NATIVE ACTION-PRIOR LEARNING FROM VIDEOS FOR WORLD ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

World action models integrate future visual dynamics with robot action prediction, but their scalability remains limited by the need for action-annotated robot trajectories. Observation-only videos contain rich evidence about interaction dynamics, but existing approaches typically use them either to pretrain visual representations that must later be adapted for control, or to infer latent actions that are subsequently grounded to robot commands. We present NAVA-WAM, which introduces *native action-prior learning* by directly pretraining the action policy from observation-only videos, avoiding indirect representation-to-control transfer or a separate latent-action model. Our training consists of two stages. First, we pretrain on observation-only videos, where future-video flow-matching supervision over visual transitions is propagated through transition-structured joint attention to optimize the Action-DiT and learn action-relevant priors. Second, we use action-labeled demonstrations to post-train the Action-DiT for robot control through joint video–action flow matching, while asymmetric attention decouples the visual branch from iterative action denoising and enables efficient action-only inference. Extensive experiments show that NAVA-WAM consistently outperforms prior approaches under both in-distribution and out-of-distribution settings, while demonstrating strong action-label efficiency and effective real-robot generalization. These results establish native action-prior learning as an effective approach to directly pretrain action policies from observation-only videos, providing a scalable path beyond action-labeled robot data.

1 INTRODUCTION

World action models (WAMs) (Zhu et al., 2025; Li et al., 2025; Ye et al., 2026c; Li et al., 2026a; Bi et al., 2026a) incorporate future visual dynamics into action learning, with strong potential to improve policy generalization. However, scaling WAMs remains challenging because they rely on action-annotated robot trajectories that are costly to collect and hard to standardize across embodiments. In contrast, large-scale observation-only videos provide abundant information about physical interactions and state transitions without action labels. The key challenge is therefore how to leverage such videos to learn priors that effectively improve action policy learning.

Existing approaches largely follow two paradigms for exploiting observation-only videos. *Vision representation learning* first learns visual or predictive representations from videos and then trains an action policy on top of them using action-labeled robot demonstrations (Nair et al., 2023; Hu et al., 2025; Assran et al., 2025; Sun et al., 2026). Because the learned video prior remains encoded in a vision-centric representation, its connection to executable actions is established only during downstream robot training, introducing an indirect representation-to-control transfer. *Latent-action learning* instead infers action-like variables from visual transitions and uses them as intermediate supervision for subsequent policy learning (Schmidt & Jiang, 2024; Ye et al., 2025; Chen et al., 2025b; Bu et al., 2025b; Bi et al., 2026a; Chen et al., 2026b). While more action-oriented, this paradigm requires an additional latent-action modeling stage and relies on complex architectural or optimization constraints (Yang et al., 2026b; Wang et al., 2026b; Zhang et al., 2026d) to avoid shortcut solutions in which the latent action simply copies information from the target observation. These constraints increase modeling complexity and can restrict the expressiveness of the latent-action space, thereby limiting the supervision provided to the policy and impairing its control learning.

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

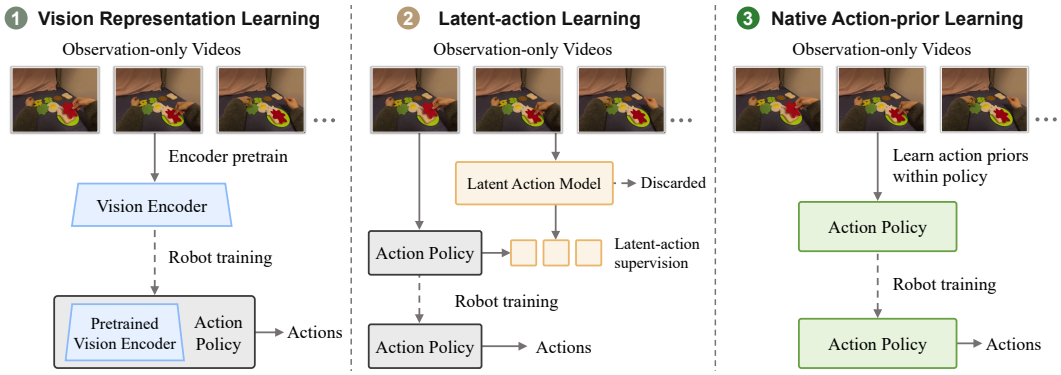

Figure 1: **Paradigms of leveraging observation-only videos for robot policy learning.** *(1) Vision representation learning* first learns visual representations from videos within a vision encoder and then trains an action policy on top, introducing an indirect representation-to-control transfer. *(2) Latent-action learning* extracts latent actions from visual transitions as intermediate supervision for policy learning, introducing an additional modeling stage and a potential representational bottleneck. *(3) Native action-prior learning* directly uses video dynamics to optimize the action policy, avoiding additional intermediate interfaces and representational bottlenecks.

These limitations motivate a simple question: *Can observation-only videos directly optimize an action policy without relying on an intermediate interface?* We answer this question with NAVA-WAM, which introduces *native action-prior learning* for world action models. As illustrated in Fig. 1, instead of first learning a separate visual representation or latent-action space and then using it in policy training, NAVA-WAM uses future-video prediction to directly pretrain the action policy. This creates a direct learning path from scalable observation-only videos to the action policy, avoiding intermediate representation constraints and enabling more effective policy learning.

Specifically, NAVA-WAM adopts a joint Mixture-of-Transformers (MoT) architecture (Liang et al., 2024), comprising a Video-DiT and an Action-DiT that interact through joint attention while retaining modality-specific parameters. During action-free pre-training, the Video-DiT is frozen and the Action-DiT is optimized solely through future-video flow-matching supervision. Our transition-structured joint attention allows the Action-DiT to model visual transitions while participating in future-state prediction, enabling the video objective to directly optimize the action policy. In this way, the Action-DiT learns action-relevant priors directly from visual transitions without requiring an intermediate interface. During downstream post-training, we jointly optimize the video and action streams on action-labeled robot demonstrations through video–action flow matching, adapting the pretrained Action-DiT to embodiment-specific control. We make cross-stream attention asymmetric to decouple the visual stream from the evolving action sample, allowing its representations to be cached and enabling efficient action-only inference without explicitly generating future videos.

We evaluate NAVA-WAM on the simulation benchmarks LIBERO/LIBERO-Plus (Liu et al., 2023; Fei et al., 2025) and RoboTwin 2.0 (Chen et al., 2025a) under both in-distribution and out-of-distribution settings, demonstrating superior performance across both regimes. We further deploy NAVA-WAM on a physical Franka FR3, achieving 93.3% success across tabletop manipulation tasks, compared with 66.7% for DreamZero (Ye et al., 2026c) and 53.3% for $\pi_{0.5}$ (Physical Intelligence et al., 2025). Controlled experiments show that native action-prior learning consistently outperforms representation-based and latent-action-based approaches across action-label budgets, with larger gains in low-budget regimes. Together, these results demonstrate that native action-prior learning can effectively leverage observation-only videos to learn action priors, improving generalization and action-label efficiency.

2 Related Work

2.1 World Action Models

Direct visuomotor policies learn observation-to-action mappings across tasks and embodiments (Brohan et al., 2023; Team et al., 2024; Liu et al., 2025), while vision–language–action (VLA)

models further leverage semantic priors from large-scale vision–language pre-training (Zitkovich et al., 2023; Kim et al., 2024; Black et al., 2024a; Physical Intelligence et al., 2025; Bjorck et al., 2025; Gemini Robotics Team et al., 2025; Team et al., 2026; Zhang et al., 2026a; Shin et al., 2026; Zhang et al., 2026f; Yang et al., 2026a). Their primary objective, however, remains conditional action prediction, leaving environment dynamics to be learned implicitly. World action models (WAMs) instead explicitly couple action learning with predictive modeling of future visual evolution. Existing WAMs can be broadly categorized into cascaded and joint architectures. *Cascaded WAMs* first extract predictive information from a video or world model and then use it to drive a separate action module (Du et al., 2023; Black et al., 2024b; Ko et al., 2024). Some methods explicitly generate future observations, visual subgoals, or motion cues before action prediction (Bharadhwaj et al., 2025; Zhou et al., 2024; Chen et al., 2026d; Li et al., 2026b; Lou et al., 2026; Wang et al., 2026a), while others condition policies on predictive intermediate representations without explicit pixel generation (Hu et al., 2025; Zhang et al., 2026e; Yan et al., 2026; Su et al., 2026; Luo et al., 2026b). *Joint WAMs* model future observations and actions within a shared architecture, enabling direct interaction between visual and action representations (Zhu et al., 2025; Li et al., 2025; Kim et al., 2026; Ye et al., 2026a; Huang et al., 2026b; Bi et al., 2026b). DreamZero and LingBot-VA (Ye et al., 2026c; Li et al., 2026a) jointly perform future visual prediction and action generation during closed-loop control, whereas Fast-WAM (Yuan et al., 2026) retains future-video prediction as training-time supervision while performing action-only inference. Dyna-2 (Dyna Robotics, 2026) further shows that adding human videos through co-training improves cross-embodiment generalization. NAVA-WAM follows the joint WAM paradigm with action-only inference, while enabling the action-generating expert to be directly pretrained from observation-only videos.

2.2 Learning from Observation-Only Videos

Observation-only videos provide scalable supervision for physical interactions and state transitions without action annotations. Existing robot learning approaches primarily leverage them through two paradigms. *Vision representation learning* learns visual or predictive representations from videos and uses them during downstream policy learning (Sun et al., 2026; Lin et al., 2026a; Feng et al., 2026; Nair et al., 2023; Goswami et al., 2025; Zhang et al., 2026b;c; Liang et al., 2026; Ren et al., 2026). VPP (Hu et al., 2025) conditions policies on predictive video features, and Masquerade (Lepert et al., 2025) learns visual representations through future robot-keypoint prediction. V-JEPA 2 and JEPA-VLA (Assran et al., 2025; Miao et al., 2026) use predictive video representations for planning and VLA learning. Such approaches retain video-derived knowledge in visual representations whose connection to executable actions is established only during downstream policy training, resulting in an indirect representation-to-control transfer. *Latent-action learning* instead infers action-like variables from visual transitions as intermediate supervision for policy learning (Bruce et al., 2024; Chen et al., 2025b; Nikulin et al., 2025; Bu et al., 2025b; Tharwat et al., 2025; Chen et al., 2026b; Liu et al., 2026; Lin et al., 2026b; Wan et al., 2026; Lee et al., 2026; Luo et al., 2026a). LAPO (Schmidt & Jiang, 2024) learns latent actions with a latent-to-action decoder, while LAPA (Ye et al., 2025) pretrains a VLA through discrete latent-action prediction before grounding them to robot actions. Motus (Bi et al., 2026a) incorporates optical-flow motion cues, while RepWAM and LingBot-VA 2.0 (Wang et al., 2026b; Zhang et al., 2026d) learn transition-oriented action representations in semantic visual spaces. However, these methods introduce an additional latent-action modeling stage and require architectural or optimization constraints to prevent shortcut solutions (Gao et al., 2026; Huang et al., 2026a; Yang et al., 2026b; Wang et al., 2026b; Zhang et al., 2026d; Chen et al., 2026a), which can constrain the learned latent-action space and the supervision it provides to the policy. In contrast, NAVA-WAM introduces *native action-prior learning* to learn transition priors directly within the action policy, providing a direct path from scalable observation-only videos to action policy pre-training.

3 Method

We present NAVA-WAM, a world action model that learns action-relevant priors directly within the robot policy from observation-only videos, without relying on an intermediate interface. Our method is illustrated in Fig. 2. Sec. 3.1 introduces the problem formulation and model architecture. Sec. 3.2 presents native action-prior pre-training from observation-only videos, and Sec. 3.3 describes downstream post-training and inference.

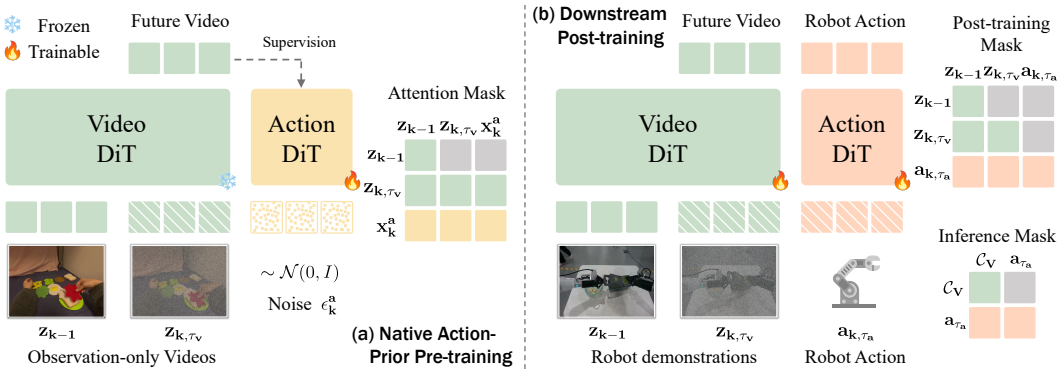

Figure 2: **Overview of NAVA-WAM.** During native action-prior pre-training, the Action-DiT is optimized solely through future-video supervision, learning action-relevant priors directly from observation-only videos. During post-training, the Video-DiT and Action-DiT are jointly optimized on action-labeled robot demonstrations using video and action flow-matching objectives, adapting the pretrained Action-DiT to continuous robot control. At inference, NAVA-WAM denoises only robot actions without explicitly generating future videos, enabling efficient closed-loop control.

3.1 Preliminaries

Problem formulation. Let o_t denote the visual observation at time t, l the language instruction, and q_t the current proprioceptive state. A robot policy $\mathcal{P}_\theta$ predicts a continuous action chunk $a_{t:t+H-1} \in \mathbb{R}^{H \times d_a}$ over a horizon of H steps according to

$$\mathcal{P}_\theta \left(a_{t:t+H-1} \mid o_t, l, q_t \right). \tag{1}$$

We consider two sources of training data. The observation-only dataset $\mathcal{D}_v = \{(o_{0:T}, l)\}$ contains videos and language descriptions without action annotations. The action-labeled robot dataset $\mathcal{D}_r = \{(o_{0:T}, l, q, a_{0:T-1})\}$ additionally provides proprioceptive states q and continuous robot actions $a_{0:T-1}$. A frozen spatiotemporal VAE E (Wan et al., 2025) encodes each video clip into latent visual states $z_{0:K} = E(o_{0:T})$, where each adjacent pair (z_{k-1}, z_k) defines a visual state transition indexed by k. Our goal is to leverage the abundant action-free transitions in $\mathcal{D}_v$ to pretrain the action policy before adapting it to embodiment-specific control using $\mathcal{D}_r$.

World action model architecture. We instantiate $\mathcal{P}_\theta$ as a Mixture-of-Transformers (MoT) (Liang et al., 2024) comprising a pretrained Video-DiT with parameters θ_V and an Action-DiT with parameters θ_A. The Video-DiT models visual dynamics, while the Action-DiT generates continuous robot actions. The two experts retain modality-specific parameters while exchanging information through joint attention. Each expert independently constructs its queries, keys, and values. Queries from modality $m \in \{V, A\}$ attend jointly to the keys and values from both streams:

$$H_m = \mathrm{Attn}(Q_m, [K_V; K_A], [V_V; V_A]). \tag{2}$$

Together, the two experts form a unified architecture for modeling visual dynamics and robot actions.

3.2 Native Action-Prior Pre-training

Observation-only videos capture visual state transitions that reveal the consequences of physical interactions, providing a rich and scalable source for learning action-relevant priors. Our goal is to improve WAM training by introducing a pre-training stage that learns action priors from such videos, while keeping the model architecture consistent with downstream WAM training and making only minimal changes. To this end, we propose *native action-prior pre-training*, which directly exploits transitions in $\mathcal{D}_v$ to pretrain the action policy without introducing an intermediate interface.

Concretely, we retain the same MoT architecture used for downstream WAM training. During pre-training, action tokens are not supervised by robot actions; instead, we structure joint attention around visual transitions and optimize the Action-DiT solely through future-video supervision. This enables the action policy itself to acquire action priors directly from observation-only videos.

Inverse-forward dynamics modeling with attention. We motivate action-prior learning through an inverse–forward dynamics factorization (Schmidt & Jiang, 2024; Ye et al., 2025). Given two adjacent states z_{k-1} and z_k, inverse dynamics extracts a transition-dependent variable r_k, while forward dynamics uses r_k together with the preceding state to predict the future state:

$$r_k = q(z_{k-1}, z_k), \qquad \hat{z}_k = f(z_{k-1}, r_k). \tag{3}$$

Rather than introducing separate inverse and forward models (Cui et al., 2024; Wang et al., 2026b; Zhang et al., 2026d; Yang et al., 2026b), we realize an analogous information flow implicitly within the policy through *transition-structured joint attention*.

For each transition $z_{k-1} \to z_k$, we construct a segment containing the clean preceding state z_{k-1}, a noised version of the future state z_k, denoted z_{k,τ_v} at timestep τ_v, and the corresponding group of Action-DiT tokens x^a_k, as illustrated in Fig. 2. Transition segments are isolated by the attention mask (detailed in Appendix B.3). Within each segment, clean preceding-state queries attend only to the clean state, whereas noised-future and action queries interact with all three token groups.

Under this transition-structured mask, the Action-DiT attention is given by

$$H_{A,k} = \mathrm{Attn}(Q_{A,k}, [K_{V,k-1}; K_{V,k,\tau_v}; K_{A,k}], [V_{V,k-1}; V_{V,k,\tau_v}; V_{A,k}]) \,. \tag{4}$$

The Action-DiT jointly accesses the preceding and future visual states and forms an internal representation of their transition. This information flow is analogous to inverse dynamics in Eq. 3, while keeping the transition representation implicit rather than materializing explicit latent actions.

Conversely, the noised future-state tokens attend to both visual states and the corresponding Action-DiT representations:

$$H_{V,k,\tau_v} = \mathrm{Attn}(Q_{V,k,\tau_v}, [K_{V,k-1}; K_{V,k,\tau_v}; K_{A,k}], [V_{V,k-1}; V_{V,k,\tau_v}; V_{A,k}]) \,. \tag{5}$$

This interaction parallels forward dynamics in Eq. 3, where future-state prediction depends on both the preceding state and a transition representation. Here, the Action-DiT accesses the future state only through its noised version z_{k,τ_v}, precluding the trivial shortcut of directly copying the clean prediction target z_k (Garrido et al., 2026; Ye et al., 2025). See Appendix A for formal analysis. Together, Eqs. 4 and 5 realize an implicit inverse–forward interaction within the joint transformer, with the Action-DiT modeling the transition between adjacent visual states.

Pre-training formulation. We use standard flow matching for future-video prediction (Lipman et al., 2023). For each target future state z_k, we sample $\epsilon^v_k \sim \mathcal{N}(0, I)$ and a flow timestep $\tau_v \in [0, 1]$, and construct

$$z_{k,\tau_v} = (1 - \tau_v) z_k + \tau_v \epsilon^v_k, \qquad u^v_k = \epsilon^v_k - z_k, \tag{6}$$

where $\tau_v = 0$ and $\tau_v = 1$ correspond to the data and noise endpoints, respectively, and u^v_k is the target velocity. For the corresponding Action-DiT input, we independently sample $\epsilon^a_k \sim \mathcal{N}(0, I)$ with the action timestep at $\tau_a = 1$.

For each transition, the Video-DiT receives $[z_{k-1}, z_{k,\tau_v}]$, while the Action-DiT receives ϵ^a_k. We freeze the pretrained Video-DiT, $\theta_V = \bar{\theta}_V$, and optimize only the Action-DiT parameters θ_A. With stream-wise timesteps $\boldsymbol{\tau} = (\tau_v, 1)$, the pre-training objective is

$$\mathcal{L}_{\mathrm{pre}} = \mathbb{E}\left[\sum_{k=1}^{K} w(\tau) \left\| v^v_{\theta_V, \theta_A}([z_{k-1}, z_{k,\tau_v}], \epsilon^a_k, \boldsymbol{\tau} \mid l) - u^v_k \right\|_2^2\right]. \tag{7}$$

Through transition-structured joint attention, future-video prediction depends on the Action-DiT representations. Because the Video-DiT is frozen, minimizing the future-video objective directly updates θ_A through the joint-attention pathway. This direct optimization of the Action-DiT from visual transitions constitutes our native action-prior learning, allowing action-relevant priors to emerge without a separate transfer interface.

3.3 Post-training and Inference

Action-labeled robot demonstrations provide direct supervision for embodiment-specific continuous control. We post-train on $\mathcal{D}_r$ to ground the action priors learned from observation-only videos to executable robot actions. We retain the transition-based structure from pre-training while adapting the cross-stream interaction to direct action supervision and efficient inference.

Asymmetric transition attention. During native action-prior pre-training, future-video tokens attend to Action-DiT representations through Eq. 5, allowing the video objective to propagate supervision into the Action-DiT. During post-training, action labels become available and the Action-DiT receives direct supervision. We therefore make the visual stream independent of the action stream while retaining visual conditioning for the Action-DiT. This asymmetric interaction enables the visual representations to be cached and reused throughout action denoising at inference.

Specifically, as shown in Fig. 2, Video-DiT queries attend only to the visual stream:

$$H_V = \mathrm{Attn}(Q_V, K_V, V_V)\,, \tag{8}$$

while Action-DiT queries continue to attend to both visual and action streams:

$$H_A = \mathrm{Attn}(Q_A, [K_V; K_A], [V_V; V_A])\,. \tag{9}$$

Post-training formulation. We jointly train the visual and action streams using standard flow matching. For each transition $z_{k-1} \to z_k$ and its corresponding continuous action chunk a_k, we independently sample $\epsilon_k^v, \epsilon_k^a \sim \mathcal{N}(0, I)$ and flow timesteps $\tau_v, \tau_a \in [0, 1]$, and construct

$$\begin{aligned} z_{k,\tau_v} &= (1-\tau_v)z_k + \tau_v \epsilon_k^v, & u_k^v &= \epsilon_k^v - z_k, \\ a_{k,\tau_a} &= (1-\tau_a)a_k + \tau_a \epsilon_k^a, & u_k^a &= \epsilon_k^a - a_k. \end{aligned} \tag{10}$$

With $\boldsymbol{\tau} = (\tau_v, \tau_a)$, the post-training objective is $\mathcal{L}_{\mathrm{post}} = \lambda_v \mathcal{L}_{\mathrm{FM}}^v + \lambda_a \mathcal{L}_{\mathrm{FM}}^a$, where

$$\mathcal{L}_{\mathrm{FM}}^m = \mathbb{E}\left[\sum_{k=1}^{K} w_m(\tau_m) \left\| v_{\theta_V,\theta_A}^m([z_{k-1}, z_{k,\tau_v}], a_{k,\tau_a}, \boldsymbol{\tau} \mid l, q) - u_k^m \right\|_2^2\right], \qquad m \in \{v, a\}. \tag{11}$$

Here, λ_v and λ_a weight the visual and action objectives, respectively. The video objective adapts the visual expert to downstream robot observations, while the action objective grounds the pretrained Action-DiT to embodiment-specific continuous control.

Inference. Asymmetric attention makes the Video-DiT independent of the evolving action sample. We initialize the future-video and action streams with Gaussian noise and evaluate them at the first denoising step. During this forward pass, we cache the Video-DiT keys and values at each layer ℓ:

$$\mathcal{C}_V = \{K_\ell^V, V_\ell^V\}_{\ell=1}^L. \tag{12}$$

Subsequent denoising steps integrate only the action flow while reusing the visual cache $\mathcal{C}_V$:

$$\frac{da_{\tau_a}}{d\tau_a} = v_{\theta_A}^a(a_{\tau_a}, \tau_a \mid \mathcal{C}_V, l, q)\,, \qquad \tau_a : 1 \to 0. \tag{13}$$

Thus, after the initial joint forward pass, iterative denoising runs only through the Action-DiT, enabling efficient closed-loop control without explicitly generating future videos.

4 Experiments

We evaluate NAVA-WAM on standard simulation benchmarks under both *in-domain* (ID) and *out-of-domain* (OOD) settings, and further validate its deployment on a physical robot.

4.1 Experimental Setup

NAVA-WAM initializes the Video-DiT from Wan2.2-5B (Wan et al., 2025) and uses a 1B-parameter Action-DiT with hidden dimension 1,024 (Li et al., 2026a; Yuan et al., 2026). Native action-prior pre-training uses observation-only videos from Open X-Embodiment (O'Neill et al., 2024), AgiBot-World (Bu et al., 2025a), and EgoDex (Hoque et al., 2026). For downstream adaptation, we optimize the video and action streams with $\lambda_v = \lambda_a = 1$. More details are provided in Appendix B.

Simulation evaluation. We evaluate on two simulation benchmark families. *LIBERO and LIBERO-Plus.* LIBERO (Liu et al., 2023) contains four 10-task suites, while LIBERO-Plus (Fei et al., 2025) extends them with seven distribution shifts. Following Zhang et al. (2026e) and Yuan et al. (2026), we train on 50 LIBERO demonstrations per task and evaluate on LIBERO (ID) and LIBERO-Plus

Table 1: **Results on LIBERO/LIBERO-Plus (*left*) and RoboTwin 2.0 (*right*).** We report average success rate (%). LIBERO and RoboTwin Clean measure in-domain performance, while LIBERO-Plus and RoboTwin Random evaluate out-of-domain generalization. Best results are shown in **bold**.

Method	LIBERO	LIBERO-Plus
Direct action policies		
π_0 (2024a)	94.1	53.6
π_0-FAST (2025)	85.5	61.6
StarVLA-α (2026b)	96.5	77.0
$\pi_{0.5}$ (2025)	96.9	77.4
ABot-M0 (2026c)	98.6	80.5
World action models		
JEPA-VLA (2026)	96.4	25.6
Fast-WAM (2026)	97.6	51.5
Image-WAM (2026e)	98.4	83.1
Being-H0.7 (2026b)	**99.2**	82.1
NAVA-WAM	99.0	**83.5**

Method	Clean	Random
Direct action policies		
DP (2025)	28.0	0.6
RDT (2025)	34.5	13.7
π_0 (2024a)	46.4	16.3
UP-VLA (2025)	52.9	15.2
World action models		
Fast-WAM (2026)	71.9	6.3
BagelVLA (2026)	75.3	20.5
HALO (2026)	80.5	26.4
Image-WAM (2026e)	85.0	37.6
MV-WAM (2026c)	84.0	55.7
NAVA-WAM	**88.5**	**73.6**

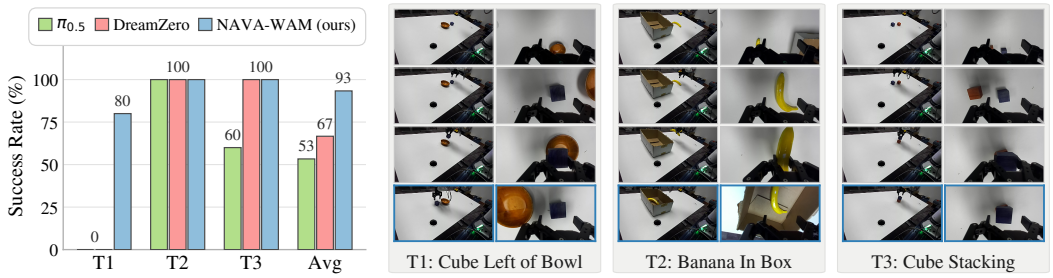

Figure 3: **Real-robot evaluation on a Franka FR3.** *Left*: success rate over five trials per task and average success over all 15 trials. *Right*: NAVA-WAM rollouts for the three evaluation tasks, shown from third-person and wrist-camera views. NAVA-WAM achieves 93.3% average success and correctly follows the spatial relation in T1, where both baselines fail across all trials.

(OOD). *RoboTwin 2.0.* We train exclusively on 50 Clean-domain demonstrations per task from RoboTwin 2.0 (Chen et al., 2025a) and use no Random-domain demonstrations. Following Chen et al. (2026c), we evaluate in both Clean and Random domains, treating Random as the OOD setting.

Real-robot evaluation. We additionally deploy NAVA-WAM post-trained on DROID (Khazatsky et al., 2024) on a physical Franka FR3 arm *without task-specific fine-tuning*. We evaluate three table-top manipulation tasks with five trials each and compare against $\pi_{0.5}$ (Physical Intelligence et al., 2025) and the DreamZero-DROID policy (Ye et al., 2026c). Additional details are in Appendix C.5.

4.2 Main Results

Simulation results. Tab. 1 shows that NAVA-WAM achieves 99.0% success on LIBERO and 83.5% on LIBERO-Plus, maintaining strong performance under substantial distribution shifts. On RoboTwin 2.0, NAVA-WAM reaches 88.5% on Clean and 73.6% on Random. Compared with the strongest baseline, NAVA-WAM improves success by 3.5 points on Clean and 17.9 points on Random, while exhibiting a much smaller Clean-to-Random performance drop. Together, these results show that native action-prior learning preserves strong in-domain control while substantially improving generalization under visual and environmental distribution shifts.

Real-robot results. As shown in Fig. 3, NAVA-WAM succeeds in 14/15 trials (93.3%), compared with 10/15 (66.7%) for DreamZero and 8/15 (53.3%) for $\pi_{0.5}$. The largest difference occurs on T1, which requires placing a cube on the *left side* of a bowl: NAVA-WAM succeeds in 4/5 trials, whereas both baselines fail in all five trials. These results complement the simulation benchmarks and demonstrate that the learned action prior transfers effectively to physical robot deployment.

Table 2: **Ablation studies on RoboTwin 2.0.** (a) Comparison of observation-only pre-training methods under varying action-label budgets. (b) Action relevance of learned representations, measured by ridge-regression R^2 to ground-truth actions under different probe-data budgets. (c) Inference compute per action chunk and Random-domain performance. Best results are shown in **bold**.

(a) Pre-training & label efficiency

	Demos / task		
Pre-training method	10	25	50
No pre-training	60.5	67.6	69.7
Representation-based	58.2	60.6	68.1
Latent-action-based	61.6	65.0	71.2
Ours	**66.5**	**72.0**	**73.6**

(b) Action relevance

	Probe data		
Feature source	10%	50%	100%
Representation-based	0.080	0.157	0.153
Latent-action (CoMo)	0.088	0.192	0.203
Latent-action (DynaMo)	0.052	0.143	0.154
Ours	**0.247**	**0.320**	**0.329**

(c) Inference efficiency

Method	TFLOPs	Random
Fast-WAM	**3.56**	6.3
HALO	67.69	26.4
Image-WAM	4.09	37.6
Ours	4.30	**73.6**

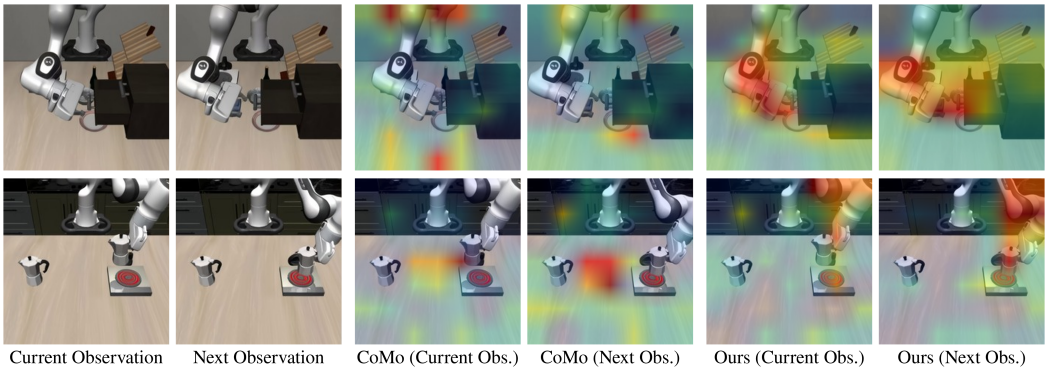

Figure 4: **Attention of learned action representations.** We visualize attention from action representations to visual tokens on held-out data, overlaid on the *current* and *next* input observations. Our pretrained Action-DiT attends to interaction-relevant regions and aligns with observed motion, while the CoMo latent-action baseline shows more diffuse attention over static regions.

4.3 Ablation Studies

We conduct ablations on RoboTwin 2.0 to analyze native action-prior learning and its impact on downstream control. Unless otherwise specified, all variants follow the same training and evaluation protocol and differ only in the component under study.

Observation-only pre-training method and action-label efficiency. We compare different methods for leveraging observation-only videos under the same downstream protocol. As shown in Tab. 2a, NAVA-WAM consistently outperforms representation-based and latent-action-based pre-training (Yang et al., 2026b) across all action-label budgets. With 25 demonstrations per task, it achieves 72.0% success versus 60.6% and 65.0%, respectively. Moreover, NAVA-WAM with only 25 demonstrations surpasses all baselines trained with 50, demonstrating both more effective observation-only pre-training and improved action-label efficiency.

Observation-only data scale. We study how scaling observation-only pre-training data affects downstream generalization. As shown in Fig. 5, Random-domain success consistently improves with corpus size under both action-label budgets, demonstrating that native action-prior learning benefits from more observation-only data.

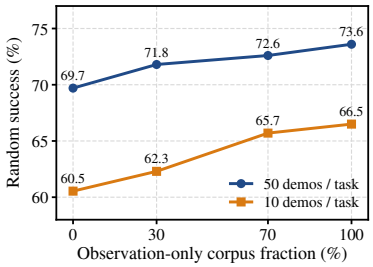

Figure 5: **Scaling observation-only pre-training data.** Success across data scales at different action-label budgets.

Action relevance of learned representations. We probe the action relevance of representations learned from observation-only videos: Video-DiT features from vision representation learning, inferred latent actions from latent-action learning (Cui et al., 2024; Yang et al., 2026b), and Action-DiT features from native action-prior learning.

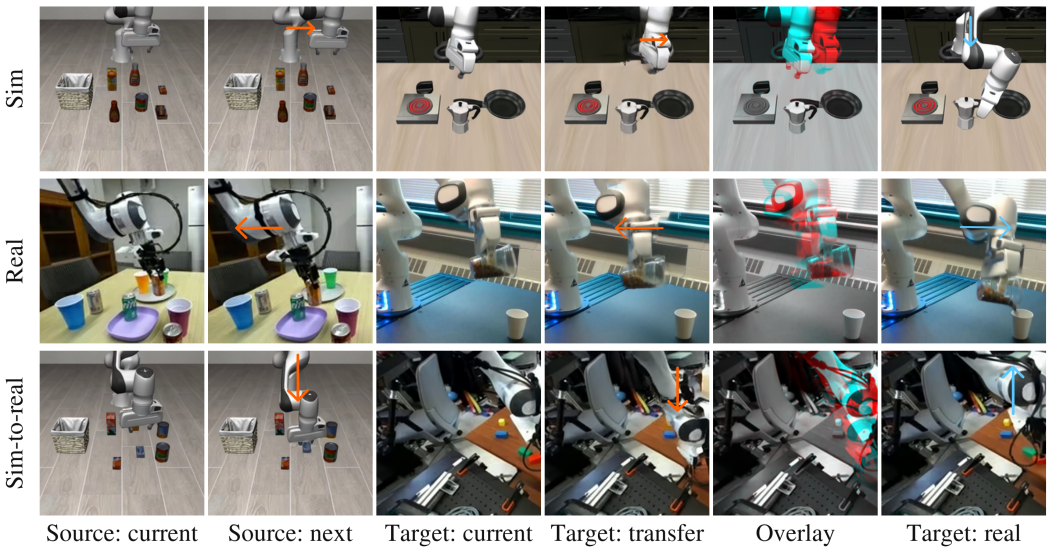

Figure 6: **Qualitative visualization of motion transfer across domains.** Each example transfers the transition encoded by a source pair (*Source: current* $\rightarrow$ *Source: next*) to a different target observation. The source transition specifies the underlying motion cue (orange arrows), which is applied to *Target: current* to obtain *Target: transfer*. The *Overlay* visualizes the transferred state relative to the current target observation, while *Target: real* shows the corresponding real transition (blue arrows). Across simulation, real-world, and sim-to-real examples, the transferred states follow the source motion despite substantial changes in visual appearance and scene configuration.

In Tab. 2b, NAVA-WAM achieves the highest R^2 across all probe-data budgets. Fig. 4 further provides complementary qualitative evidence: our Action-DiT focuses on interaction-relevant regions such as the gripper and manipulated objects, whereas the latent-action baseline (Yang et al., 2026b) exhibits more diffuse attention over static regions. Together, these results indicate that native action-prior learning encodes transition-relevant action information directly within the Action-DiT.

Qualitative motion transfer. Fig. 6 analyzes the action prior learned by our pretrained Action-DiT. Given a source transition, we extract its Action-DiT activations and transfer them to a different target observation to predict the corresponding next state (Garrido et al., 2026; Todd et al., 2024). Across simulation, real-world, and sim-to-real examples, the transferred states reproduce the source motion despite substantial changes in appearance and scene configuration. These results show that native action-prior learning captures transferable action priors beyond the specific visual content of the source transition.

Inference efficiency. Tab. 2c compares the inference compute across methods. NAVA-WAM requires 4.30 TFLOPs, comparable to Image-WAM (4.09 TFLOPs) and Fast-WAM (3.56 TFLOPs), while achieving substantially higher Random-domain success at 73.6%. This demonstrates that our model achieves strong out-of-domain generalization with low inference overhead.

5 Conclusion

We present NAVA-WAM, which introduces *native action-prior learning* for world action models, directly pre-training the action policy from observation-only videos without relying on an intermediate interface. By learning action-relevant priors from visual transitions and adapting them to robot control with action-labeled demonstrations, NAVA-WAM unifies action-free pre-training and downstream policy learning within a single model. Experiments demonstrate strong generalization in simulation and real-robot deployment, with efficient action-only inference. Overall, our results suggest that directly learning action priors from scalable observation-only videos offers a simple and effective path toward scaling world action models beyond costly action-labeled robot data.

AI USE STATEMENT

In this work, we used generative AI tools to assist with the writing of proofs, provide feedback on research methodology and experimental design, support method implementation, assist with qualitative and thematic data analysis, and help interpret experimental results. We did not use generative AI tools to formulate mathematical claims, generate synthetic datasets, develop theoretical models or conceptual frameworks, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, clean or reformat datasets, or assist with translation. All AI-assisted work was reviewed and verified by the authors. AI-suggested experimental designs were assessed by the authors before being carried out, AI-assisted implementations were tested for correctness and verified against the reported results, and AI-assisted proofs, analyses, and interpretations were checked against the underlying methodology, data, and experimental results. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The method formulation and training objectives are specified in Sec. 3. Appendix A provides the theoretical analysis and proofs supporting the native action-prior learning formulation. Appendix B details the observation-only pre-training data and optimization setup, downstream training hyperparameters, transition-structured attention masks, benchmark and evaluation protocols, and action-only inference procedure. Appendix C provides additional experimental results, qualitative analyses, detailed benchmark breakdowns, and the real-robot evaluation and deployment setup. Together, these details specify the model, training, inference, and evaluation procedures needed to reproduce the reported experiments.

ETHICS STATEMENT

This work studies robot learning from observation-only videos and action-labeled robot demonstrations, with evaluation in simulation and on a physical manipulation platform. We use existing datasets and do not collect new human-subject data as part of this study. As with other learned robotic control systems, deployment outside the evaluated conditions may lead to unintended or unsafe physical behavior. Such systems should therefore be validated under their intended operating conditions and deployed with appropriate physical safeguards and human oversight.

REFERENCES

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2Act: Human video generation in novel scenarios enables generalizable robot manipulation. In *Conference on Robot Learning*, 2025.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35101–35113, 2026a.

Hongzhe Bi, Zihao Zhou, Yihang Tang, Jingrui Pang, Shuhe Huang, Haitian Liu, Runqing Wang, Shuai Huang, Yichen Wang, Yiming Cheng, et al. Motus2: A self-evolving general world model for dexterous manipulation. *arXiv preprint arXiv:2608.30237*, 2026b.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

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

Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pre-trained image-editing diffusion models. In *International Conference on Learning Representations*, volume 2024, pp. 33431–33452, 2024b.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. RT-1: Robotics transformer for real-world control at scale. In *Robotics: Science and Systems*, 2023.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *International conference on machine learning*, 2024.

Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025a.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. UniVLA: Learning to act anywhere with task-centric latent actions. In *Robotics: Science and Systems*, 2025b.

Guangyan Chen, Qi Shao, Te Cui, Zichen Zhou, Weixin Mao, Luojie Yang, Meiling Wang, Yi Yang, Hua Chen, and Yufeng Yue. Learning a unified latent action space from videos with action-centric cycle consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12871–12880, 2026a.

Jialei Chen, Kai Wang, Kang Chen, Shuaihang Chen, Feng Gao, Wenhao Tang, Zhiyuan Li, Weilin Liu, Zhuyu Yao, Boxun Li, et al. Lawam: Latent world action models for efficient dynamics-aware robot policies. *arXiv preprint arXiv:2606.15768*, 2026b.

Jintao Chen, Peidong Jia, Qingpo Wuwu, Jiaming Liu, Mengfei Du, Chun-Kai Fan, Xiaowei Chi, Hao Chen, Chengyu Bai, Zezhong Qian, et al. MV-WAM: Manifold-aware world action model with value augmentation. *arXiv preprint arXiv:2606.21088*, 2026c.

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

Yi Chen, Yuying Ge, Yizhuo Li, Yixiao Ge, Mingyu Ding, Ying Shan, and Xihui Liu. Moto: Latent motion token as the bridging language for robot manipulation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2025b.

Yixiang Chen, Peiyan Li, Yuan Xu, Qisen Ma, Jiabing Yang, Kai Wang, Jianhua Yang, Dong An, He Guan, Gaoteng Liu, et al. Flowwam: Optical flow as a unified action representation for world action models. *arXiv preprint arXiv:2607.13017*, 2026d.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Zichen J Cui, Hengkai Pan, Aadhithya Iyer, Siddhant Haldar, and Lerrel Pinto. DynaMo: In-domain dynamics pretraining for visuo-motor control. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in neural information processing systems*, 36:9156–9172, 2023.

Dyna Robotics. Dyna-2: A 1-million-hour scaling law for world-action models. https://www.dyna.co/dyna-2, August 2026.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. LIBERO-Plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

Yicheng Feng, Wanpeng Zhang, Ye Wang, Hao Luo, Haoqi Yuan, Sipeng Zheng, and Zongqing Lu. Spatial-aware vla pretraining through visual-physical alignment from human videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 712–723, 2026.

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

Quentin Garrido, Tushar Nagarajan, Basile Terver, Nicolas Ballas, Yann LeCun, and Michael Rabbat. Learning latent action world models in the wild. In *International Conference on Machine Learning*, 2026.

Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini robotics: Bringing AI into the physical world. *arXiv preprint arXiv:2503.20020*, 2025.

Raktim Gautam Goswami, Amir Bar, David Fan, Tsung-Yen Yang, Gaoyue Zhou, Prashanth Krishnamurthy, Michael Rabbat, Farshad Khorrami, and Yann LeCun. World models for learning dexterous hand-object interactions from human videos. *arXiv preprint arXiv:2512.13644*, 2025.

Ryan Hoque, Peide Huang, David Yoon, Jian Zhang, et al. Egodex: Learning dexterous manipulation from large-scale egocentric video. In *International Conference on Learning Representations*, volume 2026, pp. 4218–4237, 2026.

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 *International Conference on Machine Learning*, 2025.

Yucheng Hu, Jianke Zhang, Yuanfei Luo, Yanjiang Guo, Xiaoyu Chen, Xinshu Sun, Kun Feng, Qingzhou Lu, Sheng Chen, Yangang Zhang, et al. Bagelvla: Enhancing long-horizon manipulation via interleaved vision-language-action generation. *arXiv preprint arXiv:2602.09849*, 2026.

Huang Huang, Sriram Yenamandra, Arjun Majumdar, Elie Aljalbout, Tushar Nagarajan, Tsung-Yen Yang, Akshara Rai, Michael Rabbat, Li Fei-Fei, Jiajun Wu, et al. Cross-embodiment robot foundation world models with latent actions. In *International Conference on Machine Learning*, 2026a.

Ze Huang, Jiahui Zhang, Hairuo Liu, Chenxi Zhang, Ran Cheng, and Li Zhang. Learning transferable dynamics priors from action to world modeling. In *European Conference on Computer Vision*, pp. 430–449. Springer, 2026b.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

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

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

Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B Tenenbaum. Learning to act from actionless videos through dense correspondences. In *International Conference on Learning Representations*, volume 2024, pp. 40938–40958, 2024.

Jung Min Lee, Dohyeok Lee, Seokhun Ju, Taehyun Cho, Jin Woo Koo, Li Zhao, Sangwoo Hong, and Jungwoo Lee. Mvp-lam: Learning action-centric latent action via cross-viewpoint reconstruction. In *International Conference on Machine Learning*, 2026.

Marion Lepert, Jiaying Fang, and Jeannette Bohg. Masquerade: Learning from in-the-wild human videos using data-editing. *arXiv preprint arXiv:2508.09976*, 2025.

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

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. In *Robotics: Science and Systems*, 2025.

Zhiyi Li, Peilin Wu, Xiaoshen Han, Ruojin Cai, and Yilun Du. Structured 4d latent predictive model for robot planning. In *International Conference on Machine Learning*, 2026b.

Qiwei Liang, Boyang Cai, Minghao Lai, Sitong Zhuang, Tao Lin, Yan Qin, Yixuan Ye, Jiaming Liang, and Renjing Xu. Bootstrap dynamic-aware 3d visual representation for scalable robot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13419–13429, 2026.

Weixin Liang, Lili Yu, Liang Luo, Srinivasan Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen-tau Yih, Luke Zettlemoyer, et al. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models. *arXiv preprint arXiv:2411.04996*, 2024.

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*, 2026a.

Yihan Lin, Haoyang Li, Yang Li, Haitao Shen, Yihan Zhao, Chao Shao, and Jing Zhang. From pixels to tokens: A systematic study of latent action supervision for vision-language-action models. In *International Conference on Machine Learning*, 2026b.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Mengya Liu, Baoxiong Jia, Jiangyong Huang, Jingze Zhang, and Siyuan Huang. Lara: Latent action representation alignment for vision-language-action models. In *International Conference on Machine Learning*, 2026.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1B: a diffusion foundation model for bimanual manipulation. In *International Conference on Learning Representations*, volume 2025, pp. 29982–30009, 2025.

Yunfan Lou, Xiaowei Chi, Xiaojie Zhang, Zezhong Qian, Chengxuan Li, Rongyu Zhang, Yaoxu Lyu, Guoyu Song, Chuyao Fu, Haoxuan Xu, et al. Mask world model: Predicting what matters for robust robot policy learning. In *International Conference on Machine Learning*, 2026.

Hao Luo, Ye Wang, Wanpeng Zhang, Haoqi Yuan, Yicheng Feng, Haiweng Xu, Sipeng Zheng, and Zongqing Lu. Joint-aligned latent action: Towards scalable vla pretraining in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026a.

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*, 2026b.

Shangchen Miao, Ningya Feng, Jialong Wu, Ye Lin, Xu He, Dong Li, and Mingsheng Long. JEPA-VLA: Video predictive embedding is needed for VLA models. *arXiv preprint arXiv:2602.11832*, 2026.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A universal visual representation for robot manipulation. In *Conference on Robot Learning*, volume 205, pp. 892–909. PMLR, 2023.

Alexander Nikulin, Ilya Zisman, Denis Tarasov, Nikita Lyubaykin, Andrei Polubarov, Igor Kiselev, and Vladislav Kurenkov. Latent action learning requires supervision in the presence of distractors. In *International Conference on Machine Learning*, 2025.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and RT-X models. In *IEEE International Conference on Robotics and Automation*, pp. 6892–6903, 2024.

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. *arXiv preprint arXiv:2501.09747*, 2025.

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

Zhongwei Ren, Yunchao Wei, Xiao Yu, Guixun Luo, Yao Zhao, Bingyi Kang, Jiashi Feng, and Xiaojie Jin. Videoworld 2: Learning transferable knowledge from real-world videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Dominik Schmidt and Minqi Jiang. Learning to act without actions. In *International Conference on Learning Representations*, volume 2024, pp. 9379–9395, 2024.

Jinwoo Shin, John Won, Kyungmin Lee, Huiwon Jang, and Dongyoung Kim. Dual-stream diffusion for world-model augmented vision-language-action model. In *International Conference on Machine Learning*, 2026.

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

Yue Su, Sijin Chen, Haixin Shi, Mingyu Liu, Zhengshen Zhang, Ningyuan Huang, Weiheng Zhong, Zhengbang Zhu, Yuxiao Liu, and Xihui Liu. World guidance: World modeling in condition space for action generation. In *International Conference on Machine Learning*, 2026.

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*, pp. 478–497. Springer, 2026.

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

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.

Bahey Tharwat, Yara Nasser, Ali Abouzeid, and Ian Reid. Latent action pretraining through world modeling. *arXiv preprint arXiv:2509.18428*, 2025.

Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron Wallace, and David Bau. Function vectors in large language models. In *International conference on learning representations*, volume 2024, pp. 17282–17333, 2024.

Shenghua Wan, Xiaohai Hu, Xunlan Zhou, Lei Yuan, Le Gan, and De-Chuan Zhan. Multi-view consistent latent action learning for world modeling and control. In *International Conference on Machine Learning*, 2026.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Jiaxu Wang, Yicheng Jiang, Tianlun He, Jingkai Sun, Qiang Zhang, Junhao He, Jiahang Cao, Zesen Gan, Mingyuan Sun, Qiming Shao, et al. Mvista-4d: View-consistent 4d world model with test-time action inference for robotic manipulation. In *International Conference on Machine Learning*, 2026a.

Junke Wang, Qihang Zhang, Shuai Yang, Yiming Luo, Yujun Shen, Zuxuan Wu, Yu-Gang Jiang, and Yinghao Xu. Repwam: World action modeling with representation visual-action tokenizers. *arXiv preprint arXiv:2606.13674*, 2026b.

Zhuoyuan Wu and Jun Gao. OSCAR: Omni-embodiment action-conditioned world model for robotics. *arXiv preprint arXiv:2606.04463*, 2026.

Haodong Yan, Zhide Zhong, Jiaguan Zhu, Junjie He, Weilin Yuan, Wenxuan Song, Xin Gong, Yingjie Cai, Guanyi Zhao, Xu Yan, et al. S-vam: Shortcut video-action model by self-distilling geometric and semantic foresight. In *European Conference on Computer Vision*, 2026.

Fuxiang Yang, Donglin Di, Lulu Tang, Xuancheng Zhang, Lei Fan, Hao Li, Wei Chen, Tonghua Su, and Baorui Ma. Chain of world: World model thinking in latent motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6675–6684, 2026a.

Jiange Yang, Yansong Shi, Haoyi Zhu, Mingyu Liu, Kaijing Ma, Yating Wang, Gangshan Wu, Tong He, and Limin Wang. Como: Learning continuous latent motion from internet videos for scalable robot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 42352–42363, 2026b.

Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Dekang Qi, Junjin Xiao, Haoyun Liu, Ronghan Chen, Yuzhi Chen, Dongjie Huo, et al. ABot-M0: VLA foundation model for robotic manipulation with action manifold learning. *arXiv preprint arXiv:2602.11236*, 2026c.

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

Jinhui Ye, Ning Gao, Senqiao Yang, Jinliang Zheng, Zixuan Wang, Yuxin Chen, Pengguang Chen, Yilun Chen, Shu Liu, and Jiaya Jia. StarVLA-α: Reducing complexity in vision-language-action systems. *arXiv preprint arXiv:2604.11757*, 2026b.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. In *International Conference on Learning Representations*, volume 2025, pp. 28213–28239, 2025.

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

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026.

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.

Jianke Zhang, Yanjiang Guo, Yucheng Hu, Xiaoyu Chen, Xiang Zhu, and Jianyu Chen. UP-VLA: A unified understanding and prediction model for embodied agents. *arXiv preprint arXiv:2501.18867*, 2025.

Jianke Zhang, Yucheng Hu, Yanjiang Guo, Xiaoyu Chen, Yichen Liu, Wenna Chen, Chaochao Lu, and Jianyu Chen. Unijepa: Enhancing robot policy via unified continuous and discrete representation learning. In *International Conference on Machine Learning*, 2026b.

Kaizhao Zhang, Tian Niu, Tianyu Liu, Chenen Guo, Zijun Xu, Qingda Hu, and Wenchao Ding. Diffuview: Multi-view diffusion pretraining for 3d aware robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23601–23611, 2026c.

Qihang Zhang, Lin Li, Luyao Zhang, Shuai Yang, Yiming Luo, Shuaiting Li, Ruilin Wang, Junke Wang, Jiahao Shao, Gangwei Xu, et al. Native video-action pretraining for generalizable robot control. *arXiv preprint arXiv:2607.08639*, 2026d.

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*, 2026e.

Zhuoyang Zhang, Shang Yang, Qinghao Hu, Luke J Huang, James Hou, Yufei Sun, Yao Lu, and Song Han. Foreact: Steering your vla with efficient visual foresight planning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026f.

Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. RoboDreamer: Learning compositional world models for robot imagination. In *International Conference on Machine Learning*, 2024.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. In *Robotics: Science and Systems*, 2025.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, 2023.

A Theoretical Analysis of Native Action-Prior Learning

We provide a simplified theoretical analysis of why the Action-DiT pathway captures transition information and how future-state corruption prevents a direct target copying shortcut. Following the notation in Sec. 3.2, consider a transition from the preceding visual state z_{k-1} to the future state z_k. During pre-training, the Action-DiT does not directly observe the clean future state z_k, but instead accesses its noised version

$$z_{k,\tau} = (1-\tau)z_k + \tau\epsilon_k^v, \qquad \epsilon_k^v \sim \mathcal{N}(0, I), \tag{14}$$

as defined in Eq. 6. We abstract the transition representation formed by the Action-DiT as

$$r_k = E(z_{k-1}, z_{k,\tau}), \tag{15}$$

and consider predicting the clean future state from the preceding state and this representation,

$$\hat{z}_k = D(z_{k-1}, r_k). \tag{16}$$

This abstraction isolates the information carried through the Action-DiT pathway while suppressing the architectural details of the joint transformer.

The action pathway captures transition information. We first characterize the predictive value of the transition representation r_k. Define the Bayes-optimal prediction errors

$$\mathcal{L}_z = \mathbb{E}\Big[\|z_k - \mathbb{E}[z_k \mid z_{k-1}]\|_2^2\Big], \tag{17}$$

$$\mathcal{L}_{z,r} = \mathbb{E}\Big[\|z_k - \mathbb{E}[z_k \mid z_{k-1}, r_k]\|_2^2\Big]. \tag{18}$$

Proposition A.1 (Predictive value of the transition representation)**.** *The reduction in Bayes-optimal future-state prediction error from conditioning on r_k is*

$$\mathcal{L}_z - \mathcal{L}_{z,r} = \mathbb{E}\Big[\|\mathbb{E}[z_k \mid z_{k-1}, r_k] - \mathbb{E}[z_k \mid z_{k-1}]\|_2^2\Big] \geq 0. \tag{19}$$

Thus, r_k is useful to the extent that it captures information about the future state beyond what is already predictable from the preceding state z_{k-1}.

Proof. Let

$$m_{k-1} = \mathbb{E}[z_k \mid z_{k-1}], \qquad m_{k-1,r} = \mathbb{E}[z_k \mid z_{k-1}, r_k]. \tag{20}$$

We decompose

$$z_k - m_{k-1} = (z_k - m_{k-1,r}) + (m_{k-1,r} - m_{k-1}). \tag{21}$$

The two terms are orthogonal in expectation because

$$\mathbb{E}[z_k - m_{k-1,r} \mid z_{k-1}, r_k] = 0. \tag{22}$$

Therefore,

$$\mathbb{E}\|z_k - m_{k-1}\|_2^2 = \mathbb{E}\|z_k - m_{k-1,r}\|_2^2 + \mathbb{E}\|m_{k-1,r} - m_{k-1}\|_2^2, \tag{23}$$

which yields Eq. 19. □

Future corruption prevents a target-copying shortcut. If the Action-DiT directly observed the clean future state z_k, the prediction objective would admit the degenerate shortcut

$$r_k = z_k, \qquad D(z_{k-1}, r_k) = r_k, \tag{24}$$

which achieves zero prediction error by directly transmitting the prediction target, without requiring the action pathway to capture a meaningful transition representation. Instead, our attention construction exposes the Action-DiT only to the Gaussian-corrupted future state $z_{k,\tau}$.

Proposition A.2 (Gaussian corruption prevents exact target copying)**.** *Assume*

$$z_k \mid z_{k-1} \sim \mathcal{N}(\mu(z_{k-1}), \Sigma), \qquad \Sigma \succ 0, \tag{25}$$

and let $r_k = E(z_{k-1}, z_{k,\tau})$ be any representation computed from the corrupted future state in Eq. 14. Then

$$\mathcal{L}_{z,r} \geq \mathbb{E}\Big[\|z_k - \mathbb{E}[z_k \mid z_{k-1}, z_{k,\tau}]\|_2^2\Big]. \tag{26}$$

Moreover, for any $\tau > 0$,

$$\mathrm{Cov}(z_k \mid z_{k-1}, z_{k,\tau}) = \left(\Sigma^{-1} + \frac{(1-\tau)^2}{\tau^2} I\right)^{-1}, \tag{27}$$

and therefore

$$\mathcal{L}_{z,r} \geq \mathrm{tr}\left[\left(\Sigma^{-1} + \frac{(1-\tau)^2}{\tau^2} I\right)^{-1}\right] > 0. \tag{28}$$

Hence, at any non-zero corruption level, the clean future state cannot be transmitted perfectly through the transition representation r_k.

Proof. Because r_k is computed from $(z_{k-1}, z_{k,\tau})$, conditioning on (z_{k-1}, r_k) cannot provide more information about z_k than conditioning directly on $(z_{k-1}, z_{k,\tau})$. Therefore,

$$\mathcal{L}_{z,r} \geq \mathbb{E}\Big[\|z_k - \mathbb{E}[z_k \mid z_{k-1}, z_{k,\tau}]\|_2^2\Big]. \tag{29}$$

Under the Gaussian assumption in Eq. 25 and the corruption process in Eq. 14, standard Gaussian conditioning gives

$$\mathrm{Cov}(z_k \mid z_{k-1}, z_{k,\tau}) = \left(\Sigma^{-1} + \frac{(1-\tau)^2}{\tau^2} I\right)^{-1}. \tag{30}$$

The minimum mean-squared prediction error is the trace of this conditional covariance. Since $\Sigma \succ 0$ and $\tau > 0$, the conditional covariance is positive definite, yielding the strictly positive lower bound in Eq. 28. □

Together, Propositions A.1 and A.2 characterize the complementary roles of the transition representation and future-state corruption. Future prediction encourages the Action-DiT pathway to capture information about the transition beyond what is available from the preceding state, while future corruption prevents the objective from being solved by the trivial shortcut of directly transmitting the clean prediction target.

B Additional Implementation Details

This section provides additional implementation details for native action-prior pre-training, downstream post-training, benchmark evaluation, and action-only inference.

B.1 Native Action-Prior Pre-training

Pre-training data. We train on 235,790 observation-only manipulation episodes totaling 1,054.3 hours of video from Open X-Embodiment (O'Neill et al., 2024), AgiBotWorld (Bu et al., 2025a), and EgoDex (Hoque et al., 2026). For AgiBotWorld and EgoDex, we follow the OSCAR curation procedure (Wu & Gao, 2026), filtering short or low-quality interactions based on camera motion, manipulator activity, and hand visibility, followed by near-duplicate removal using visual and manipulator-trajectory similarity. Videos are divided into non-overlapping 65-frame clips, rendered on a 448×448 canvas, and temporally subsampled with a stride of 4, yielding approximately 1.41M training clips. Each clip is encoded by the frozen spatiotemporal VAE into five latent blocks, resulting in approximately 7.0M latent blocks for pre-training.

Optimization. Native action-prior pre-training uses only the future-video flow-matching objective in Eq. 7. As described in Sec. 3.2, the Video-DiT, text encoder, and VAE remain frozen, while only the Action-DiT is optimized. We train for 14,000 optimization steps on 128 GPUs using bf16 mixed precision and AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.95$, and weight decay 10^{-2}. We use a fixed learning rate of 1×10^{-4} and a global batch size of 512.

Table 3: **Downstream training hyperparameters on LIBERO.** We first perform action-free target-domain cold start and subsequently jointly post-train the Video-DiT and Action-DiT using action-labeled demonstrations.

	Cold Start	**Joint Post-training**
Trainable	Video-DiT	Video-DiT + Action-DiT
Action horizon	—	16
Learning rate	4×10^{-5} (video)	1×10^{-4} (video), 2×10^{-4} (action)
Epochs	5	10
Scheduler	cosine, 5% warm-up	
Global batch size	256	
Observation	224×448 composite (agent-view + wrist-view)	

Table 4: **Downstream training hyperparameters on RoboTwin 2.0.** We first perform action-free target-domain cold start and subsequently jointly post-train the Video-DiT and Action-DiT using Clean-domain demonstrations.

	Cold Start	**Joint Post-training**
Trainable	Video-DiT	Video-DiT + Action-DiT
Action horizon	—	16
Learning rate	2×10^{-4} (video)	1×10^{-4} (video), 3×10^{-4} (action)
Epochs	1	5
Scheduler	cosine, 5% warm-up	
Global batch size	1024	
Observation	384×320 composite (head-view + two wrist-views)	

B.2 Downstream Post-training

For downstream adaptation, we first perform an action-free cold start to adapt the Video-DiT to the target visual domain, following Chen et al. (2026c). During this stage, the pretrained Action-DiT remains frozen, while the Video-DiT is optimized using future-video flow matching on target-domain videos without action labels. This stage serves solely as target-domain visual initialization and introduces no additional action supervision. We then unfreeze both experts and jointly post-train the Video-DiT and Action-DiT using Eq. 11 with $\lambda_v = \lambda_a = 1$.

For both LIBERO and RoboTwin 2.0, downstream training is initialized from the same checkpoint pretrained with native action-prior learning. Benchmark-specific cold-start and joint post-training configurations are summarized in Tabs. 3 and 4, respectively.

B.3 Detailed Transition-Structured Attention Masks

Fig. 7 provides a detailed visualization of the transition-structured attention masks introduced in Sec. 3.2. For a sequence of visual transitions, we construct one segment for each transition, where segment k contains the clean preceding state z_{k-1}, the noised future state z_{k,τ_v}, and the corresponding Action-DiT token group x_k^a. Different segments are fully isolated from one another, ensuring that each segment models its corresponding transition without accessing information from other transitions.

Within each segment, the clean preceding-state queries attend only to themselves. During pre-training, the noised-future and action-token queries attend to all token groups within the segment, allowing future-video supervision to propagate through the Action-DiT. During post-training, noised-future video queries attend only to visual tokens, while action queries retain access to both visual and action tokens, yielding the asymmetric interaction described in Sec. 3.3.

B.4 Benchmark and Evaluation Details

Baselines. We compare NAVA-WAM with representative action policies and recent world action models reported in Tab. 1. For RoboTwin 2.0, baseline results follow the Clean-to-Random eval-

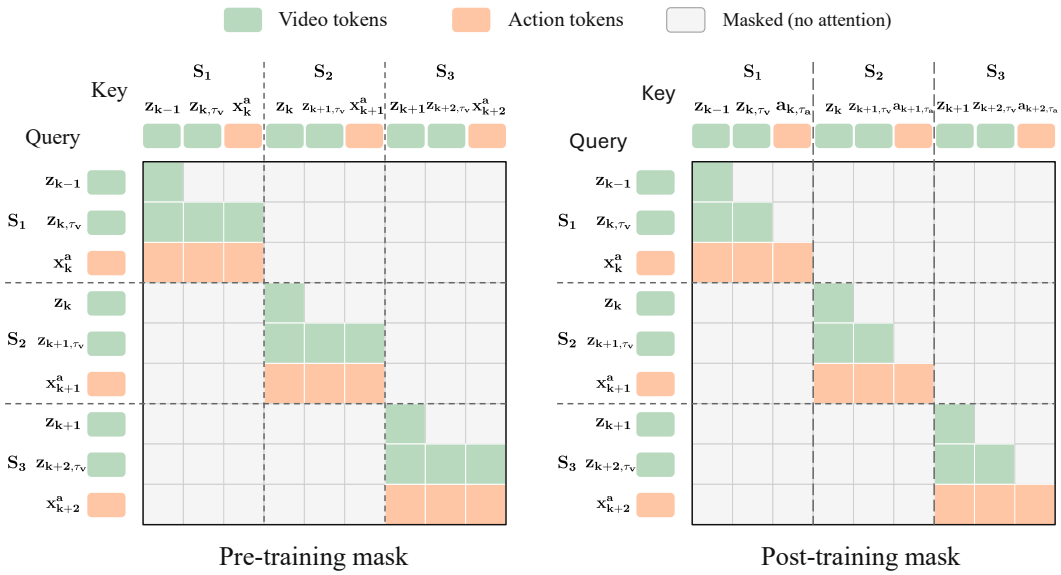

Figure 7: **Transition-structured attention masks.** The pre-training mask (*left*) and post-training mask (*right*) are shown for three consecutive transition segments S_1–S_3. Each segment contains a clean preceding visual state, a noised future state, and the corresponding Action-DiT tokens, with attention isolated across segments. Pre-training allows noised-future and action-token queries to attend to all tokens within their segment, whereas during post-training, noised-future video queries attend only to visual tokens.

uation reported by MV-WAM (Chen et al., 2026c). Unless otherwise specified, all other baseline results are taken from the corresponding papers.

LIBERO. LIBERO (Liu et al., 2023) is built on the robosuite simulator and comprises four task suites covering different manipulation and generalization settings. *LIBERO-Spatial* varies the spatial relationships between objects, *LIBERO-Object* varies the object category to be manipulated, and *LIBERO-Goal* fixes the objects and their spatial arrangement while varying the task goal. The fourth suite, *LIBERO-Long*, focuses on long-horizon tasks that combine these factors. Each suite contains ten tasks with 50 demonstrations per task. Following the standard protocol (Zhang et al., 2026e; Yuan et al., 2026), we train a single multi-task policy on all 40 tasks using 50 demonstrations per task, totaling 2,000 trajectories, and evaluate each task over 50 rollouts with a 7-D action space.

LIBERO-Plus. LIBERO-Plus (Fei et al., 2025) reconstructs the 40 LIBERO evaluation tasks under controlled perturbations along seven factors: *object layout*, *camera viewpoints*, *robot initial states*, *language instructions*, *lighting conditions*, *background textures*, and *sensor noise*. The full benchmark comprises 10,030 test-only task instances. We use no LIBERO-Plus trajectories for training and directly evaluate the policy post-trained on the original LIBERO demonstrations. Following Zhang et al. (2026e), we perform one rollout per instance and report the macro-averaged success rate across perturbation types and task suites in Sec. C.3.

RoboTwin 2.0. RoboTwin 2.0 (Chen et al., 2025a) provides bimanual manipulation tasks under Clean and domain-randomized conditions. It applies structured domain randomization along five axes: *task-irrelevant distractor objects*, *background textures*, *lighting conditions*, *tabletop height*, and *language instructions*. We evaluate the 50-task multi-task setting and, following Chen et al. (2026c), train exclusively on 50 Clean-domain demonstrations per task, totaling 2,500 action-labeled trajectories. No Random-domain demonstrations are used for training, making the Random domain an OOD evaluation relative to Clean-domain training. We evaluate 100 rollouts per task under both conditions with a 14-D action space.

Algorithm 1 Action-only inference with cached Video-DiT representations

Require: observation latent z_0; instruction l; proprioception q; flow steps S; step size $\Delta\tau = 1/S$
Ensure: action chunk $\hat{a}$ of horizon H

$\epsilon^v, \epsilon^a \sim \mathcal{N}(0, I)$
$a \leftarrow \epsilon^a; \quad \tau_a \leftarrow 1$
$(u^a, \mathcal{C}_V) \leftarrow \mathrm{JointStep}([z_0, \epsilon^v], a, \tau_v = 1, \tau_a = 1 \mid l, q)$
$a \leftarrow a - \Delta\tau\, u^a$
$\tau_a \leftarrow \tau_a - \Delta\tau$
for $s = 2$ **to** S **do**
  $u^a \leftarrow v^a_{\theta_A}(a, \tau_a \mid \mathcal{C}_V, l, q)$
  $a \leftarrow a - \Delta\tau\, u^a$
  $\tau_a \leftarrow \tau_a - \Delta\tau$
end for
return $\hat{a} \leftarrow a$

B.5 Action-Only Inference

The asymmetric transition attention introduced in Sec. 3.3 makes the Video-DiT independent of the evolving action sample, allowing its representations to be reused across action-denoising steps. At each replanning step, we initialize the future-video and action streams with independent Gaussian noise and perform a single joint forward pass at $\tau_v = \tau_a = 1$. During this pass, we cache the layer-wise Video-DiT keys and values. All subsequent denoising steps evaluate only the Action-DiT while reusing the cached visual representations.

We integrate the action flow using $S = 10$ uniform Euler steps for both LIBERO and RoboTwin 2.0. The future-video latent is neither iteratively denoised nor decoded, and the visual cache is recomputed once at each replanning step. Algorithm 1 summarizes the inference procedure.

C Additional Experimental Results

This section provides additional analyses of native action-prior pre-training, qualitative visualizations of learned transition priors, detailed benchmark breakdowns, and RoboTwin 2.0 policy rollouts.

C.1 Analysis of Native Action-Prior Pre-training

Pre-training dynamics. Fig. 8 shows the training and held-out future-video flow-matching losses throughout native action-prior pre-training. Both losses decrease rapidly during the first few thousand optimization steps and continue to decline gradually thereafter, indicating stable optimization.

Future-video prediction. Fig. 9 visualizes future-video predictions on held-out egocentric manipulation clips after native action-prior pre-training. Each temporally subsampled clip contains 17 frames and is encoded into 5 VAE latent blocks. The first block is kept clean, while each subsequent block is independently denoised from Gaussian noise using 10 flow-matching steps. The predictions show that the model captures both hand motion and the evolution of manipulated objects across diverse interactions, indicating that the pretrained Action-DiT effectively supports visual-transition prediction.

C.2 Qualitative Analysis of Learned Transition Priors

Fig. 10 provides additional motion-transfer examples complementing Fig. 6. Across source–target pairs with substantial differences in scene appearance, layout, and camera viewpoint, the transferred target states follow the motion specified by the corresponding source transitions in simulation, real-world, and sim-to-real settings. These examples suggest that the pretrained Action-DiT captures transition-relevant action information that transfers across visual domains.

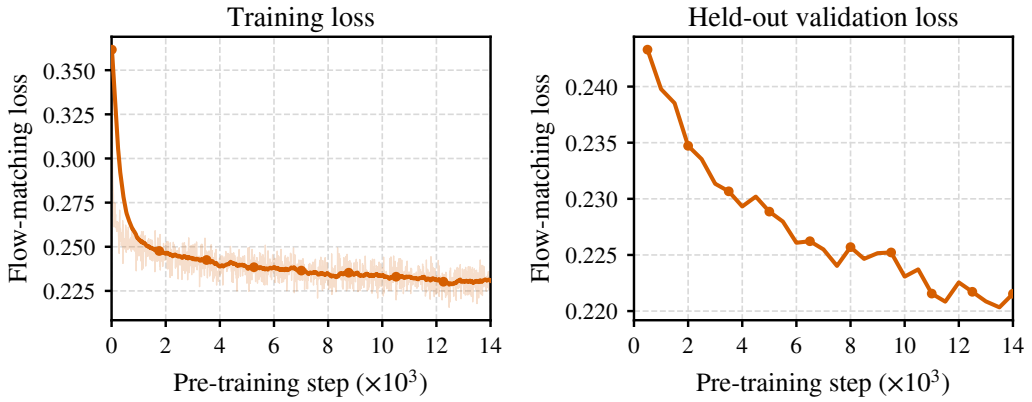

Figure 8: **Native action-prior pre-training dynamics.** Training and held-out future-video flow-matching losses over pre-training. The training curve shows the raw per-step loss (faint) and its exponential moving average (solid), while the held-out loss is evaluated periodically without smoothing.

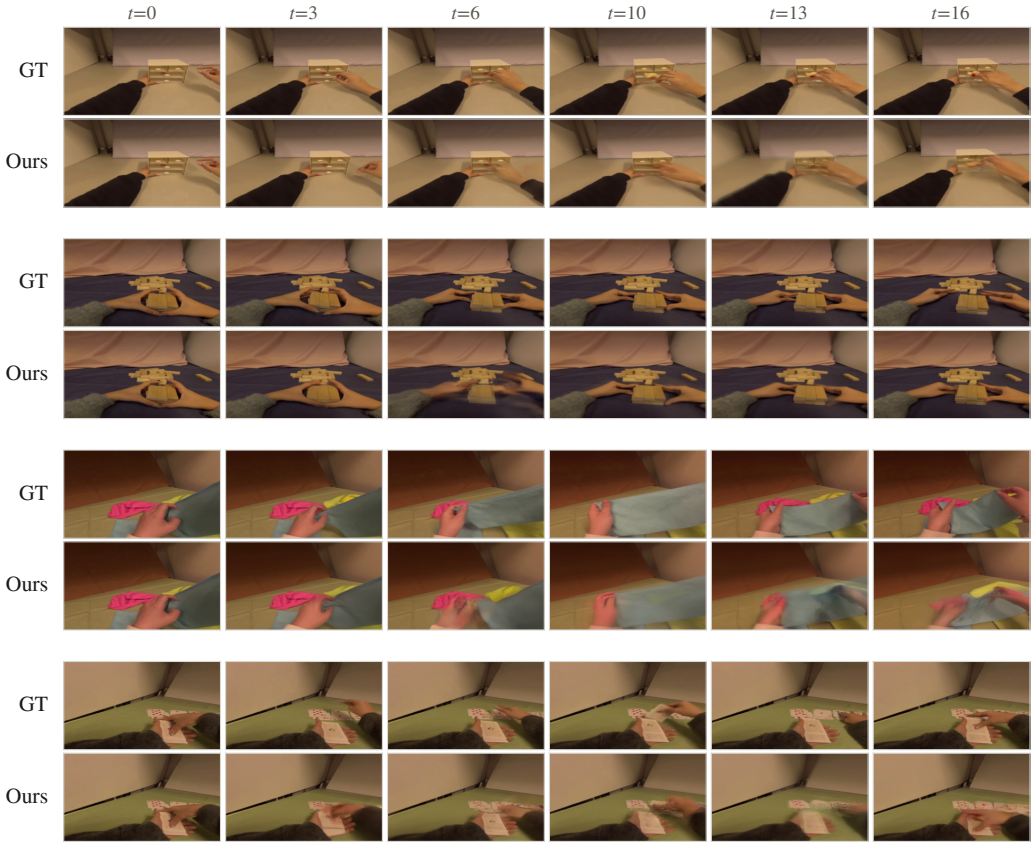

Figure 9: **Future-video prediction after native action-prior pre-training.** Predictions on four held-out egocentric manipulation clips. Each example shows the ground-truth sequence (*GT*) and the corresponding prediction from NAVA-WAM (*Ours*) across block stacking, two-handed assembly, cloth folding, and card manipulation.

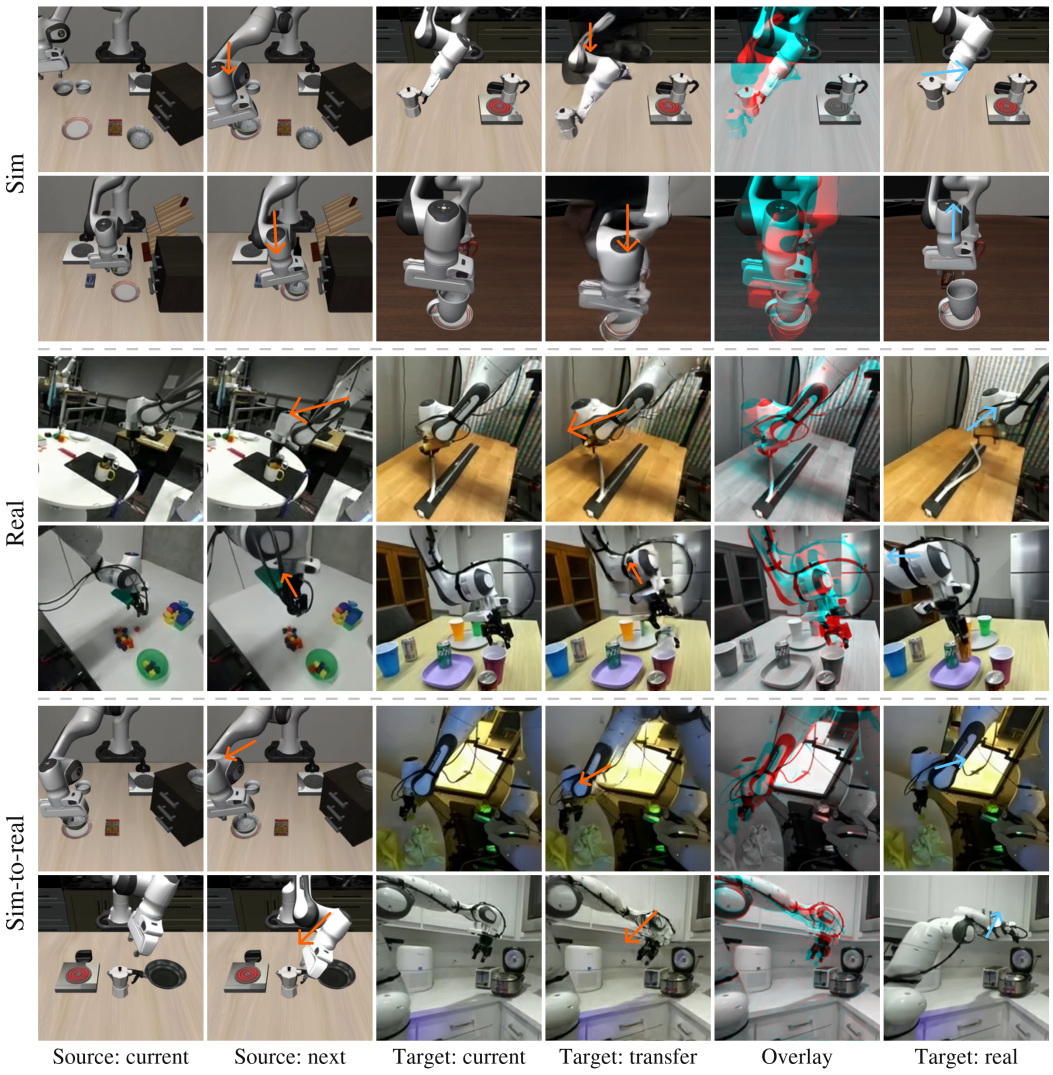

Figure 10: **Additional qualitative visualizations of motion transfer across domains.** Each example transfers the transition encoded by a source pair (*Source: current* → *Source: next*) to a different target observation. The source transition specifies the underlying motion cue (orange arrows), which is applied to *Target: current* to obtain *Target: transfer*. The *Overlay* visualizes the transferred state relative to the current target observation, while *Target: real* shows the corresponding real transition (blue arrows). Across simulation, real-world, and sim-to-real examples, the transferred states follow the source motion despite substantial changes in visual appearance and scene configuration.

C.3 Detailed Benchmark Results

We report per-suite, per-perturbation, and per-task results underlying the aggregate success rates in Tab. 1.

LIBERO. Tab. 5 reports success rates across the four LIBERO suites. NAVA-WAM performs consistently across all suites and achieves 99.0% success on LIBERO-Long, where several compared methods show their largest performance drop.

LIBERO-Plus. Tab. 6 reports success rates across the seven LIBERO-Plus perturbation types, with baseline results taken from Zhang et al. (2026e). NAVA-WAM performs best on robot-initialization, language, and object-layout shifts, reaching 65.6%, 93.8%, and 87.1%, respectively.

Table 5: **Per-suite success rates (%) on LIBERO.** All methods are trained on 50 demonstrations per task and evaluated over 50 rollouts per task. *Avg.* denotes the macro-average across the four task suites. Baselines with available per-suite results are reported. The best result in each column is shown in **bold**.

Method	Spatial	Object	Goal	Long	Avg.
Direct action policies					
π_0 (2024a)	96.8	98.8	95.8	85.2	94.1
$\pi_{0.5}$ (2025)	**98.8**	98.2	98.0	92.4	96.9
World action models					
Fast-WAM (2026)	98.2	**100.0**	97.0	95.2	97.6
Image-WAM (2026e)	97.2	99.2	**98.8**	98.4	98.4
NAVA-WAM	**98.8**	99.6	98.6	**99.0**	**99.0**

Table 6: **Per-perturbation success rates (%) on LIBERO-Plus.** Each column averages over the four LIBERO task suites, and *Avg.* denotes the macro-average across the seven perturbation types. No LIBERO-Plus trajectories are used for training. The best result in each column is shown in **bold**.

Method	Camera	Robot	Language	Light	Background	Noise	Layout	Avg.
π_0 (2024a)	13.8	6.0	58.8	85.0	81.4	79.0	68.9	53.6
π_0-FAST (2025)	65.1	21.6	61.0	73.2	73.2	74.4	68.8	61.6
Fast-WAM (2026)	16.4	44.5	68.9	78.2	53.7	37.7	60.7	51.5
Image-WAM (2026e)	**80.8**	50.3	91.4	**98.1**	**85.5**	**93.8**	80.5	83.1
NAVA-WAM	72.8	**65.6**	**93.8**	97.9	82.8	84.8	**87.1**	**83.5**

Image-WAM performs better on camera, lighting, background, and sensor-noise perturbations. Overall, NAVA-WAM achieves the highest macro-average success rate, demonstrating strong out-of-domain generalization across diverse perturbations.

RoboTwin 2.0. Tab. 7 reports success rates across all 50 RoboTwin 2.0 tasks. NAVA-WAM achieves the best or tied-best Random-domain performance on 44 of the 50 tasks and the best or tied-best Clean-domain performance on 32 tasks. Its aggregate Clean-to-Random performance drop is also substantially smaller than those of the compared baselines, indicating stronger robustness to the Random-domain shift.

C.4 Qualitative RoboTwin 2.0 Rollouts

Fig. 11 presents representative NAVA-WAM rollouts from the RoboTwin 2.0 Random domain. All examples are drawn from the same evaluation setting as Tab. 7, where the policy is trained exclusively on Clean-domain demonstrations and evaluated under domain randomization unseen during training. Each Random-domain rollout therefore constitutes an out-of-domain evaluation under variations in distractor objects, background textures, lighting, tabletop height, and language instructions. Videos of these rollouts are provided in the *HTML file* included in the supplementary material.

Successful rollouts. The successful rollouts span diverse manipulation skills in RoboTwin 2.0. *Handover Mic* and *Lift Pot* require bimanual coordination, with the former transferring an object between the two arms and the latter synchronously lifting an object with both grippers. *Place Dual Shoes* combines bimanual manipulation with a terminal pose constraint, requiring both shoes to be placed inside the shoebox with the prescribed orientation. *Place Bread Skillet* requires precise single-arm placement into a small container, while *Open Laptop* requires manipulating an articulated object along its hinge. *Blocks Ranking RGB* and *Stack Blocks Three* are long-horizon multi-object tasks that require grounding color references and executing multiple consecutive pick-and-place operations without disturbing previously placed blocks. Across these tasks, NAVA-WAM completes

Table 7: **Per-task success rates (%) on RoboTwin 2.0.** All methods are trained exclusively on Clean-domain demonstrations and evaluated in both the Clean and Random (*Rand.*) domains. Baseline per-task results are taken from MV-WAM (Chen et al., 2026c). The final row reports the average across all 50 tasks. The best result for each task and domain is shown in **bold**.

Task	DP		RDT		π_0		UP-VLA		BagelVLA		HALO		Fast-WAM		MV-WAM		**Ours**	
	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.
Adjust Bottle	97.0	0.0	81.0	75.0	90.0	56.0	**100.0**	17.0	**100.0**	14.0	97.0	9.0	95.0	0.0	83.0	65.0	**100.0**	**97.5**
Beat Block Hammer	42.0	0.0	77.0	37.0	43.0	21.0	66.0	16.0	87.0	16.0	96.0	11.0	74.0	2.0	75.0	**53.0**	**100.0**	42.5
Blocks Ranking RGB	0.0	0.0	3.0	0.0	19.0	5.0	38.0	0.0	84.0	4.0	94.0	7.0	0.0	1.0	99.0	88.0	**100.0**	**92.5**
Blocks Ranking Size	1.0	0.0	0.0	0.0	7.0	1.0	21.0	0.0	45.0	2.0	58.0	2.0	38.0	0.0	63.0	43.0	**70.0**	**60.0**
Click Alarmclock	61.0	5.0	61.0	12.0	63.0	11.0	69.0	41.0	85.0	20.0	83.0	14.0	**100.0**	38.0	90.0	24.0	**100.0**	**100.0**
Click Bell	54.0	0.0	80.0	9.0	44.0	3.0	54.0	72.0	**100.0**	35.0	**100.0**	10.0	**100.0**	22.0	95.0	43.0	**100.0**	**90.0**
Dump Bin Bigbin	49.0	0.0	64.0	32.0	83.0	24.0	81.0	35.0	91.0	51.0	93.0	28.0	96.0	3.0	92.0	61.0	**97.5**	**77.5**
Grab Roller	98.0	0.0	74.0	43.0	96.0	80.0	99.0	28.0	99.0	41.0	95.0	57.0	95.0	4.0	**100.0**	94.0	**100.0**	**100.0**
Handover Block	10.0	0.0	45.0	14.0	45.0	8.0	4.0	0.0	38.0	0.0	81.0	36.0	5.0	0.0	**83.0**	19.0	82.5	**70.0**
Handover Mic	53.0	0.0	90.0	31.0	98.0	13.0	45.0	0.0	75.0	8.0	96.0	61.0	**100.0**	0.0	99.0	86.0	95.0	**100.0**
Hanging Mug	8.0	0.0	23.0	16.0	11.0	3.0	4.0	0.0	12.0	1.0	**48.0**	5.0	32.0	0.0	46.0	16.0	45.0	**27.5**
Lift Pot	39.0	0.0	72.0	9.0	84.0	36.0	20.0	0.0	87.0	32.0	95.0	34.0	89.0	0.0	**100.0**	46.0	97.5	**60.0**
Move Can Pot	39.0	0.0	25.0	12.0	58.0	21.0	48.0	0.0	78.0	0.0	94.0	15.0	85.0	7.0	87.0	24.0	**95.0**	**25.0**
Move Pillbottle Pad	1.0	0.0	8.0	0.0	21.0	1.0	51.0	7.0	92.0	1.0	76.0	26.0	92.0	1.0	90.0	34.0	**100.0**	**85.0**
Move Playingcard Away	47.0	0.0	43.0	11.0	53.0	22.0	79.0	13.0	92.0	30.0	89.0	53.0	99.0	2.0	98.0	89.0	**100.0**	**92.5**
Move Stapler Pad	1.0	0.0	2.0	0.0	0.0	2.0	8.0	0.0	27.0	0.0	45.0	19.0	35.0	0.0	24.0	11.0	**65.0**	**47.5**
Open Laptop	49.0	0.0	59.0	32.0	85.0	46.0	86.0	21.0	**96.0**	37.0	89.0	37.0	90.0	8.0	94.0	59.0	92.5	**67.5**
Open Microwave	5.0	0.0	37.0	20.0	80.0	**50.0**	2.0	7.0	0.0	0.0	**86.0**	24.0	44.0	1.0	50.0	2.0	35.0	5.0
Pick Diverse Bottles	6.0	0.0	2.0	0.0	27.0	6.0	52.0	18.0	83.0	34.0	76.0	17.0	69.0	4.0	81.0	50.0	**90.0**	**57.5**
Pick Dual Bottles	24.0	0.0	42.0	13.0	57.0	12.0	82.0	31.0	93.0	56.0	82.0	30.0	76.0	9.0	93.0	56.0	**100.0**	**75.0**
Place A2B Left	2.0	0.0	3.0	1.0	31.0	1.0	74.0	4.0	79.0	12.0	68.0	8.0	76.0	1.0	90.0	69.0	**95.0**	**77.5**
Place A2B Right	13.0	0.0	1.0	1.0	27.0	6.0	56.0	1.0	81.0	11.0	52.0	9.0	84.0	3.0	87.0	68.0	**92.5**	**90.0**
Place Bread Basket	14.0	0.0	10.0	2.0	17.0	4.0	63.0	20.0	90.0	29.0	90.0	26.0	**96.0**	4.0	92.0	71.0	92.5	**90.0**
Place Bread Skillet	11.0	0.0	5.0	1.0	23.0	1.0	71.0	16.0	91.0	26.0	85.0	23.0	85.0	3.0	90.0	60.0	**92.5**	**87.5**
Place Burger Fries	72.0	0.0	50.0	27.0	80.0	4.0	97.0	26.0	99.0	11.0	99.0	37.0	96.0	8.0	90.0	79.0	**100.0**	**100.0**
Place Can Basket	18.0	0.0	19.0	6.0	41.0	5.0	20.0	0.0	63.0	0.0	68.0	34.0	58.0	0.0	**80.0**	44.0	**80.0**	**60.0**
Place Cans Plasticbox	40.0	0.0	6.0	5.0	34.0	2.0	66.0	24.0	94.0	5.0	98.0	47.0	93.0	2.0	**99.0**	48.0	97.5	**90.0**
Place Container Plate	41.0	0.0	78.0	17.0	88.0	45.0	86.0	48.0	**100.0**	58.0	96.0	22.0	98.0	17.0	96.0	**86.0**	97.5	77.5
Place Dual Shoes	8.0	0.0	4.0	4.0	15.0	0.0	45.0	0.0	57.0	0.0	15.0	3.0	24.0	0.0	42.0	24.0	**80.0**	**52.5**
Place Empty Cup	37.0	0.0	56.0	7.0	37.0	11.0	74.0	27.0	97.0	34.0	95.0	28.0	98.0	10.0	92.0	**86.0**	**100.0**	85.0
Place Fan	3.0	0.0	12.0	2.0	20.0	10.0	31.0	1.0	62.0	5.0	62.0	9.0	75.0	1.0	76.0	42.0	**85.0**	**80.0**
Place Mouse Pad	0.0	0.0	1.0	0.0	7.0	1.0	27.0	0.0	46.0	14.0	51.0	12.0	71.0	0.0	73.0	34.0	**85.0**	**57.5**
Place Object Basket	15.0	0.0	33.0	17.0	16.0	2.0	56.0	1.0	66.0	3.0	89.0	25.0	50.0	2.0	**93.0**	**78.0**	80.0	65.0
Place Object Scale	1.0	0.0	1.0	0.0	10.0	0.0	36.0	4.0	71.0	0.0	55.0	5.0	80.0	0.0	**91.0**	53.0	82.5	**77.5**
Place Object Stand	22.0	0.0	15.0	5.0	36.0	11.0	76.0	24.0	87.0	21.0	84.0	33.0	95.0	12.0	93.0	72.0	**97.5**	**75.0**
Place Phone Stand	13.0	0.0	15.0	6.0	35.0	7.0	32.0	0.0	61.0	2.0	**91.0**	10.0	85.0	0.0	79.0	52.0	85.0	**72.5**
Place Shoe	23.0	0.0	35.0	7.0	28.0	6.0	76.0	12.0	90.0	29.0	70.0	18.0	83.0	7.0	91.0	79.0	**95.0**	**97.5**
Press Stapler	6.0	0.0	41.0	24.0	62.0	29.0	79.0	56.0	94.0	58.0	92.0	64.0	58.0	19.0	**99.0**	45.0	90.0	**67.5**
Put Bottles Dustbin	22.0	0.0	21.0	4.0	54.0	13.0	7.0	0.0	42.0	10.0	80.0	13.0	83.0	2.0	**93.0**	43.0	87.5	**75.0**
Put Object Cabinet	42.0	0.0	33.0	18.0	**68.0**	18.0	7.0	0.0	52.0	0.0	59.0	8.0	41.0	0.0	45.0	25.0	50.0	**37.5**
Rotate QRcode	13.0	0.0	50.0	5.0	68.0	15.0	56.0	2.0	81.0	21.0	69.0	11.0	76.0	0.0	79.0	40.0	**90.0**	**70.0**
Scan Object	9.0	0.0	4.0	1.0	18.0	1.0	47.0	23.0	77.0	32.0	73.0	24.0	77.0	4.0	81.0	59.0	**85.0**	**67.5**
Shake Bottle Horizontally	59.0	18.0	84.0	51.0	99.0	51.0	**100.0**	68.0	**100.0**	73.0	**100.0**	66.0	**100.0**	41.0	**100.0**	97.0	**100.0**	**100.0**
Shake Bottle	65.0	8.0	74.0	45.0	97.0	60.0	98.0	54.0	**100.0**	74.0	98.0	73.0	**100.0**	49.0	**100.0**	98.0	**100.0**	**100.0**
Stack Blocks Three	0.0	0.0	2.0	0.0	17.0	0.0	8.0	0.0	45.0	5.0	96.0	37.0	0.0	0.0	97.0	51.0	**97.5**	**87.5**
Stack Blocks Two	7.0	0.0	21.0	2.0	42.0	1.0	61.0	0.0	95.0	6.0	**100.0**	60.0	2.0	0.0	**100.0**	75.0	**100.0**	**95.0**
Stack Bowls Three	63.0	0.0	51.0	17.0	66.0	24.0	42.0	1.0	63.0	13.0	**92.0**	25.0	77.0	1.0	86.0	**68.0**	80.0	62.5
Stack Bowls Two	61.0	0.0	76.0	30.0	91.0	41.0	69.0	12.0	90.0	52.0	**98.0**	49.0	96.0	11.0	90.0	81.0	97.5	**90.0**
Stamp Seal	2.0	0.0	1.0	0.0	3.0	4.0	34.0	2.0	77.0	8.0	60.0	21.0	68.0	0.0	73.0	30.0	**90.0**	**52.5**
Turn Switch	36.0	1.0	35.0	15.0	27.0	23.0	43.0	26.0	49.0	30.0	**65.0**	27.0	56.0	17.0	62.0	65.0	55.0	**67.5**
Average	28.0	0.6	34.5	13.7	46.4	16.3	52.9	15.2	75.3	20.5	80.5	26.4	71.9	6.3	84.0	55.7	**88.5**	**73.6**

the instructed behaviors despite substantial variation in distractors, appearance, and scene configuration.

Failure cases. The two representative failures occur during fine-grained grasping and contact-rich manipulation. In *Move Can Pot*, the reaching motion knocks the can over instead of establishing a stable grasp; the arm then executes the transport motion with an empty gripper without re-attempting the grasp, and the episode eventually reaches its step limit. In *Hanging Mug*, the policy completes the initial pick sub-goal, lifting the mug with the left arm and placing it beside the rack, but the right arm fails to continue the task by lifting the mug onto the rack, and the episode eventually reaches its step limit.

Together, these cases suggest that failures can arise during fine-grained, contact-rich interactions even when the target object and overall task progression are correctly identified. In both examples, the policy fails to recover from a local execution error, causing it to propagate into task failure.



Figure 11: **Qualitative NAVA-WAM rollouts on RoboTwin 2.0.** We show seven successful rollouts (top) spanning diverse manipulation skills and two representative failures (bottom). Each row contains four temporal keyframes, with the head-camera view on the left and the two wrist-camera views stacked on the right. The row header gives the RoboTwin task description, and the final keyframe is outlined in blue for success and red for failure. All examples are from the Random (OOD) domain.

Table 8: **Real-robot success rates and inference cost.** We report successful trials out of five runs per task on a Franka FR3, with *Avg.* denoting the success rate across all 15 trials. T1: *move the cube to the left side of the bowl*; T2: *put the banana in the box*; T3: *put the blue cube on the red cube*. *ms / call* denotes the server-side wall-clock latency per policy call on a single H100. The best result in each column is shown in **bold**.

		Successful trials ↑			
Model	ms / call ↓	T1	T2	T3	Avg (%) ↑
$\pi_{0.5}$ (Physical Intelligence et al., 2025)	**195.7**	0/5	**5/5**	3/5	53.3
DreamZero (Ye et al., 2026c)	3523.9	0/5	**5/5**	**5/5**	66.7
NAVA-WAM	379.2	**4/5**	**5/5**	**5/5**	**93.3**

C.5 Real-Robot Evaluation Details

To evaluate real-world transfer, we further post-train NAVA-WAM on DROID (Khazatsky et al., 2024) and directly deploy the resulting policy on a physical Franka FR3 robot following Ye et al. (2026c). DROID is a large-scale in-the-wild manipulation dataset containing approximately 76K teleoperated trajectories across diverse tasks and environments. *No additional fine-tuning is performed on the evaluation tasks or physical test environment.*

DROID post-training. Starting from the native action-prior pretrained checkpoint, we post-train NAVA-WAM on DROID following the training setup of DreamZero (Ye et al., 2026c). We use a global batch size of 256, a cosine learning-rate schedule with 5% warm-up, and a learning rate of 5×10^{-5} for both experts. The policy predicts action chunks with a horizon of 32, and visual inputs are processed at a resolution of 352×640.

Deployment setup. The policy receives one wrist-camera view and two third-person RGB views, composited into the same 352×640 visual input used during DROID post-training. NAVA-WAM predicts 32-step absolute joint-position action chunks, of which the first 24 steps are executed open-loop before replanning. The robot controller runs at 15 Hz, and each trial is capped at $1{,}000$ control steps.

Tasks and baselines. We evaluate three tabletop manipulation tasks with five trials per task: T1, *move the cube to the left side of the bowl*; T2, *put the banana in the box*; and T3, *put the blue cube on the red cube*. We compare against $\pi_{0.5}$ (Physical Intelligence et al., 2025) and the released DreamZero-DROID checkpoint (Ye et al., 2026c). All policies use the same observation pipeline, communication protocol, and 15 Hz joint-position controller. The scene is manually reset to a random nominal layout before each trial. Each policy retains its own action parameterization and deployment horizon. $\pi_{0.5}$ predicts 24 relative-action steps and executes the first 8 before replanning, DreamZero predicts and executes 24 absolute-action steps, and NAVA-WAM predicts 32 absolute-action steps and executes the first 24.

Detailed task analysis. As summarized in Tab. 8, NAVA-WAM succeeds in 14 of 15 trials (93.3%), compared with $10/15$ (66.7%) for DreamZero and $8/15$ (53.3%) for $\pi_{0.5}$. All three policies solve T2 in all five trials, showing that each policy can complete the basic manipulation under our deployment setup.

The largest difference occurs on T1, which requires grounding the instructed spatial relation: NAVA-WAM succeeds in $4/5$ trials, whereas both baselines fail in all five trials. Inspection of the baseline rollouts suggests that these failures are not primarily caused by object localization or grasping. In all ten baseline trials, the cube is successfully localized and grasped, but neither baseline ultimately places it to the left of the bowl. DreamZero instead moves the cube toward the bowl and terminates with it either inside the bowl or still held above it, while $\pi_{0.5}$ places the cube to the right of the bowl or inside it. On T3, both NAVA-WAM and DreamZero succeed in all five stacking trials, while $\pi_{0.5}$ succeeds in three. In its two failed trials, $\pi_{0.5}$ places the blue cube beside the red cube rather than on top of it. These results demonstrate that NAVA-WAM transfers effectively to physical robot deployment *without task-specific fine-tuning*.

Inference cost. We additionally report the server-side latency of one policy call on a single H100. NAVA-WAM requires 379.2 ms per call, compared with 195.7 ms for $\pi_{0.5}$ and 3523.9 ms for DreamZero. Because the policies execute different numbers of actions before replanning, per-call latency alone does not fully reflect the effective deployment cost. Normalizing by the number of executed actions per replanning step yields 15.8 ms per executed action for NAVA-WAM, 24.5 ms for $\pi_{0.5}$, and 146.8 ms for DreamZero.

D LIMITATIONS

First, NAVA-WAM can still struggle with challenging tasks that require precise grasping, complex interactions, and other fine-grained contact-rich manipulation. Fine-grained control and complex-instruction understanding therefore remain limitations that warrant further investigation. Second, the policy currently lacks an explicit mechanism for recovering from local execution errors. Developing mechanisms that enable the policy to identify such failures and recover during execution is an important direction for future work.