokens after appending the proprioceptive token; added positions are masked, and longer contexts are rejected rather than truncated.

For throughput-oriented training, the `reduce-overhead` mode uses layerwise compilation with CUDA Graph capture where eligible. When activation memory is the limiting factor, the eager orchestrator instead places three consecutive compiled layers inside each external non-reentrant checkpoint region. This recomputes the grouped activations during backward while retaining independent compiler graphs for individual layers; the expert-internal and eager mixed-attention checkpoint switches are disabled for this configuration. CUDA Graph capture is also disabled on the checkpointed path because its interaction with AOTAutograd and non-reentrant checkpointing produced incorrect gradients in validation.

Training Efficiency The proposed optimizations can be enabled independently according to the data characteristics and resource constraints at different stages of model development. On LIBERO (Liu et al., 2023) and RoboTwin (Chen et al., 2026c), combining VAE/VLM feature caching with the workload-specific layerwise MoT configuration yields end-to-end training throughput speedups of $3.02\times$ and $2.11\times$, respectively, over the corresponding unoptimized baselines. When stochastic inputs preclude feature reuse, or when the dataset makes cache materialization impractical, the throughput-oriented layerwise MoT optimization remains applicable on its own and improves end-to-end throughput by 20–30%. These optimizations reduce the time for architecture and hyperparameter exploration across training regimes.

A.9.2 Real-Robot Deployment

Deployment setup. We deploy InternW0-Δ with inference running locally on a single NVIDIA RTX 5090 GPU with 32 GiB of memory and a control rate of 30 Hz. The observation configuration varies by platform: AC-One uses one head camera and two wrist cameras; Arx5 uses one wrist camera and two external cameras; bimanual Franka+XHand uses a single external camera; and TianJi Marvin+Wuji Hand uses one head camera and one wrist camera.

Table 13: Cumulative inference-latency ablation on dexterous-hand deployment. Success rates are from separate LIBERO-plus (Fei et al., 2026) simulations.

Configuration	Round-trip time (ms) ↓	Speedup	Success rate (LIBERO-plus, %)
Standard runtime	780.5	1.00×	92.78
+ process isolation	374.1	2.09×	92.67
+ feature caching & compilation	249.8	3.12×	92.46
+ context caching	217.3	3.59×	92.41
+ grouped action execution	186.2	4.19×	92.33
+ CUDA-graph replay	152.8	5.11×	92.23

Asynchronous execution strategy. Asynchronous execution requires each new action chunk to remain consistent with actions already committed to execution. Existing strategies address this requirement through action blending, inference-time guidance, or training-time prefix conditioning. A recent empirical study of real-time World Action Models compares these approaches and documents their trade-offs in precision, smoothness, and execution efficiency (Motubrain Team, 2026).

We use the prefix-conditioned policy trained in Section A.8.3, supplying committed actions as clean inputs during sampling. This avoids additional gradient-based RTC guidance (Black et al., 2025c) and is compatible with our compiled deployment pipeline.

The controller retains a 32-step prediction horizon and launches a single background inference request every 16 execution steps while the remaining actions of the current plan execute. The worker receives a fresh observation and a temporally aligned prefix from the previous plan. Prefix tokens are held clean at timestep zero and re-clamped to the supplied actions throughout sampling. The prefix length is estimated conservatively from recent inference delays and capped at the trained maximum of 16 steps.

Inference optimization. Asynchronous inference must complete within a bounded time window. After a replan is issued, the controller must receive the next action chunk before the remaining 16 actions are exhausted. At 30 Hz, this window is 533 ms; observation transport, policy inference, prefix alignment, and controller-side command dispatch all consume it, and robot-side scheduling jitter reduces the remaining margin. The policy must therefore meet a round-trip latency constraint, not merely attain high nominal throughput.

At inference, InternW0-Δ executes the action-generation path of the directed MoT. The runtime first isolates policy inference in a separate rclpy-free process, preventing ROS callbacks, timers, and Python GIL scheduling in the robot node from interfering with CUDA launch execution. The remaining runtime exploits two invariants of this path. First, conditioning features and their attention K/V projections are fixed within one action invocation and are reused across diffusion steps. Second, the action hot path has fixed tensor shapes for a given deployment configuration. Cached K/V tensors are packed across layers, groups of consecutive MoT action layers are compiled as fixed-shape callables, and CUDA Graph Trees replay these groups to reduce launch overhead. These options do not change the model inputs, diffusion schedule, attention semantics, action interface, or RTC prefix contract.

Table 13 reports a cumulative ablation of controller-observed round-trip latency on the dexterous-hand deployment. Each request is timed from when the controller issues a replan to when the returned action chunk is available. The RTT includes controller-side request handling and transport to the isolated policy process, as well as observation preprocessing, state normalization, recent-frame/prefix alignment, conditioning, action inference, action mapping, and output post-processing. Action inference runs on one RTX 5090 GPU. Three warm-up requests are discarded, and the following 50 requests are averaged per configuration.

A.10 Complete Experimental Results

A.10.1 Evaluation Protocol

We evaluate InternW0-Δ on four simulation benchmarks, including LIBERO-Plus (Fei et al., 2026), RoboTwin 2.0 (Chen et al., 2026c), EBench (Gao et al., 2026), and RoboDojo (Chen et al., 2026b). The post-training setup for each benchmark has been described in Section A.8.2. For each benchmark, we follow its official simulator, task definitions, and success criteria without additional adaptation on the evaluation environments. Baseline results are taken from the corresponding papers or public benchmark results under matching evaluation protocols.

A.10.2 Simulation Benchmark Results

LIBERO-Plus We train InternW0-Δ on the standard LIBERO (Liu et al., 2023) training set and directly evaluate it on LIBERO-Plus (Fei et al., 2026) without adaptation to the evaluation environments. As shown in Table 1, InternW0-Δ achieves the best overall success rate of 92.8%, outperforming Qwen-RobotManip-Context (Yuan et al., 2026a) by 1.4 percentage points and the strongest prior WAM, Being-H0.7 (Luo et al., 2026), by 8.0 points. The largest gains appear under robot perturbations, where InternW0-Δ reaches 91.1%, compared with 87.4% for the previous best result. It also achieves the best performance under camera and language perturbations, reaching 90.6% and 92.9%, respectively. In addition, InternW0-Δ remains competitive on background and layout perturbations, with 99.4% and 88.4% success rates.

RoboTwin 2.0 We train InternW0-Δ using clean demonstrations only and evaluate it under both Clean2Clean and Clean2Random. The compared baseline results also use clean-only training and are taken from the corresponding papers or public benchmark results. As shown in Table 2, InternW0-Δ achieves the best performance in both settings, reaching 90.0% on Clean2Clean and 71.9% on Clean2Random, with an overall success rate of 81.0%. Compared with Qwen-RobotManip-Context (Yuan et al., 2026a), the strongest VLA baseline, InternW0-Δ improves Clean2Random by 2.5 percentage points and the overall score by 3.9 points. The advantage over existing WAMs is larger: compared with OpenWAM-α (Wang et al., 2026d), InternW0-Δ improves Clean2Random from 48.7% to 71.9%, while also slightly improving Clean2Clean from 89.4% to 90.0%. These results show that the improvement on randomized environments is achieved without sacrificing performance on the original clean distribution.

EBench We further evaluate InternW0-Δ on EBench (Gao et al., 2026), which covers precision-sensitive Table Top tasks, mobile pick-and-place tasks, and multi-stage Long Horizon manipulation. As shown in Table 3, InternW0-Δ achieves an overall success rate of 49.2% and the highest overall score of 66.0, demonstrating competitive performance across different manipulation settings.

The advantage of InternW0-Δ is particularly clear on Long Horizon tasks, where it achieves 49.4% success rate and a score of 76.5, both the highest among the compared methods. Since these tasks require multiple dependent operations over extended interaction sequences, the results indicate that InternW0-Δ can effectively maintain task progress and execution consistency over long-horizon manipulation. Together with its strong performance on Simple PnP, these results show that InternW0-Δ generalizes well from basic mobile manipulation to more temporally extended behaviors.

RoboDojo As shown in Table 4, InternW0-Δ achieves highly competitive overall performance, reaching an average SR of 23.91% and a score of 30.77. Compared with DM0.5 (Dexmal, 2026), a strong VLA baseline in terms of average performance, InternW0-Δ improves the average SR by 4.57 percentage points and the score by 5.87 points. The improvement over existing WAMs is more substantial: compared with OpenWAM-α (Wang et al., 2026d), InternW0-Δ increases the average SR from 11.92% to 23.91% and the score from 17.18 to 30.77. In individual categories, InternW0-Δ shows particularly strong performance on Gen-Std and Precision, reaching SRs of 33.78% and 23.25%, respectively, while also achieving a Long-Horizon score of 46.29. Performance is weaker on Gen-Rand and Open tasks, where GPT-6 Astra (Zhang et al., 2026c) shows stronger performance. Overall, InternW0-Δ also slightly exceeds GPT-6 Astra in average SR and score, reaching 23.91% and 30.77 compared with 22.48% and 28.97, respectively.

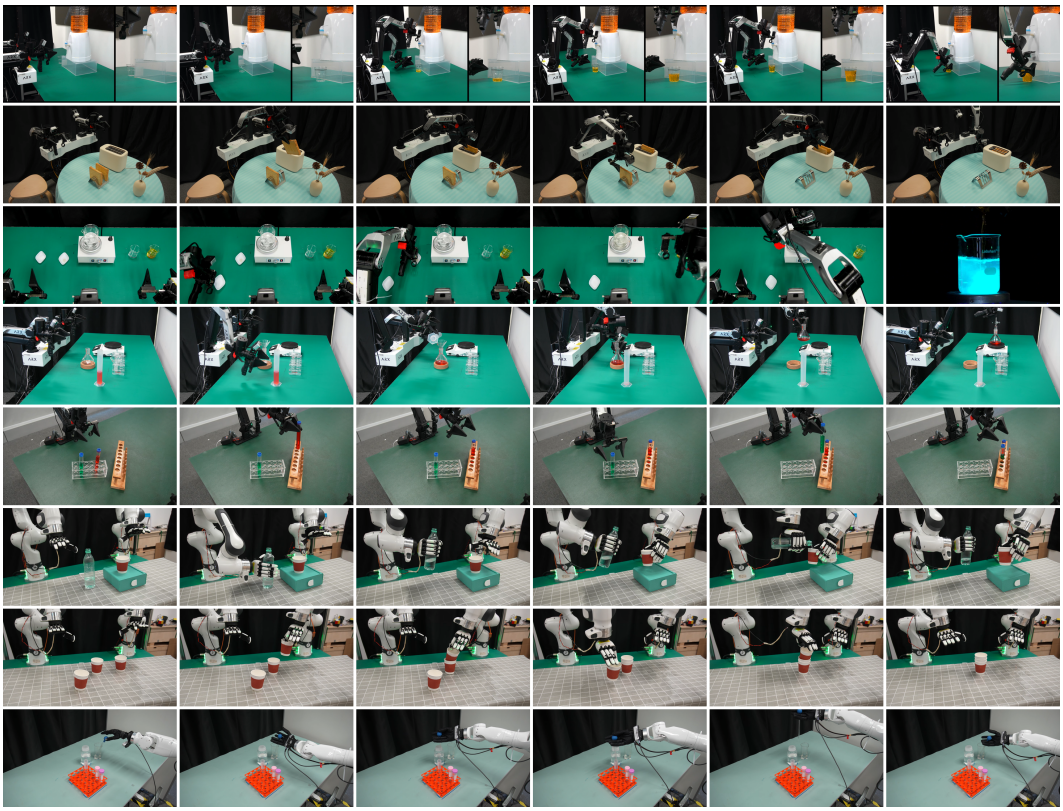


Figure 9: **Real-robot task execution.** Selected video frames illustrate eight tasks, with each row ordered chronologically from left to right. From top to bottom: *get a drink*, *toast bread*, *Luminol reaction*, *MOF experiment*, *place tube*, *pour water*, *stack cups*, and *use dropper*. The final frame of the Luminol sequence shows a close-up of the reaction outcome.

A.11 Real-Robot Experiments

Tasks and evaluation procedure. We evaluate eleven tasks across four real-robot platforms: AC-One, Arx5, bimanual Franka+XHand, and TianJi Marvin+Wuji Hand. On AC-One, the tasks comprise *get a drink*, *toast bread*, *Luminol reaction*, and *MOF experiment*. *Get a drink* episodes place a cup under the dispenser and fill it to the marked level before moving it to the right side of the dispenser; *toast bread* episodes place two bread slices into the toaster in sequence and press its switch; *Luminol reaction* episodes add NaOH, luminol, hydrogen peroxide, and potassium ferricyanide in sequence and mix the reaction; *MOF experiment* episodes pour a measured solution through a funnel into a flask, return the funnel, and place and seal the flask on the stirrer. On Franka+XHand, we evaluate *pour water*, *place fruit into box*, and *stack cups*. These tasks respectively pour water from a bottle into a disposable paper cup, place the language-specified fruit into a box, and stack disposable cups. On Arx5, *magnetic stirrer* places a magnetic stir bar into a beaker and then places the beaker on the magnetic stirrer; *place tube* transfers a test tube to a wooden test-tube rack. On TianJi Marvin+Wuji Hand, *use dropper* picks up a rubber-bulb dropper, draws liquid from a reagent bottle, and transfers it into a test tube held in a rack. The *make a sandwich* task places a bread slice from the bread rack onto a plate, sequentially layers cheese, egg, and lettuce on top, and adds a second bread slice to complete the sandwich. On these four platforms, each rollout starts from the robot's home pose with a text instruction and runs under asynchronous training-time RTC (Section A.9.2), which achieves real-time deployment. Figure 9 illustrates representative execution sequences for eight of these tasks.

Effect of pretraining on real-robot performance. We compare models with and without our pretraining stage after task-specific post-training on two AC-One tasks, using 20 evaluation trials



Figure 10: **Generalization to different target water levels.** For the AC-One *get a drink* task, the rightmost cup shows the marked target level used throughout post-training data collection. The three cups on the left show the outcomes of three successful executions at different marked target levels. Black markings indicate the target levels.

Table 14: Real-robot performance with and without our pretraining stage, followed by task-specific post-training. Each entry reports successful trials out of 20 and the corresponding success rate.

Task	Without pretraining	With pretraining
Toast bread	4/20 (20%)	19/20 (95%)
Luminol reaction	0/20 (0%)	19/20 (95%)

per task and condition. As shown in Table 14, pretraining increases success rates from 20% to 95% on *toast bread* and from 0% to 95% on *Luminol reaction*, supporting its benefit for these downstream real-robot tasks.

Generalization across physical conditions. Beyond the training conditions, the post-trained policy exhibits qualitative generalization capabilities to unseen target water levels. For the AC-One *get a drink* task, all 137 post-training demonstrations use the same marked target water level. At deployment, the policy successfully fills cups to different marked target levels, indicating that its behavior is not restricted to the single target level represented in the demonstrations. Figure 10 shows three successful executions alongside the training target-level reference.

Cross-embodiment adaptation. The same pretrained checkpoint is post-trained separately for the **gripper-based** and **dexterous hands** interfaces. For grippers, AC-One uses 14-dimensional bimanual absolute joint targets and Arx5 utilizes 6-dimensional end-effector poses plus a 1-dimensional gripper. For dexterous hands, Franka+XHand uses 6-dimensional end-effector commands per Franka arm together with 12-dimensional dexterous hand commands per hand, whereas TianJi Marvin+Wuji Hand is controlled by 7-dimensional arm joint action and 20-dimensional hand action. All four settings therefore require different post-training action adapters while sharing the same pretrained model.

RTC continuity comparison. The three RTC strategies exhibit distinct continuity behaviors in the recorded AC-One runs (Figure 11). In the illustrated trace, VJP (Black et al., 2025c) shows an abrupt change in the published action commands at the chunk boundary, with a boundary-to-within-plan change ratio of 8.94. Hard prefix (Motubrain Team, 2026) preserves the committed actions at the chunk boundary, but introduces a sharp jetting event at the prefix-to-suffix boundary, where the model switches from the committed prefix to freely generated actions. Its corresponding

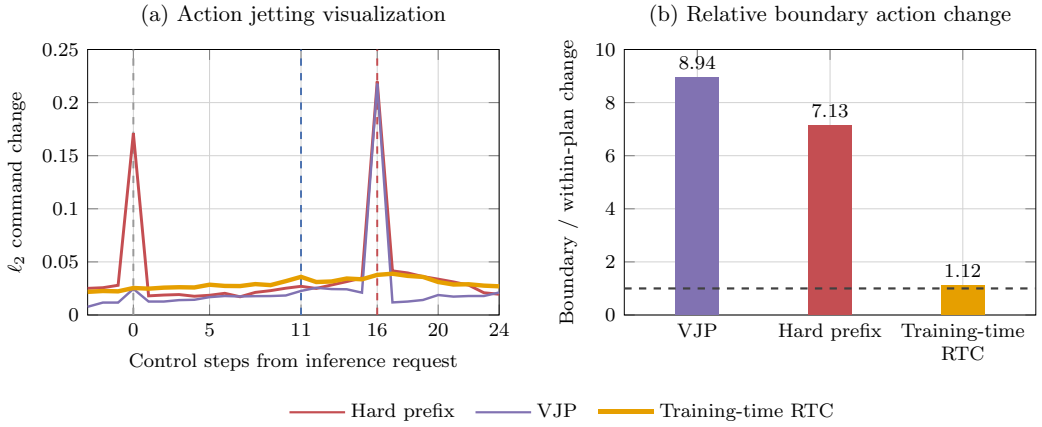


Figure 11: **Action jetting visualization.** (a) ℓ_2 changes between consecutive published 14-dimensional action commands, aligned to the inference request at $t = 0$, with t measured in published control steps. Hard prefix and training-time RTC are shown as mean curves with chunk switches at $t = 10$; VJP is shown as an individual trace with its chunk switch at $t = 16$. The gray dashed line at $t = 0$ also marks the previous hard-prefix plan's prefix-to-suffix boundary. The blue dashed line at $t = 11$ marks the training-time prefix-to-suffix boundary. The red dashed line at $t = 16$ marks both the hard-prefix prefix-to-suffix boundary and the VJP chunk boundary. (b) Mean boundary command change divided by the mean consecutive-command change within the executed new-plan segments, excluding the evaluated boundary. The evaluated boundary is the chunk boundary for VJP and the prefix-to-suffix boundary for hard prefix and training-time RTC. The dashed horizontal line denotes a ratio of one.

ratio is 7.13. Training-time conditioning (Black et al., 2025d) keeps changes at the prefix-to-suffix boundary comparable to ordinary within-plan changes, with a ratio of 1.12. These diagnostic traces support our use of training-time prefix conditioning for continuous asynchronous execution.

A.12 Additional Ablation Analysis

Sparse Memory Context. We first investigate whether lightweight temporal context benefits action prediction. Adding SMC improves the success rate from 49.59% to 53.47%, a gain of 3.88 percentage points. SMC augments the current observation with an episode-level anchor and a recent observation preceding the previous action chunk, providing both coarse task context and short-term execution history. The improvement indicates that a single current observation is insufficient to fully capture the interaction state, while a small number of carefully selected historical observations already provides useful temporal information without maintaining a dense visual history.

VLM. We next investigate the effect of the vision-language representation supplied to the action expert. With SMC fixed, introducing Qwen3.5-2B improves the success rate from 53.47% to 60.37%, showing the benefit of jointly reasoning over the current visual scene and task instruction. Replacing Qwen3.5-2B with RynnBrain1.1-2B further increases the success rate to 69.08%, corresponding to an additional gain of 8.71 percentage points. Since all variants use the same training and evaluation settings, this comparison shows that the representation provided by the VLM has a strong effect on downstream manipulation performance. We therefore adopt RynnBrain1.1-2B in the remaining experiments.

Causal Imprint. We then investigate Causal Imprint, which is designed to encode information about how the observed scene is likely to evolve while using only observations available to the policy at inference time. CI is trained with its latent-difference objective $\mathcal{L}_{\Delta}$, which directly supervises the imprint representation using changes between consecutive future video latents. Introducing CI improves the success rate from 69.08% to 70.80%. We further introduce the representation alignment objective $\mathcal{L}_{\text{align}}$, which aligns Causal Imprint tokens with spatially corresponding represen-

Table 15: Ablation of distillation design on LIBERO-Plus (Fei et al., 2026). All variants include SMC, RynnBrain1.1-2B (Dang et al., 2026), CI, and $\mathcal{L}_{\text{align}}$. Action injection indicates whether the student descriptor is additionally fed into the action expert.

Teacher	Action injection	Success rate (%) ↑
None	No	76.45
CoWTracker	No	76.15
Pi3X	No	73.63
Track4World	Yes	77.78
Track4World	No	**78.37**

tations from the future video features. This increases the success rate substantially from 70.80% to 76.45%, a gain of 5.65 percentage points. The two objectives provide complementary signals: $\mathcal{L}_{\Delta}$ directly describes the temporal changes along the future trajectory, whereas $\mathcal{L}_{\text{align}}$ transfers the richer semantic and temporal representation learned by the video expert. Together, they encourage Causal Imprint to capture not only low-level visual changes but also higher-level information about the subsequent evolution of the scene. Importantly, the future observations are used only to construct supervision during training. The Causal Imprint tokens themselves cannot directly attend to the realized future, and the action expert therefore receives no privileged future observation during either training or inference. This allows future trajectories to shape the learned representation without requiring explicit future-video generation for online action prediction.

4D-aware representation distillation. As shown in Table 5, adding $\mathcal{L}_{\text{4D}}$ improves the overall success rate on LIBERO-Plus from 76.45% to 78.37%, a gain of 1.92 percentage points. We further investigate the teacher choice and action feature injection in Table 15. Distillation from Track4World (Lu et al., 2026) achieves the highest success rate of 78.37%, compared with 76.15% for CoWTracker (Lai et al., 2026) and 73.63% for Pi3X (Wang et al., 2025). Additionally, injecting the student descriptor into the action expert yields a slightly lower success rate of 77.78%. We therefore adopt auxiliary distillation of the video expert without action injection, allowing the student branch to be removed at inference with no additional policy inference cost.

A.12.1 Data Source Ablations

We investigate how different data sources contribute to model performance. We consider four types of data, including raw robot demonstrations, action-aligned human egocentric demonstrations (Ego), human demonstrations converted through our Ego2Robot pipeline (Ego2Robot), and UMI data. All variants use the same InternW0-Δ architecture and are pretrained using data from different sources. Comparisons on RoboTwin are grouped by their joint-space or EEF-space action interface; data-source effects must be compared against the baseline with the same interface, rather than across the two groups.

Across the evaluated settings, as shown in Table 16 and Table 17, robot demonstration data provides the strongest and most consistent benefit. On LIBERO-Plus, robot pretraining improves the overall success rate from 78.59% to 83.73%. The effect is particularly clear on RoboTwin 2.0 under the Clean2Random setting, where the success rate increases from 4.38% to 32.34%, indicating a substantial gain in robustness to visual and environmental variations. In comparison, action-aligned egocentric data brings only a modest improvement on the current benchmarks, increasing LIBERO-Plus performance to 80.45% and RoboTwin Clean2Random from 2.80% to 3.13%. Ego2Robot data yields a larger gain, reaching 81.86% on LIBERO-Plus and 6.66% on RoboTwin Clean2Random. UMI data provides an even stronger improvement, achieving 83.24% on LIBERO-Plus and increasing RoboTwin Clean2Random from 2.80% to 13.90%. Overall, these results suggest that human data become more effective when their observation and action spaces are better aligned with those of robots.

The results are specific to our current data scale and evaluation setting, with both LIBERO-Plus and RoboTwin primarily focusing on gripper-based manipulation. Under this setting, action-aligned egocentric data provides limited gains in task success rate, but this does not rule out stronger benefits at a larger scale or on tasks that are more closely aligned with human hand-object interaction. In par-

Table 16: Pretraining data-source ablation on LIBERO-Plus (Fei et al., 2026). The baseline uses no pretraining, whereas Robot, Ego, Ego2Robot, and UMI are pretrained exclusively on their respective data sources. All values are success rates (%).

Method	Camera	Robot	Language	Light	Background	Noise	Layout	Overall ↑
Baseline	60.10	76.77	88.68	95.01	81.23	75.58	78.69	78.59
Robot	**74.36**	68.39	93.69	96.67	81.97	**93.07**	80.85	**83.73**
Ego	65.60	74.26	91.93	94.83	**84.20**	79.01	78.82	80.45
Ego2Robot	63.10	79.81	93.69	96.76	82.62	82.70	79.15	81.86
UMI	65.92	**80.26**	**96.49**	**97.20**	79.83	84.82	**81.38**	83.24

Table 17: Pretraining data-source ablation on RoboTwin 2.0 (Chen et al., 2026c). Baseline-Joint and Baseline-EEF use no pretraining, whereas Robot-Joint, Ego2Robot-Joint, Ego-EEF and UMI-EEF are pretrained exclusively on their respective data sources. All values are success rates (%).

Method	Clean2Clean ↑	Clean2Random ↑	Overall ↑
Baseline-Joint	78.20	4.38	41.29
Robot-Joint	**81.36**	**32.34**	**56.85**
Ego2Robot-Joint	76.92	6.66	41.79
Baseline-EEF	61.58	2.80	32.08
Ego-EEF	61.49	3.13	32.31
UMI-EEF	**66.05**	**13.90**	**39.98**

ticular, human demonstrations may be more useful for dexterous manipulation, where fine-grained contact patterns and hand motions are less well covered by standard robot datasets. Moreover, our current evaluation only measures downstream success rate. Therefore, the limited improvement from Ego data should not be interpreted as evidence that it does not improve the learned representation, since such effects may not be fully reflected by success rate on the current benchmarks.

A.12.2 Exploration of GPT-Guided Policy

Recent work on GPT as an embodied policy has explored using GPT-6 Astra (Zhang et al., 2026c) together with $\pi_{0.5}$, where GPT monitors the execution of a VLA policy and provides corrective actions when necessary (Su et al., 2026). Their experiments on RoboDojo show that such test-time correction can improve the execution of a pretrained VLA policy (Su et al., 2026). Motivated by these results, we conduct a small-scale exploration to examine whether the same mechanism can also complement a World Action Model.

We select five RoboDojo tasks on which the standalone InternW0-Δ exhibits relatively low performance under seed 0, covering both semantic reasoning and manipulation challenges. As shown in Table 18, InternW0-Δ obtains an average success rate of 10.80% and an average score of 13.92 across these tasks. We then introduce GPT-6 Astra as an execution-time assistant that observes the interaction and provides end-effector (EEF) corrections when necessary. This increases the average success rate to 47.20% and the average score to 52.00, corresponding to absolute gains of 36.40 percentage points in success rate and 38.08 score points, respectively. The improvement is particularly clear on 'classify objects", where the success rate increases from 14% to 80%, and on 'general pickup", from 36% to 76%. Tasks with zero standalone success also become partially solvable: 'solve equation" and 'arrange largest number" reach 20% and 40% success rates, respectively. These results suggest that GPT-based execution correction can provide complementary high-level reasoning and online adjustment even when the underlying policy is a world-action model rather than a conventional VLA.

We further observe that the reasoning configuration of GPT-6 Astra can have a noticeable effect on correction quality, especially for tasks that require semantic understanding. For example, on "classify objects by language", using GPT-6 Astra with the xhigh reasoning setting together with InternW0-Δ successfully completes all five evaluated episodes. In contrast, the medium setting obtains only partial task scores and does not complete the full task in these trials. Although this

Table 18: Exploratory GPT-guided policy results on five selected RoboDojo tasks. SR is a percentage; the tasks were selected for low standalone performance under seed 0, so this is not a full-benchmark aggregate.

Task	Ours SR	Ours Score	+Astra SR	+Astra Score
Classify objects	14.00	22.80	**80.00**	**88.00**
Press by number	4.00	4.00	**20.00**	**20.00**
Solve equation	0.00	0.00	**20.00**	**20.00**
General pickup	36.00	36.00	**76.00**	**76.00**
Arrange largest number	0.00	6.80	**40.00**	**56.00**
Average	10.80	13.92	**47.20**	**52.00**

comparison is based on a small number of episodes, it indicates that stronger test-time reasoning may be particularly useful when policy correction requires interpreting language instructions, identifying task-relevant objects, and deciding how the underlying policy should be adjusted.

A.13 Human-to-Robot IK Details

At each local IK iteration for a fixed robot base placement, the solver combines position and approach tracking, contact-point and orientation refinement, and projected joint-space regularization:

$$\Delta\mathbf{q} = \underbrace{\Delta\mathbf{q}_p}_{\substack{\text{position and}\\ \text{approach tracking}}} + \underbrace{\Delta\mathbf{q}_s}_{\substack{\text{contact-point and}\\ \text{orientation refinement}}} + \underbrace{\mathbf{N}_p\mathbf{g}_{\text{aux}}}_{\substack{\text{joint-space}\\ \text{regularization}}}, \tag{26}$$

where $\mathbf{q}$ is the current joint configuration and $\Delta\mathbf{q}$ is its local update. The increment $\Delta\mathbf{q}_p$ tracks TCP position and approach direction, $\Delta\mathbf{q}_s$ refines contact-point alignment and orientation through the damped projection $\mathbf{N}_p$, and $\mathbf{N}_p\mathbf{g}_{\text{aux}}$ applies projected joint-space preferences. The subscript p denotes position and approach tracking, s denotes contact-point and orientation refinement, and μ is the pseudoinverse damping. The individual terms and update bounds are specified below.

Position and approach tracking. This step reduces TCP position error and approach-direction error beyond the allowed tolerance:

$$\Delta\mathbf{q}_p = \mathbf{J}_{p,\mu}^{\#}\mathbf{e}_p, \qquad \mathbf{N}_p = \mathbf{I} - \mathbf{J}_{p,\mu}^{\#}\mathbf{J}_p. \tag{27}$$

Here $(\mathbf{J}_p, \mathbf{e}_p)$ contains the weighted position-and-approach Jacobian and residual. The damped inverse is defined by $\mathbf{J}_{\mu}^{\#} = \mathbf{J}^{\top}(\mathbf{J}\mathbf{J}^{\top} + \mu\mathbf{I})^{-1}$, with adaptive damping $\mu > 0$.

Contact-point and orientation refinement. Optional gripper contact-point and full-orientation targets are refined after the position-and-approach step. Their correction accounts for the residual left by that step and uses the position-and-approach projection $\mathbf{N}_p$ to attenuate interference:

$$\Delta\mathbf{q}_s = \mathbf{N}_p(\mathbf{J}_s\mathbf{N}_p)_{4\mu}^{\#}(\mathbf{e}_s - \mathbf{J}_s\Delta\mathbf{q}_p), \tag{28}$$

where $(\mathbf{J}_s, \mathbf{e}_s)$ stacks the weighted Jacobians and residuals of the enabled contact-point and orientation terms. Contact-point alignment drives the left and right gripper-pad centers toward two virtual points on the human pinch axis. These targets specify geometric alignment. This term is zero when neither refinement is active.

Joint-space regularization. The auxiliary adjustment encourages continuity, a preferred home configuration, and separation from joint limits:

$$\mathbf{g}_{\text{aux}} = \lambda_r(\mathbf{q}^{\text{ref}} - \mathbf{q}) + \lambda_h(\mathbf{q}^{\text{home}} - \mathbf{q}) + \lambda_l\mathbf{d}_{\text{lim}}. \tag{29}$$

The reference $\mathbf{q}^{\text{ref}}$ is the warm-reference configuration when available, or otherwise the continuation seed. Attraction to this reference is proportional to the negative gradient of $\frac{1}{2}\|\mathbf{q} - \mathbf{q}^{\text{ref}}\|_2^2$. The remaining terms pull toward the home configuration $\mathbf{q}^{\text{home}}$ and follow the joint-limit avoidance direction $\mathbf{d}_{\text{lim}}$. The weights λ_r, λ_h, and λ_l control the respective contributions, with disabled terms set to zero. All three contributions pass through $\mathbf{N}_p$ in the total update.

The damped projection is not an exact null-space projector. The combined increment is clipped before updating the joint configuration, followed by joint-limit enforcement and bounds on displacement from the continuation seed. Accepted iterates are selected using task-priority criteria rather than a single aggregate loss. Continuity depends on reference attraction, solution reuse, step bounds, and trajectory-level validation.