3DFLOWWAM: BRIDGING VIDEO AND ACTION THROUGH 3D FLOW

Anonymous authors
Paper under double-blind review



Figure 1: **3DFlowWAM bridges video prediction and robot control via image-aligned 3D Flow.** Given past RGB observations, the model jointly generates future video, 3D Flow, and robot actions. RGB captures appearance and semantics, while 3D Flow pairs Point Maps (where visible points are in 3D) with world-centric Scene Flow (how they move). The predicted action chunks drive closed-loop execution, connecting video and 3D Flow predictions to embodiment-specific control.

ABSTRACT

Robot actions are expressed in embodiment-specific coordinates, but their effects can be described through shared 3D geometry and motion. We introduce 3DFlowWAM, a world action model that uses 3D scene changes to connect video priors to robot control. Its central representation, 3D Flow, describes visible points through their 3D positions and displacements, preserving correspondence with video while remaining independent of robot action coordinates. By jointly predicting video, 3D Flow, and actions, the model learns to connect visual appearance with explicit spatial dynamics. We build on a pretrained video model through 3D Flow-aware mid-training, followed by task-specific post-training. We assess the contribution of 3D Flow through controlled comparisons on a common DreamZero backbone with matched video and action data. Experiments across simulation benchmarks and seven real-world tasks show that 3D Flow improves action learning both with and without mid-training, while mid-training further strengthens real-world adaptation.

1 INTRODUCTION

World action models (WAMs) couple video generation with action prediction to learn control from future-frame supervision and broad video priors (Huang et al., 2026; Ye et al., 2026; Bi et al., 2026). This is attractive because video is readily available at scale, providing complementary supervision to robot action data that are costly to collect and tied to specific embodiments. Video captures rich

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.

appearance and semantics, but leaves limited motion cues directly relevant to robot actions. These actions unfold in 3D space, and a complementary challenge for WAMs is to make action learning as scalable as video learning.

A suitable action representation should be grounded in 3D, temporally aligned with video, informative about robot–object interactions, and independent of any particular embodiment. Prior work has explored 2D optical flow (Ranasinghe et al., 2026; Chen et al., 2026c), depth (Guo et al., 2026; Li et al., 2026), 6D object pose (Lin et al., 2026a), and latent actions inferred from video frame transitions (Bu et al., 2025b). These representations only satisfy a subset of the above requirements.

We focus on *3D Flow*, which comprises two components: an image-aligned Point Map describing *where each visible point is in 3D*, and world-centric Scene Flow describing *how it moves*. Besides meeting those requirements, 3D Flow also has the potential to scale. First, recent advances in dense 3D foundation tracking model make it increasingly practical to recover reliable 3D point trajectories from RGB videos at scale, enabling automatic construction of geometry and motion supervision (Lu et al., 2026; Zhang et al., 2026a). Second, recent 4D VAE reduce the cost of modeling these dense 3D Flow targets by compressing point maps and scene flow into compact latent space (Zhu et al., 2026). Like video latents, these flow latents allow 3D Flow to serve as a generative target for joint diffusion modeling.

We use 3D Flow to complement video prediction and provide a bridge between video and action — video provides semantics and appearance, while 3D Flow captures geometry and dynamics. We instantiate this design in **3DFlowWAM**. We first extract 3D Flow targets offline from RGB videos and encode them with a frozen 3D Flow VAE. We then interleave the resulting tokens with video and action tokens in a shared diffusion transformer, where action queries attend to both video and 3D Flow. During training, the model jointly denoises all three modalities within each prediction block, using completed ground-truth blocks as clean context through teacher forcing. At inference, the model jointly generates video, 3D Flow, and actions without requiring the offline tracker or a depth sensor. Our training pipeline consists of two stages: 3D Flow-aware mid-training and task-specific post-training. Starting from the pretrained Wan video model (Wan Team et al., 2025), we perform mid-training on a moderate-scale 3D Flow corpus ($\sim$1,300 hours; Sec. 3.3). We then adapt the model to target embodiments and tasks through post-training with precise action supervision.

Existing VLAs and WAMs use different training datasets and procedures, making it difficult to attribute performance gain to a particular design. To assess the contribution of 3D Flow, we use DreamZero (Ye et al., 2026) as a common backbone and compare four settings: video–action learning and video–flow–action co-training, each with and without additional mid-training (Sec. 4.2). Within each pair, the mid and post training data are matched. These comparisons examine both the gains from 3D Flow co-training and the further benefits of 3D Flow-aware mid-training before target-task adaptation. We evaluate these settings on LIBERO-Plus (Fei et al., 2026), RoboTwin 2.0 (Chen et al., 2026b), and seven real-world tasks on two bimanual platforms: ALOHA-style (Zhao et al., 2023) dual AgileX Piper and dual Franka Emika Panda. With matched mid-training data, 3DFlowWAM improves average real-world success over the video–action baseline by **21.3** and **22.2** percentage points on ALOHA and Franka, respectively.

Our contributions are:

- We formulate 3D Flow as a video-aligned representation of geometry and dynamics without relying on embodiment-specific action coordinates.
- We develop 3DFlowWAM to jointly generate video, 3D Flow, and actions, building on pre-trained video priors through 3D Flow-aware mid-training and task-specific post-training.
- Through controlled comparisons, we show that 3D Flow co-training improves manipulation performance and that 3D Flow-aware mid-training further benefits real-world adaptation.

2 Related Work

2D and 3D vision–language–action and world action models. VLAs typically adapt vision–language backbones, whereas WAMs build control on video-generative priors. Both families range from RGB-based prediction to explicit geometric supervision. UniVLA learns latent actions, while WorldVLA and VLA-JEPA predict future images and visual features, respectively (Bu et al., 2025b; Cen et al., 2025; Sun et al., 2026). Among WAMs, DreamZero jointly predicts video and actions,

and Motus learns a shared visual–action latent (Ye et al., 2026; Bi et al., 2026). Fast-WAM further studies the control benefit of video co-training (Yuan et al., 2026b). These approaches exploit visual dynamics without explicitly supervising metric displacement. TraceVLA and FlowVLA introduce image-plane traces or optical flow (Zheng et al., 2025; Zhong et al., 2025). FlowWAM predicts RGB and optical-flow plans for an action expert (Chen et al., 2026c). Such 2D motion retains image correspondence but leaves depth-dependent displacement ambiguous.

Geometry-aware policies expose additional spatial structure. SpatialVLA encodes back-projected 3D positions, while LingBot-VLA and Track4Action distill depth and world-centric tracking features into policies (Qu et al., 2025; Wu et al., 2026; Wang et al., 2026a). LaMP conditions an action expert on partially denoised image-plane and depth trajectories from a frozen motion expert (Wang et al., 2026b). For WAMs, X-WAM predicts future RGB-D, whereas WAM4D and MECo-WAM use training-time geometric supervision (Guo et al., 2026; Li et al., 2026; Zhang et al., 2026b). In contrast, we jointly predict video, image-aligned Point Maps, world-centric Scene Flow, and actions. Point Maps represent the 3D scene state, while Scene Flow represents its temporal change; RGB prediction preserves semantic and appearance information. Appendix A compares the closest interfaces.

Dense geometry and 3D tracking. DUSt3R recovers image-aligned point maps, and SpatialTracker and DELTA lift correspondence into 3D (Wang et al., 2024; Xiao et al., 2024; Ngo et al., 2025). MotionCrafter compresses point maps and scene flow with a 4D VAE, while Track4World predicts world-centric geometry and correspondence from video (Zhu et al., 2026; Lu et al., 2026). We use these advances to construct offline 3D Flow targets from real demonstrations for midtraining and adaptation. The targets preserve RGB correspondence without requiring shared action coordinates across robots. Deployment needs neither a tracker nor measured depth.

3 METHOD

3.1 WORLD-CENTRIC 3D FLOW FORMULATION

Let H denote the common prediction horizon. The inputs are a language instruction ℓ, an RGB frame I_0, and an optional proprioceptive state s_0, where s_t denotes the robot proprioceptive state at time t. 3DFlowWAM predicts future RGB observations $I_{1:H}$, robot actions $a_{1:H}$, and dense 3D Flow targets $M_{0:H-1}$. At each transition $t \in \{0, \dots, H-1\}$, the target is $M_t = (P_t, F_t^{\Delta})$, with $P_t, F_t^{\Delta} \in \mathbb{R}^{h \times w \times 3}$. Here, h and w denote the spatial grid height and width. With model parameters θ, the joint predictive distribution is $p_\theta(I_{1:H}, M_{0:H-1}, a_{1:H} \mid I_0, s_0, \ell)$. Here, M describes where scene points are in 3D and how they move, providing spatial information for action prediction. Figure 2 summarizes the framework.

Representation. For a visible pixel $p = (u, v)$ at time t, the point map $\mathbf{P}_t(p) \in \mathbb{R}^3$ stores its 3D position in a coordinate system shared across the sequence. Let Δ denote the temporal stride and $q_{t+\Delta}(p)$ the 3D position of the same physical point after that stride. The Scene Flow vector $\mathbf{F}_t^{\Delta}(p) = q_{t+\Delta}(p) - \mathbf{P}_t(p)$ records its *3D displacement*. We use *3D Flow* for the joint target $(\mathbf{P}_t, \mathbf{F}_t^{\Delta})$: Point Map + Scene Flow. Point Maps encode the current 3D geometry and Scene Flow encodes 3D motion, both on the RGB image grid to preserve spatial correspondence. We express them in the simulator world frame or first-camera frame. Following MotionCrafter (Zhu et al., 2026), we normalize both fields with a shared sequence-level center and scale, and denote the normalized Point Map and Scene Flow by $\bar{P}_t$ and $\bar{F}_t^{\Delta}$, respectively.

Simulation targets. LIBERO (Liu et al., 2023) and RoboTwin (Chen et al., 2026b) expose the depth $D_t(p)$, camera intrinsics K_{cam}, and rigid-body transforms. Let $T_{c,t}^W$ and $T_{b,t}^W$ map camera c and rigid body b to the simulator world frame W, respectively. Let π^{-1} denote calibrated depth unprojection. We construct $\mathbf{P}_t(p) = T_{c,t}^W \pi^{-1}(p, D_t(p); K_{\text{cam}})$. We then propagate the point with its associated body: $q_{t+\Delta}(p) = T_{b,t+\Delta}^W (T_{b,t}^W)^{-1} \mathbf{P}_t(p)$. 3D flow generated from privileged simulator state is used only for training supervision.

Real-video pseudo-labels. For real demonstrations in midtraining and ALOHA/Franka posttraining, we use Track4World (Lu et al., 2026) to estimate 3D flow from RGB videos. Specifically, Track4World recovers per-pixel 3D geometry and tracks the same physical points across frames. We

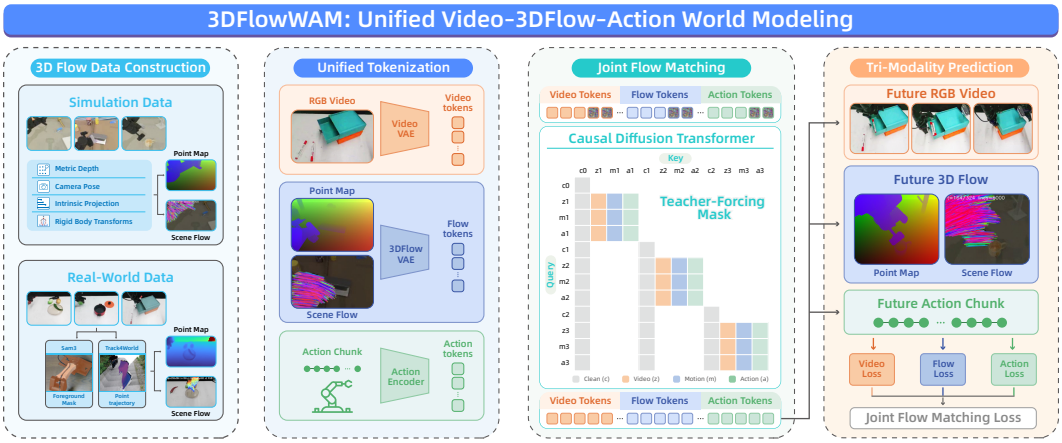

Figure 2: **3DFlowWAM jointly learns to predict video, 3D Flow, and actions in a single causal generative model.** We construct image-aligned 3D Flow targets offline from simulation trajectories or 3D tracking of real videos. Frozen video and 3D Flow VAEs, together with an action encoder, tokenize the three modalities. A shared diffusion transformer jointly denoises their tokens within each prediction block, using completed ground-truth RGB videos as clean context during training. Modality-specific flow-matching losses supervise RGB video, 3D Flow, and actions.

transform the recovered points into a coordinate system defined by the first camera frame to obtain point maps, and compute scene flow from the 3D motion of corresponding points. Since tracking is less reliable in distant background regions, we retain scene-flow supervision only where both tracker confidence and a SAM 3 foreground mask are valid (Carion et al., 2026). Section 5.4 shows that this foreground filtering improves real-world manipulation performance.

3.2 Unified Tokenization and Joint Causal Flow Matching

We tokenize video, action, and 3D Flow as $z^V = E_V(I_{0:H})$, $z^A = E_A(a_{1:H})$, and $z_t^M = [E_P(\bar{\mathbf{P}}_t); E_F(\bar{\mathbf{F}}_t^{\Delta})]$, using the frozen video VAE E_V, learned action encoder E_A, and frozen 3D Flow VAE Point Map/Scene Flow branches E_P, E_F, respectively. The 3D Flow tokens remain spatially aligned with video tokens, while language and robot state serve as conditioning inputs.

We organize prediction into N causal blocks, where t_k denotes the prediction boundary of block k and $S_k = [z_k^V; z_k^M; z_k^A]$ contains the video, 3D Flow, and action tokens of that block. Within each block, the three modalities are jointly denoised by shared transformer layers, allowing action tokens to directly access both RGB videos and 3D Flow. Each block models $p_\theta(\mathcal{S}_k \mid I_{\le t_k}, \ell, s_{t_k})$, using the RGB history available up to t_k.

For each modality $m \in \{V, M, A\}$, let z^m denote its clean latent representation. Given Gaussian noise $\epsilon^m \sim \mathcal{N}(0, I)$ and noise level $\sigma \in [0, 1]$,

$$z_\sigma^m = (1-\sigma)z^m + \sigma\epsilon^m, \qquad v^{m,*} = \epsilon^m - z^m. \tag{1}$$

The joint training objective is

$$\mathcal{L} = \sum_{m\in\{V,M,A\}} \lambda_m \mathbb{E}\big[w(\sigma)\|v_\theta^m - v^{m,*}\|_2^2\big], \tag{2}$$

where v_θ^m, $w(\sigma)$, and λ_m denote the predicted velocity, weighting function, and modality weight.

Causal teacher forcing. Many manipulation tasks involve multiple stages, making historical observations important for identifying the current stage of task execution. Following DreamZero, we encode historical RGB observations as clean context for predicting the next video, 3D Flow, and action tokens. To enable this history-conditioned prediction, we use the causal mask shown in Fig. 2, allowing each prediction block to attend to all available RGB history while preventing access to future observations. During training, the clean history is taken from the ground-truth trajectory; at inference, it consists of RGB observations collected during execution. After each action chunk is executed, newly observed frames are appended to the context for the next prediction.

3.3 3D Flow-aware midtraining

We extend Wan's video prior (Wan Team et al., 2025) with a 3D Flow-aware midtraining stage before target-task adaptation. We augment action-aligned videos with the 3D Flow targets defined in Section 3.1. Each trajectory provides RGB observations and robot actions, augmented with Point Map and Scene Flow targets from Track4World and SAM 3. This stage aligns RGB appearance with 3D Flow before adaptation to limited target-task demonstrations. Action prediction remains conditioned on the source embodiment.

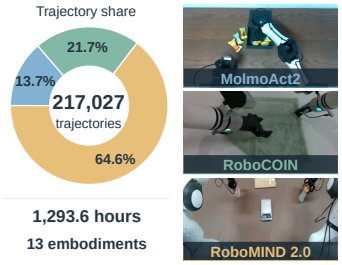

Figure 3: **Midtraining corpus.**

After quality filtering, the corpus contains 217,027 trajectories and 139.7M action-aligned time steps (Figure 3). Training across diverse scenes and embodiments allows 3D flow to serve as a shared bridge between visual dynamics and different action spaces, facilitating downstream adaptation.

4 Experiment

4.1 Benchmarks

Simulation benchmarks. We evaluate four ten-task suites on LIBERO (Liu et al., 2023) and seven observation and initialization perturbations on LIBERO-Plus (Fei et al., 2026). We also test all 50 bimanual RoboTwin 2.0 tasks under clean and randomized conditions (Chen et al., 2026b). Success rates (%) are averaged over suites for LIBERO, over 10,030 evaluation episodes for Plus, and over tasks for RoboTwin. LIBERO-Plus is only used for evaluation. We train on all four LIBERO suites for 20K updates and jointly on clean and randomized RoboTwin demonstrations for 50K updates.

Real-world benchmarks. We evaluate seven bimanual tasks: four on two AgileX Piper arms (ALOHA-style (Zhao et al., 2023)) and three on a dual-Franka Emika Panda system (Fig. 4). For each platform, we jointly train a single policy on all its tasks and evaluate the same policy on all four ALOHA tasks or all three Franka tasks. Both platforms use an overhead camera and two wrist-mounted cameras. Because wrist-mounted cameras may move rapidly or have their views occluded by manipulated objects, we restrict 3D Flow tracking, supervision, and generation to the overhead view. The main benchmark uses 50 demonstrations per task (52 for Franka stacking), with 20 ALOHA or 30 Franka trials per policy and task. Success requires completing the full instruction, and task means are reported separately by platform. We further evaluate two tasks' OOD generalization: changes in towel size and color for towel folding, and shifts in the range of initial pen positions for placing pens into a drawer (Fig. 10). Appendix B.2 gives the task criteria and protocols.

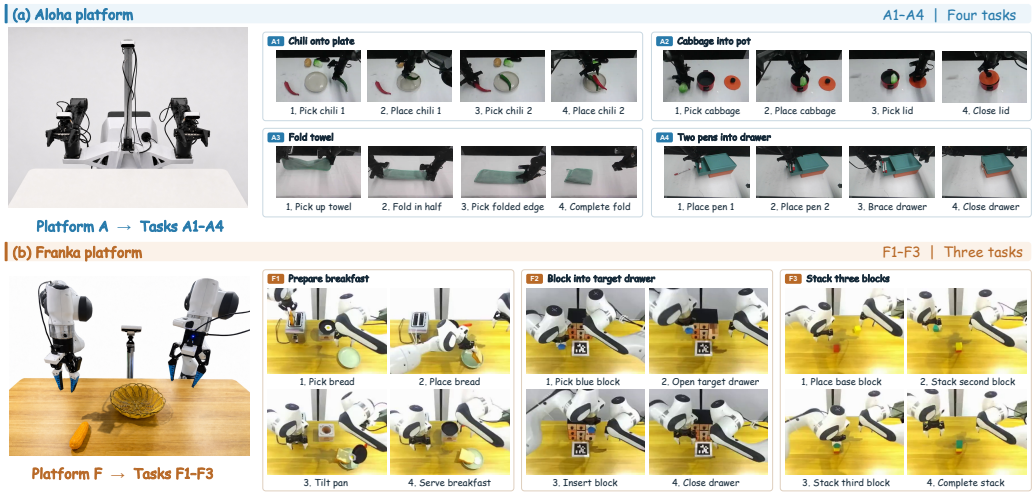

Figure 4: **Real-world platforms and tasks. (a)** ALOHA A1–A4. **(b)** Franka F1–F3.

4.2 Model–Training Setups

Among existing World Action Model (WAM) architectures, we choose DreamZero (Ye et al., 2026) as our base model. We evaluate the role of 3D Flow and midtraining through four model–training setups (Fig. 5).

(1) DreamZero uses the original DreamZero architecture and its pretrained weights, without our additional midtraining, and learns only video and action during post-training. **(2) 3DFlowWAM w/o midtrain** uses our architecture, which modifies DreamZero by introducing a dedicated 3D Flow module. It skips midtraining and jointly learns video, 3D Flow, and action during post-training. **(3) DreamZero*** retains the original DreamZero architecture and undergoes video–action midtraining on 1,294 hours of real-world robot data (Section 3.3), followed by post-training with the same video–action objectives. **(4) 3DFlowWAM** is our full model, which jointly learns video, 3D Flow, and action during both midtraining and post-training.

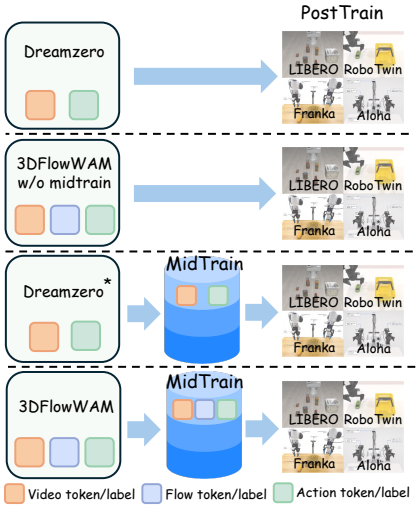

Figure 5: **Overview of the four model–training setups.**

Comparing (1) with (2), or (3) with (4), isolates the benefit of incorporating 3D Flow under the corresponding training setting. Comparing (2) with (4) assesses the benefit of 3D Flow-aware midtraining. We follow the four training paths in Fig. 5, comparing DreamZero with 3DFlowWAM w/o midtrain and DreamZero* with midtrained 3DFlowWAM.

4.3 Simulation Benchmark Results

LIBERO-Plus. 3DFlowWAM achieves top-tier performance on LIBERO and the highest LIBERO-Plus score among the compared WAMs at 87.7% (Tab. 1). Full results for each LIBERO-Plus perturbation dimension are provided in Appendix F (Tab. 8). Since performance on standard LIBERO is already saturated, we focus primarily on LIBERO-Plus, where observation and initialization perturbations test generalization under distribution shifts. 3DFlowWAM exceeds RGB-based DreamZero* (77.4%) and Cosmos Policy (Kim et al., 2026) (82.2%), supporting explicit spatial supervision beyond RGB prediction. With or without midtraining, 3DFlowWAM outperforms DreamWAM (Yuan et al., 2026a), another 3D-aware WAM that augments optical flow with depth and semantic supervision, on LIBERO-Plus overall. By encouraging the model to learn about 3D positions and displacements, explicit 3D Flow modeling may offer more effective spatial supervision.

Table 1: **LIBERO (2023) and LIBERO-Plus (2026) success (%).** Bold/underline: best/second-best;

Type	Method	LIBERO Avg.	LIBERO-Plus Total
I	OpenVLA-OFT (2025)	97.1	69.6
	π_0-FAST (2025)	85.5	61.6
	$\pi_{0.5}$ (2025)	96.9	<u>85.7</u>
II	UniVLA (2025b)	95.2	42.9
	VLA-JEPA (2026)	97.2	77.9
III	HoloBrain-0-GD (2026b)	96.7	74.0
	LaMP (2026b)	98.3	79.3
IV	Fast-WAM (2026b)	97.6	50.0
	DC-WAM (2026)	98.1	60.9
	DreamZero* (2026)	96.1	77.4
	Cosmos Policy (2026)	98.5	82.2
	ABot-M0.5 (2026a)	**99.4**	83.4
V	Being-H0.7 (2026)	<u>99.2</u>	82.1
VI	ImageWAM (4B) (2026c)	98.4	83.1
	ImageWAM (9B)	–	85.2
VII	DreamWAM (2026a)	98.9	74.6
VIII	**3DFlowWAM w/o midtrain**	97.2	78.1
	3DFlowWAM w/ midtrain	98.2	**87.7**

Paradigms. **I**: 2D VLA: RGB–language policies. **II**: 2D VLA: latent visual / action representations. **III**: 3D-aware VLA: geometric encoding / motion priors. **IV**: 2D WAM: video prediction / latent-action modeling. **V**: 2D WAM: latent world prediction. **VI**: 2D WAM: image-editing prediction. **VII**: 3D-aware WAM: RGB / optical flow with depth-feature supervision. **VIII**: 3D WAM: RGB + 3D Flow.

RoboTwin. We compare WAMs both without and with midtraining (Tab. 2). Without midtraining, 3DFlowWAM outperforms the corresponding Motus (Bi et al., 2026) and FlowWAM (Chen et al., 2026c) models under both clean and randomized conditions. Among models that explicitly predict motion, FlowWAM uses optical flow without explicit 3D modeling, which should explain our higher success rates in this setting. With midtraining, we compare our

Table 2: **RoboTwin 2.0 (2026b) success (%).** Mean over all 50 tasks. Task results: Appendix E.

Type	Method	Clean	Rand.
I	GO-1 (2025a)	37.80	36.24
	$\pi_{0.5}$ (M) (2025)	42.98	43.84
	$\pi_{0.5}$ (L)	82.74	76.76
	X-VLA (2026)	72.80	72.84
	LingBot-VLA w/o depth (2026)	86.50	85.34
III	LingBot-VLA w/ depth	88.56	86.68
IV	Motus w/o pretrain (2026)	77.56	77.00
	Motus w/ pretrain	88.66	87.02
	DreamZero* (2026)	74.30	72.94
	FlowWAM w/o pretrain (2026c)	82.40	80.80
IX	X-WAM (2026)	89.80	**90.70**
VIII	**3DFlowWAM w/o midtrain**	89.76	88.54
	3DFlowWAM w/ midtrain	**91.48**	89.90

Paradigms. Numbering follows Table 1. **I**: 2D VLA. **III**: 3D-aware VLA. **IV**: 2D WAM. **VIII**: 3D WAM: RGB + 3D Flow. **IX**: 3D WAM: RGB-D prediction.

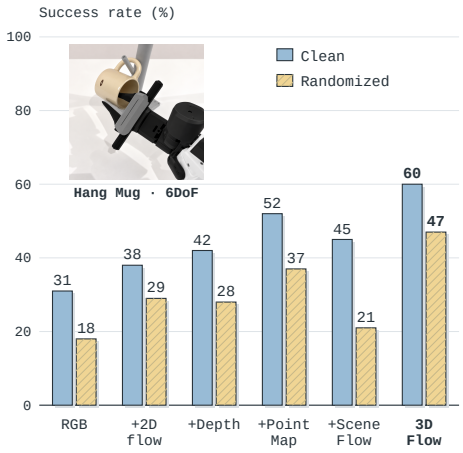

Figure 6: **3D representation design on Hang Mug.** All variants retain RGB; 3D Flow combines Point Maps and Scene Flow.

full model with pretrained Motus. This comparison uses different midtraining corpora and recipes, whose scale and composition are difficult to match across models. Although Motus uses approximately five times as many midtraining trajectories as our model, 3DFlowWAM still achieves higher success rates under both clean and randomized conditions. Within our own model, however, midtraining yields smaller gains on RoboTwin than on LIBERO-Plus. We hypothesize that joint post-training on clean and randomized demonstrations already enables strong adaptation to the evaluated distributions, leaving less room for midtraining to improve generalization.

4.4 Real-Robot Benchmark Results

Table 3: **Real-world success (%).** ALOHA/Franka: 20/30 trials per task. Bold/underline: best/second-best across all five methods. Baselines: DreamZero (Ye et al., 2026) and $\pi_{0.5}$ (Black et al., 2025).

	ALOHA					Franka			
Method	Chilies	Cabbage	Towel	Pens	Avg.	Breakfast	Drawer	Blocks	Avg.
w/o mid-training									
DreamZero	40.0	10.0	30.0	40.0	30.0	26.7	13.3	40.0	26.7
3DFlowWAM	50.0	30.0	60.0	50.0	47.5	33.3	26.7	46.7	35.6
mid-trained									
$\pi_{0.5}$	70.0	**60.0**	70.0	60.0	65.0	**93.3**	73.3	60.0	75.6
DreamZero*	60.0	**60.0**	50.0	60.0	57.5	73.3	60.0	46.7	60.0
3DFlowWAM	**80.0**	**60.0**	**85.0**	**90.0**	**78.8**	**93.3**	**80.0**	**73.3**	**82.2**

Note. We group $\pi_{0.5}$ under mid-trained models because its training includes extensive non-public bimanual ALOHA manipulation data, which we consider to provide strong prior alignment with this platform.

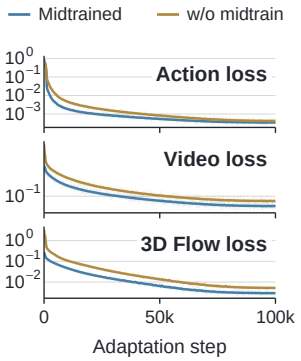

Figure 7: **Adaptation losses on real-world post-train.**

3DFlowWAM outperforms the corresponding video–action baseline on both platforms: DreamZero without midtraining and DreamZero* (with video-action midtraining) (Tab. 3). Our midtraining corpus (Section 3.3) consists primarily of bimanual manipulation data, exposing the model to viewpoints and motion patterns relevant to both the ALOHA and Franka platforms. Consistent with this, midtraining substantially improves success rates on both platforms (Tab. 3). $\pi_{0.5}$ (Black et al., 2025) also performs well without our midtraining corpus, likely benefiting from the large-scale ALOHA-style manipulation data already included in its pretraining.

During midtraining, DreamZero* and 3DFlowWAM use the same video and action data, but 3DFlowWAM additionally predicts 3D Flow, providing inherent supervision for scene-point posi-

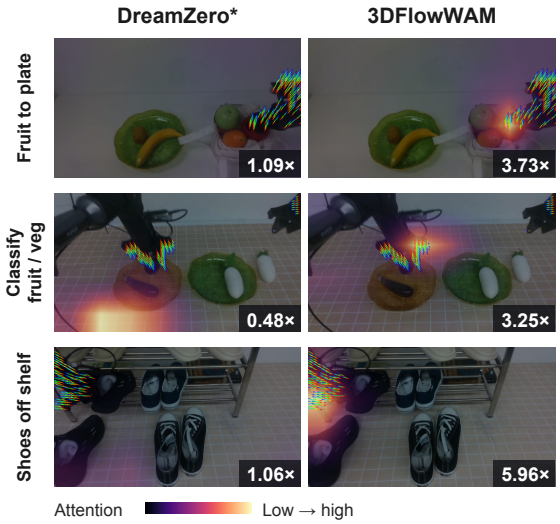

Figure 8: **3D Flow promotes interaction-focused visual attention.** Colored traces indicate predicted Scene Flow. × measures attention in moving regions relative to the image-wide mean; higher values indicate stronger interaction-focused attention. Paired views use the same attention settings.

Table 4: **Chunk inference rate (Hz).** Δt denotes the relative latency increase of 3DFlowWAM over DreamZero.

Steps / Forwards	DreamZero	3DFlowWAM	Δt
4 / 4	6.72	6.60	+1.82%
16 / 8	3.82	3.65	+4.66%

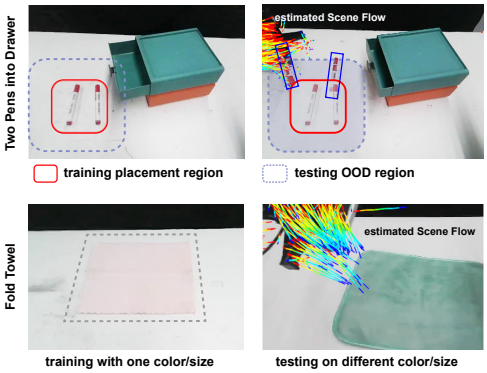

Figure 9: **Scene Flow in new scenes.** Predicted Scene Flow on observed RGB. Red: training placement region; dashed blue: test placement extent.

tions and displacements. By grounding where objects are and how they move in 3D, this additional supervision may help the policy translate visual goals into precise placement actions. The gains on Franka drawer manipulation and block stacking support this interpretation, since both tasks require aligning an object with a receptacle or another object.

To examine how midtraining affects downstream adaptation, we compare ALOHA post-training runs with and without midtraining (Fig. 7). Using the same target data and 100K post-training updates, the midtrained model achieves lower action, video, and 3D Flow losses. The reduction in action loss suggests that midtraining benefits action prediction as well as visual and geometric prediction, consistent with the higher task success rates in Tab. 3. Appendix B.3 provides the training setup and loss statistics.

5 Analysis

5.1 3D Flow Representation

Task and protocol. We evaluate different 3D representations on RoboTwin's challenging *Hang Mug* task (Chen et al., 2026b), which requires precise 6DoF manipulation and provides a strong test of 3D spatial reasoning. All variants retain RGB and action prediction (Fig. 6). Depth and 2D optical flow use the Wan VAE (Wan Team et al., 2025), whereas Point Maps and Scene Flow use separate 3D Flow VAE branches (Zhu et al., 2026) that can encode either field independently. The full *3D Flow* variant combines both fields.

Point Maps and Scene Flow are complementary. Point Maps specify where points are in 3D, while Scene Flow describes how they move. Scene Flow alone outperforms optical flow in clean scenes but performs worse under randomization (21% vs. 29%), suggesting that motion without explicit 3D location is insufficient for robust spatial reasoning. Combining Point Maps with Scene Flow provides both 3D location and motion, outperforming either representation alone in both settings. These results support our joint 3D Flow representation over Point Maps or Scene Flow alone.

5.2 Visual Grounding and Generalization

3D Flow improves visual grounding. Fig. 8 compares DreamZero* (Ye et al., 2026) and 3DFlowWAM on three midtraining examples under the same observations and heat-map scale. With 3D Flow, action-to-video attention concentrates more strongly on interaction regions. These ex-

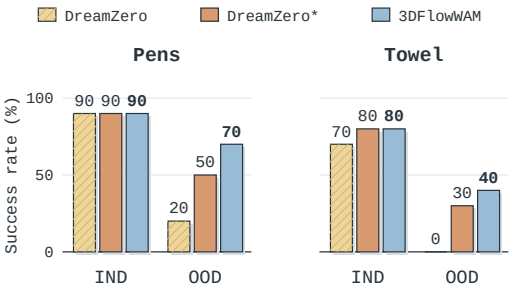

Figure 10: **IND/OOD success.** Results of three models on 2 tasks, with 10 trials per task.

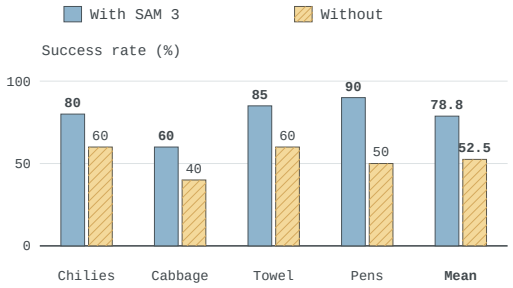

Figure 11: **SAM 3 foreground filtering.** Success over 20 trials per task.

amples suggest that 3D Flow helps action prediction better ground on interaction-relevant visual information.

3D Flow improves OOD generalization. We adapt the models on a subset of in-domain (IND) demonstrations and evaluate OOD generalization to unseen placements and appearances, as illustrated in Fig. 9. Although embodied midtraining substantially improves OOD success (20%→50% on pens and 0%→30% on towels), 3D Flow provides further gains, reaching 70% and 40%, respectively (Fig. 10). Notably, DreamZero* and 3DFlowWAM achieve identical IND success, suggesting that the benefit of 3D Flow emerges specifically under distribution shifts. These results show that explicit 3D geometry and motion further improve generalization beyond demonstrated scenes.

5.3 Inference overhead

On an NVIDIA H100 with 4 denoising steps, adding 3D Flow introduces only 1.82% inference latency overhead (Tab. 4). Repeated DiT evaluations dominate inference cost, consistent with DreamZero, making the additional sequence length relatively inexpensive. Moreover, 3D Flow uses less than half the video spatial resolution and fewer than one quarter as many tokens, while sharing the same denoising loop and requiring no online tracker.

5.4 Foreground filtering

SAM 3 foreground filtering (Carion et al., 2026) reduces noisy background Scene Flow, increasing average success from 52.5% to 78.8% (Fig. 11). Distant regions typically have larger depth uncertainty, and even small camera-pose errors can produce large spurious motion in the background. Since our third-person cameras are fixed, most background regions should have near-zero Scene Flow. When accurate full-scene Scene Flow cannot be reliably recovered, we therefore use SAM 3 to segment the foreground and set background Scene Flow to zero, providing a closer approximation to the true 3D flow while preserving foreground motion that can be faithfully reconstructed by the 3D Flow VAE. We additionally verify that MotionCrafter's frozen 4D VAE effectively encodes and reconstructs 3D Flow with zeroed background, suggesting that foreground filtering does not disrupt its learned latent space.

6 Discussion

Scaling with better tracking and latent modeling. The benefits of 3D Flow may grow as 4D tracking and latent modeling improve. Advances in 4D tracking could provide cleaner training data for 4D VAEs. Faster extraction and more efficient latent modeling could further lower computational costs. Together, these advances could improve both the scalability of 3D Flow supervision and its contribution to action learning. Our results offer an initial step toward this direction.

Learning from human egocentric videos. Another future work is learning from human egocentric videos. Modeling camera poses and 3D point trajectories in a shared reference frame can disentangle camera ego-motion from scene motion. This could allow 3D Flow to capture interaction dynamics across moving viewpoints and support transfer from human videos to robot control.

AI USE DISCLOSURE

Large language models were used solely for language editing and improving the readability of the manuscript. They were not used for research methodology, experimental design or analysis, or scientific claims. All AI-assisted edits were reviewed by the authors, who take full responsibility for the final manuscript.

ETHICS STATEMENT

Our experiments are conducted in simulation and controlled robotic laboratory environments and involve no human participants or personal data.

REPRODUCIBILITY STATEMENT

We provide detailed implementation and experimental settings to facilitate reproduction of our results. The supplementary material includes the model code and videos documenting real-world policy execution. Section 3 describes the 3D Flow representation, data construction, and training objectives, while Section 4.1 specifies the training and evaluation settings. Appendix B provides details of midtraining, target adaptation, and real-world evaluation protocols. Appendices E and F report complete RoboTwin and LIBERO/LIBERO-Plus results, respectively.

REFERENCES

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

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Rich Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: A vision-language-action model with open-world generalization. In *Proceedings of the 9th Conference on Robot Learning (CoRL 2025)*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 2025.

Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xu Huang, Shu Jiang, et al. AgiBot World Colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3549–3556, 2025a.

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

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Radle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollar, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. SAM 3: Segment anything with concepts. In *International Conference on Learning Representations*, pp. 138846–138923, 2026.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, Deli Zhao, and Hao Chen. WorldVLA: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025.

Ronghan Chen, Yandan Yang, Zuojin Tang, Dongjie Huo, Tong Lin, Haoning Wu, Haoyun Liu, Yuzhi Chen, Lulu Zheng, Botai Yuan, Tianlun Li, Mingxin Wang, Dekang Qi, Bin Hu, Wei Mei, Yuze Xuan, Haolong Yang, Yanqing Zhu, Mu Xu, Zhiheng Ma, and Xinyuan Chang. ABot-M0.5: Unified mobility-and-manipulation world action model. *arXiv preprint arXiv:2607.00678*, 2026a.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan-ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. In *International Conference on Machine Learning*, 2026b.

Yixiang Chen, Peiyan Li, Yuan Xu, Qisen Ma, Jiabing Yang, Kai Wang, Jianhua Yang, Dong An, He Guan, Gaoteng Liu, Jianlou Si, Jun Huang, Jing Liu, Nianfeng Liu, Yan Huang, and Liang Wang. FlowWAM: Optical flow as a unified action representation for world action models. *arXiv preprint arXiv:2607.13017*, 2026c.

Haoquan Fang, Jiafei Duan, Donovan Clay, Sam Wang, Shuo Liu, Weikai Huang, Xiang Fan, Wei-Chuan Tsai, Shirui Chen, Yi Ru Wang, Shanli Xing, Jaemin Cho, Jae Sung Park, Ainaz Eftekhar, Peter Sushko, Karen Farley, Angad Wadhwa, Cole Harrison, Winson Han, Ying-Chun Lee, Eli VanderBilt, Rose Hendrix, Suveen Ellawela, Lucas Ngoo, Joyce Chai, Zhongzheng Ren, Ali Farhadi, Dieter Fox, and Ranjay Krishna. MolmoAct2: Action reasoning models for real-world deployment. *arXiv preprint arXiv:2605.02881*, 2026.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: A progressive robustness benchmark for visual-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 38574–38583, June 2026.

Jun Guo, Qiwei Li, Peiyan Li, Zilong Chen, Nan Sun, Yifei Su, Heyun Wang, Yuan Zhang, Xinghang Li, and Huaping Liu. Unified 4d world action modeling from video priors with asynchronous denoising. *arXiv preprint arXiv:2604.26694*, 2026.

Chengkai Hou et al. RoboMIND 2.0: A multimodal, bimanual mobile manipulation dataset for generalizable embodied intelligence. *arXiv preprint arXiv:2512.24653*, 2025.

Siqiao Huang, Jialong Wu, Qixing Zhou, Shangchen Miao, and Mingsheng Long. Vid2world: Crafting video diffusion models to interactive world models. In *International Conference on Learning Representations*, pp. 10763–10806, 2026.

Haoyuan Ji, Lingxiang Fan, Shang Su, Yinqiao Lu, Mengkai Shi, Jun Gao, and Shuo Feng. DC-WAM: Dynamic-centric visual supervision and reasoning for world-action models. *arXiv preprint arXiv:2607.25918*, 2026.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *Proceedings of Robotics: Science and Systems*, 2025.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning. In *International Conference on Learning Representations*, pp. 71531–71552, 2026.

Ying Li, Xiaobao Wei, Jiajun Cao, Hao Wang, Xiaowei Chi, Chengyu Bai, Qianpu Sun, Jiajun Li, Xiaojie Zhang, Jian Tang, Sirui Han, and Shanghang Zhang. WAM4D: Fast 4d world action model via spatial register tokens. *arXiv preprint arXiv:2606.14048*, 2026.

Haitao Lin, Hanyang Yu, Jingshun Huang, He Zhang, Yonggen Ling, Ping Tan, Xiangyang Xue, and Yanwei Fu. Universal pose pretraining for generalizable vision-language-action policies. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2026a.

Xuewu Lin, Tianwei Lin, Yun Du, Hongyu Xie, Yiwei Jin, Jiawei Li, Shijie Wu, Qingze Wang, Mengdi Li, Mengao Zhao, Ziang Li, Chaodong Huang, Hongzhe Bi, Lichao Huang, and Zhizhong Su. HoloBrain-0 technical report. *arXiv preprint arXiv:2602.12062*, 2026b.

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

Jiahao Lu, Jiayi Xu, Wenbo Hu, Ruijie Zhu, Chengfeng Zhao, Sai-Kit Yeung, Ying Shan, and Yuan Liu. Track4world: Feedforward world-centric dense 3d tracking of all pixels. In *European Conference on Computer Vision*, pp. 184–204. Springer, 2026.

Hao Luo, Wanpeng Zhang, Yicheng Feng, Sipeng Zheng, Haiweng Xu, Chaoyi Xu, Ziheng Xi, Yuhui Fu, and Zongqing Lu. Being-H0.7: A latent world-action model from egocentric videos. *arXiv preprint arXiv:2605.00078*, 2026.

Tuan Duc Ngo, Peiye Zhuang, Chuang Gan, Evangelos Kalogerakis, Sergey Tulyakov, Hsin-Ying Lee, and Chaoyang Wang. DELTA: Dense efficient long-range 3d tracking for any video. In *International Conference on Learning Representations*, pp. 44620–44641, 2025.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient action tokenization for vision-language-action models. In *Proceedings of Robotics: Science and Systems*, 2025.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Jiayuan Gu, Zhigang Wang, Yan Ding, Bin Zhao, Dong Wang, and Xuelong Li. SpatialVLA: Exploring spatial representations for visual-language-action models. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025.

Kanchana Ranasinghe, Honglu Zhou, Yu Fang, Luyu Yang, Le Xue, Ran Xu, Caiming Xiong, Silvio Savarese, Michael S. Ryoo, and Juan Carlos Niebles. Future optical flow prediction improves robot control and video generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Findings*, 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.

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

Chenyi Wang, Xinkai Wang, Bokai Lin, Jialin Tian, Fucheng Zhang, Cewu Lu, and Lixin Yang. Track4Action: Distilling world-centric 3d tracker into vision-language-action policies. *arXiv preprint arXiv:2608.03727*, 2026a.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20697–20709, 2024.

Xinkai Wang, Chenyi Wang, Yifu Xu, Mingzhe Ye, Fucheng Zhang, Jialin Tian, Xinyu Zhan, Lifeng Zhu, Cewu Lu, and Lixin Yang. LaMP: Learning vision-language-action policy with 3d scene flow as latent motion prior. In *European Conference on Computer Vision*, pp. 227–244. Springer, 2026b.

Shihan Wu et al. RoboCOIN: An open-sourced bimanual robotic data collection for integrated manipulation. *arXiv preprint arXiv:2511.17441*, 2025.

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

Yuxi Xiao, Qianqian Wang, Shangzhan Zhang, Nan Xue, Sida Peng, Yujun Shen, and Xiaowei Zhou. SpatialTracker: Tracking any 2d pixels in 3d space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20406–20417, 2024.

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

Shanglin Yuan, Weiheng Zhao, Xin Shi, Haoyi Jiang, Xianda Guo, Liu Liu, Wenyu Liu, Wei Sui, and Xinggang Wang. DreamWAM: Beyond RGB future prediction for world action models. *arXiv preprint arXiv:2608.04996*, 2026a.

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

Chuhan Zhang, Guillaume Le Moing, Skanda Koppula, Ignacio Rocco, Liliane Momeni, Junyu Xie, Shuyang Sun, Rahul Sukthankar, Joëlle K. Barral, Raia Hadsell, Zoubin Ghahramani, Andrew Zisserman, Junlin Zhang, and Mehdi S. M. Sajjadi. Efficiently reconstructing dynamic scenes one D4RT at a time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7382–7392, June 2026a.

Jianjun Zhang, Jian Zhu, Taiyi Su, Chong Ma, Zitai Huang, Yi Xu, and Hanli Wang. Learning 4d geometric priors for inference-efficient world action models. *arXiv preprint arXiv:2607.05468*, 2026b.

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*, 2026c.

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

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Ya-Qin Zhang, Jiangmiao Pang, Jingjing Liu, Tai Wang, and Xianyuan Zhan. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *International Conference on Learning Representations*, pp. 60580–60606, 2026.

Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. TraceVLA: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. In *International Conference on Learning Representations*, pp. 54277–54296, 2025.

Zhide Zhong, Haodong Yan, Junfeng Li, Xiangchen Liu, Xin Gong, Tianran Zhang, Wenxuan Song, Jiayi Chen, Xinhu Zheng, Hesheng Wang, and Haoang Li. FlowVLA: Visual chain of thought-based motion reasoning for vision-language-action models. *arXiv preprint arXiv:2508.18269*, 2025.

Ruijie Zhu, Jiahao Lu, Wenbo Hu, Xiaoguang Han, Jianfei Cai, Ying Shan, and Chuanxia Zheng. Motioncrafter: Dense geometry and motion reconstruction with a 4d vae. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 40303–40315, 2026.

A Representation Taxonomy of the Closest Methods

Table 5 separates what is supervised from what the policy retains at deployment. A 3D-aware VLA can learn geometry without generating a future world state, while a video-backbone WAM can use only 2D motion. Our distinction is an image-aligned 3D Flow stream retained in joint video–3D Flow–action prediction. This distinction goes beyond the presence of a dynamics objective.

Table 5: **How related methods connect scene information to action.** Categories concern explicit intermediate representations, not robot-action dimensionality. Training-only branches can transfer geometry into the deployed policy.

Method / family	Scene or motion representation	Connection to action
SpatialVLA / 3D VLA (2025)	Depth-back-projected 3D positions	Geometry is encoded in visual tokens; no joint future RGB–motion stream.
LingBot-VLA / 3D VLA (2026)	Depth-teacher features	Learned queries receive geometric distillation during policy learning.
HoloBrain-0 / 3D VLA (2026b)	Calibrated camera geometry and robot-link poses	Explicit embodiment-aware geometric conditioning; no joint future RGB–3D-displacement stream.
Track4Action / 3D VLA (2026a)	World-centric 3D tracking features	Action-aligned track queries; the tracking branch is training-only.
LaMP / 3D VLA (2026b)	Image-plane and depth motion	Frozen motion-expert features condition a separate action expert; no future RGB prediction.
Motus / 2D WAM (2026)	Optical-flow-derived latent actions	A mixture of understanding, video, and action experts shares latent motion information.
FlowWAM / 2D WAM (2026c)	RGB and 2D optical-flow videos	Joint video–flow prediction provides a flow plan to the action expert.
X-WAM / 3D WAM (2026)	Future multi-view RGB-D	Dedicated depth generation alongside video and action; asynchronous denoising.
WAM4D / 3D WAM (2026)	Future-depth spatial registers	Geometric supervision during training; spatial branch removed at deployment.
MECo-WAM / 3D WAM (2026b)	VGGT geometric relational features	Training-only 4D expert distills geometric priors into the video–action model.
3DFlowWAM / 3D WAM	Image-aligned Point Maps and world-centric Scene Flow	Retains a generative 3D Flow stream in joint RGB–3D Flow–action denoising.

B Midtraining and Target-Adaptation Details

B.1 Midtraining corpus

Figure 3 summarizes the processed 3D Flow-annotated midtraining corpus. MolmoAct2-BimanualYAM contributes 29,746 trajectories and 636.0 hours (Fang et al., 2026), RoboCOIN contributes 47,050 trajectories and 317.3 hours (Wu et al., 2025), and RoboMIND 2.0 contributes 140,231 trajectories and 340.2 hours (Hou et al., 2025). Together, these datasets cover 13 action-schema configurations and 741 unique instructions. After geometric quality filtering, the corpus contains 217,027 trajectories and 139.7M action-aligned frames. Training samples are drawn uniformly over the concatenated frame index.

Track4World (Lu et al., 2026) provides 3D Flow supervision from the fixed third-person view, while all camera views are retained for RGB prediction and control. Midtraining and target adaptation use the same joint video–3D Flow–action interface, enabling shared 3D Flow learning across diverse scenes and embodiments before adaptation to each target platform.

B.2 Real-world tasks and evaluation protocol

Figure 4 shows the seven real-world tasks on the ALOHA and Franka platforms. The four ALOHA tasks are *Chili onto Plate* (A1), *Cabbage into Pot* (A2), *Fold Towel* (A3), and *Two Pens into Drawer*

(A4). The three Franka tasks are *Prepare Breakfast* (F1), *Block into Target Drawer* (F2), and *Stack Three Blocks* (F3). Success requires completion of the full task, including the final placement, folding, drawer-closing, or stable-stacking stage shown in Figure 4.

The main ALOHA benchmark uses 50 demonstrations per task and 20 evaluation trials per policy and task. Franka uses 50 demonstrations for breakfast and drawer manipulation and 52 for block stacking, with 30 evaluation trials per policy and task. The separate generalization experiment uses 40 IND demonstrations per task (Appendix C). Success rate is computed as $100\, N_{\text{success}}/N_{\text{trials}}$. Track4World provides 3D Flow targets during training, with SAM 3 (Carion et al., 2026) filtering where specified; deployment requires only RGB observations.

Table 3 compares target-only and midtrained policies. DreamZero (Ye et al., 2026) and DreamZero* provide the corresponding video–action baselines, while 3DFlowWAM additionally models 3D Flow. For controlled comparisons, the corresponding DreamZero and 3DFlowWAM variants use the same target demonstrations and adaptation schedule. With midtraining, adding 3D Flow improves mean success from 57.5% to 78.8% on ALOHA and from 60.0% to 82.2% on Franka.

B.3 Optimization-curve protocol

Figure 7 compares ALOHA adaptation with and without 3D Flow-aware midtraining. Both runs use the same target data, global batch size, seed, learning-rate schedule, and 100K-step adaptation budget, with the same pretrained video backbone and frozen 3D Flow VAE (Zhu et al., 2026). Pale curves show training losses sampled every 100 steps, and solid curves show 500-step trailing means. Over the final 10% of training, midtraining reduces action, video, and 3D Flow losses by 18.3%, 15.7%, and 44.3%, respectively. These optimization results are consistent with the downstream improvements in Table 3.

B.4 Foreground-filtering ablation

We ablate SAM 3 foreground filtering during ALOHA target adaptation while keeping the midtraining initialization, demonstrations, model, optimization, Track4World predictions, confidence filtering, and evaluation protocol fixed. Foreground filtering improves average success from 52.5% to 78.8% (Figure 11), indicating that reducing tracker noise provides cleaner Scene Flow supervision for learning the 3D Flow representation.

C Real-world Generalization and Scene Flow Prediction

Two-task generalization protocol. Figure 10 and Table 6 compare target-only DreamZero (Ye et al., 2026), midtrained DreamZero*, and midtrained 3DFlowWAM on the ALOHA pen-and-drawer and towel-folding tasks. All models are adapted on the same 40 in-domain (IND) demonstrations per task. DreamZero skips embodied midtraining, while DreamZero* and 3DFlowWAM use video–action and video–3D Flow–action midtraining, respectively. OOD evaluation shifts the initial pen placement or changes the towel appearance, with OOD conditions excluded from adaptation. Midtraining improves DreamZero's OOD success by 30 percentage points on both tasks, while 3D Flow further improves over DreamZero* by 20 points on pens and 10 points on towels.

Table 6: **Real-world IND/OOD success rates (%).** All models are adapted with the same 40 IND demonstrations per task. DreamZero (Ye et al., 2026) skips embodied midtraining; DreamZero* and 3DFlowWAM include it. OOD changes pen placement or uses a green towel.

	IND		OOD	
Method	Pens	Towel	Pens	Towel
DreamZero	**90**	70	20	0
DreamZero*	**90**	**80**	50	30
3DFlowWAM	**90**	**80**	**70**	**40**

Qualitative Scene Flow visualization. Figure 9 visualizes predicted Scene Flow under the distribution shifts above. We decode the predicted 3D Flow latents with 3D Flow VAE (Zhu et al., 2026) and recover their scale using the corresponding normalization statistics. The red region summarizes

the training placement distribution, while the dashed blue region shows the placement extent of the shifted pen evaluation set. Scene Flow is overlaid on observed RGB for visualization, using the same projection and display scaling across the corresponding panels. No ground-truth Scene Flow is used for these predictions.

D Inference Timing Definitions

Table 4 compares inference with and without 3D Flow on NVIDIA H100 with `torch.compile` and CUDA Graphs enabled for both variants. Latency is measured per action-chunk prediction. With four solver steps and four DiT evaluations, DreamZero and 3DFlowWAM run at 6.72 and 6.60 Hz, corresponding to 1.82% additional latency. With sixteen solver steps and eight DiT evaluations, they run at 3.82 and 3.65 Hz, corresponding to 4.66% additional latency. Thus, the 3D Flow stream adds less than 5% latency under both measured schedules.

E Full RoboTwin Results

Table 7 reports per-task results on all 50 RoboTwin 2.0 tasks. We evaluate 100 episodes per task and setting. Our midtrained configuration uses 50K target updates, 512×512 input, 16-dimensional end-effector actions, 32 denoising steps, and 48 executed actions per query. The final row reports the mean across all 50 tasks.

Table 7: **Full RoboTwin 2.0 results.** Per-task success rates (%) under clean and randomized settings. The final row reports the mean over all 50 tasks. Bold and underline indicate the best and second-best results in each setting. Baselines include $\pi_{0.5}$ (Black et al., 2025), X-VLA (Zheng et al., 2026), and Motus (Bi et al., 2026).

	$\pi_{0.5}$		X-VLA		Motus w/o Pretrain		Motus w/ Pretrain		3DFlowWAM w/o midtrain		3DFlowWAM	
Task	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.
Adjust Bottle	79	83	**100**	99	99	97	89	93	94	**100**	96	**100**
Beat Block Hammer	63	50	92	88	88	90	95	88	78	83	**97**	**97**
Blocks Ranking RGB	43	35	83	83	92	88	99	**97**	**100**	**97**	99	95
Blocks Ranking Size	8	14	67	74	38	50	75	63	76	75	**78**	**81**
Click Alarm Clock	97	93	99	99	**100**	99	**100**	**100**	99	**100**	**100**	99
Click Bell	75	76	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
Dump Bin Big Bin	30	42	79	77	94	96	**95**	91	92	**97**	84	93
Grab Roller	90	89	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	99	**100**
Handover Block	18	19	73	37	34	15	86	**73**	51	33	**87**	70
Handover Mic	28	18	0	0	**98**	95	78	63	79	94	88	**96**
Hanging Mug	3	3	23	27	14	10	38	38	**73**	**50**	60	40
Lift Pot	0	0	99	**100**	90	87	96	99	**100**	99	99	**100**
Move Can Pot	29	27	89	86	43	53	34	74	97	91	**99**	**94**
Move Pill Bottle Pad	33	29	73	71	83	83	**93**	**96**	91	90	85	88
Move Playing Card Away	59	67	93	**98**	50	47	**100**	96	93	92	**100**	92
Move Stapler Pad	16	18	78	73	49	37	83	85	**87**	79	82	**87**
Open Laptop	19	35	93	**100**	89	89	95	91	**97**	99	**97**	98
Open Microwave	35	37	79	71	83	82	**95**	**91**	87	81	85	74
Pick Diverse Bottles	5	3	58	36	53	62	90	**91**	**96**	70	95	86
Pick Dual Bottles	10	6	47	36	58	68	96	90	**100**	86	**100**	**92**
Place A2B Left	62	60	48	49	78	79	88	79	**95**	92	93	**98**
Place A2B Right	62	57	36	36	86	83	91	87	**92**	91	91	**96**
Place Bread Basket	48	56	81	71	73	83	91	**94**	84	91	**92**	84
Place Bread Skillet	38	46	77	67	71	71	86	83	92	**92**	**93**	89
Place Burger Fries	66	70	94	94	95	90	**98**	**98**	93	95	**98**	96
Place Can Basket	19	25	49	52	46	62	**81**	76	67	71	63	**81**
Place Cans Plastic Box	40	47	97	98	96	**99**	98	94	**100**	91	**100**	97
Place Container Plate	71	78	97	95	97	**100**	98	99	98	98	**99**	98
Place Dual Shoes	12	7	79	88	78	80	93	87	85	77	**100**	**98**
Place Empty Cup	75	86	**100**	98	97	97	99	98	**100**	**100**	99	**100**
Place Fan	25	36	80	75	77	85	**91**	**87**	85	84	84	86
Place Mouse Pad	21	26	70	70	62	68	66	68	87	**82**	**90**	**82**
Place Object Basket	43	36	44	39	74	75	81	87	**92**	**89**	82	82
Place Object Scale	40	49	52	74	84	83	88	85	84	**94**	**95**	91
Place Object Stand	74	65	86	88	91	93	**98**	**97**	**98**	**97**	**98**	**97**
Place Phone Stand	49	53	88	**87**	80	78	87	86	83	86	**89**	**87**
Place Shoe	57	61	96	95	95	92	**99**	97	90	95	96	**98**
Press Stapler	80	70	92	**98**	**97**	94	93	**98**	95	97	93	96
Put Bottles Dustbin	12	9	74	77	36	33	81	79	88	**91**	**94**	88

Continued on next page

Table 7 continued

Task	$\pi_{0.5}$ Clean	$\pi_{0.5}$ Rand.	X-VLA Clean	X-VLA Rand.	Motus w/o Pretrain Clean	Motus w/o Pretrain Rand.	Motus w/ Pretrain Clean	Motus w/ Pretrain Rand.	3DFlowWAM w/o midtrain Clean	3DFlowWAM w/o midtrain Rand.	3DFlowWAM Clean	3DFlowWAM Rand.
Put Object Cabinet	24	15	46	48	84	64	**88**	71	78	82	80	**90**
Rotate QR Code	47	56	34	33	80	60	**89**	73	**89**	**91**	**89**	80
Scan Object	42	38	14	36	42	50	67	66	92	91	**94**	**92**
Shake Bottle Horizontally	96	**100**	**100**	**100**	**100**	97	**100**	98	99	99	98	98
Shake Bottle	91	**100**	99	**100**	**100**	96	**100**	97	98	99	99	99
Stack Blocks Three	15	16	6	10	71	76	91	**95**	94	91	**95**	91
Stack Blocks Two	48	56	92	87	96	94	**100**	98	99	**100**	**100**	**100**
Stack Bowls Three	33	35	76	86	90	74	79	87	**95**	**92**	86	71
Stack Bowls Two	78	66	96	93	**98**	**98**	**98**	**98**	94	96	96	97
Stamp Seal	36	23	76	82	80	88	93	92	96	93	**100**	**96**
Turn Switch	5	6	40	61	69	60	**84**	**78**	56	64	58	55
50-task mean	42.98	43.84	72.80	72.84	77.56	77.00	88.66	87.02	89.76	88.54	**91.48**	**89.90**

F FULL LIBERO AND LIBERO-PLUS RESULTS

Table 8: **LIBERO-Plus (2026) robustness (%).** Success under seven observation and initialization perturbations. Bold/underline: best/second-best displayed score; –: unavailable; ~: reweighted aggregate. Sources: cited evaluations.

Type	Method	Camera	Robot	Lang.	Light	Bkgd.	Noise	Layout	Total
I	OpenVLA-OFT (2025)	56.4	31.9	79.5	88.7	93.3	75.8	74.2	69.6
	π_0-FAST (2025)	65.1	21.6	61.0	73.2	73.2	74.4	68.8	61.6
	$\pi_{0.5}$ (2025)	75.4	77.5	85.6	96.9	94.6	89.7	85.7	85.7
II	UniVLA (2025b)	1.8	46.2	69.6	69.0	81.0	21.2	31.9	42.9
	VLA-JEPA (2026)	64.2	67.7	88.1	91.8	93.4	65.8	83.9	77.9
III	HoloBrain-0-GD (2026b)	65.5	58.2	78.7	88.1	90.3	66.9	79.5	74.0
	LaMP (2026b)	64.5	69.6	88.2	95.3	**97.4**	76.9	73.8	79.3
IV	Fast-WAM (2026b)	16.4	44.5	68.9	78.2	53.7	37.7	60.7	50.0
	DC-WAM (2026)	23.9	51.7	83.4	91.7	61.3	54.2	69.8	60.9
	DreamZero (2026)	53.3	59.8	81.2	92.1	86.2	82.8	70.8	73.9
	DreamZero* (2026)	78.2	58.0	80.0	91.5	85.0	85.0	70.0	77.4
	Cosmos Policy (2026)	75.8	63.3	81.7	96.5	88.9	92.7	82.2	82.2
	ABot-M0.5 (2026a)	70.5	**87.4**	88.6	94.0	89.7	75.5	85.2	83.4
V	Being-H0.7 (2026)	82.0	59.0	82.8	97.8	90.0	93.5	**88.5**	82.1
VI	ImageWAM (4B) (2026c)	80.8	50.3	91.4	98.1	85.5	**93.8**	80.5	83.1
	ImageWAM (9B)	79.8	58.7	**95.2**	96.1	91.2	93.3	83.1	85.2
VII	DreamWAM (2026a)	53.8	63.6	94.8	96.7	71.6	67.2	80.7	~74.6
VIII	**3DFlowWAM w/o midtrain**	46.8	76.4	89.1	98.1	86.5	73.4	85.4	78.1
	3DFlowWAM w/ midtrain	**93.7**	74.2	86.8	**99.2**	82.4	92.3	86.4	**87.7**

We report standard LIBERO-Plus and LIBERO results separately, as they evaluate complementary aspects of policy performance. LIBERO measures in-distribution task success, while LIBERO-Plus evaluates robustness under seven observation and initialization perturbations. Tables 9 and 8 provide the corresponding comparisons.

Table 9: **LIBERO (2023) task success (%).** Success rates on the four standard LIBERO suites and their average. Bold/underline: best/second-best displayed score; –: unavailable. Sources: cited evaluations.

Type	Method	Spatial	Object	Goal	Long-Horizon	Avg.
I	OpenVLA-OFT (2025)	97.6	98.4	97.9	94.5	97.1
	π_0-FAST (2025)	96.4	96.8	88.6	60.2	85.5
	$\pi_{0.5}$ (2025)	98.8	98.2	98.0	92.4	96.9
II	UniVLA (2025b)	96.5	96.8	95.6	92.0	95.2
	VLA-JEPA (2026)	–	–	–	–	97.2
III	HoloBrain-0-GD (2026b)	97.8	98.2	95.2	95.6	96.7
	LaMP (2026b)	99.4	<u>99.8</u>	97.4	96.7	98.3
IV	Fast-WAM (2026b)	98.2	**100.0**	97.0	95.2	97.6
	DC-WAM (2026)	–	–	–	–	98.1
	DreamZero (2026)	96.8	97.4	90.8	93.6	94.65
	DreamZero* (2026)	97.3	97.6	94.3	95.2	96.1
	Cosmos Policy (2026)	98.1	**100.0**	98.2	<u>97.6</u>	98.5
	ABot-M0.5 (2026a)	**100.0**	<u>99.8</u>	**99.4**	**98.4**	**99.4**
V	Being-H0.7 (2026)	–	–	–	–	<u>99.2</u>
VI	ImageWAM (4B) (2026c)	97.2	99.2	98.8	**98.4**	98.4
	ImageWAM (9B)	–	–	–	–	–
VII	DreamWAM (2026a)	<u>99.6</u>	<u>99.8</u>	98.6	<u>97.6</u>	98.9
VIII	3DFlowWAM w/o midtrain	96.3	98.2	<u>98.9</u>	95.4	97.2
	3DFlowWAM w/ midtrain	97.5	99.4	98.6	97.2	98.2

Paradigms. **I**: 2D VLA: RGB–language policies. **II**: 2D VLA: latent visual/action representations. **III**: 3D-aware VLA: geometric encoding/motion priors. **IV**: 2D WAM: video prediction/latent-action modeling. **V**: 2D WAM: latent world prediction. **VI**: 2D WAM: image-editing prediction. **VII**: 3D-aware WAM. **VIII**: 3D WAM: RGB + 3D Flow.

G Training and Inference Algorithm

Algorithms 1 and 2 summarize the training and closed-loop inference procedures of 3DFlowWAM. We follow the notation in Section 3: z^V, z^M, and z^A denote video, 3D Flow, and action latents, respectively, and $S_k = [z_k^V; z_k^M; z_k^A]$ denotes prediction block k. During training, the three modalities are jointly denoised using ground-truth RGB history as clean context. At inference, this context is replaced by RGB observations collected during execution.

Algorithm 1 3DFlowWAM Training with Joint Video–3D Flow–Action Prediction

Require: RGB trajectory $I_{0:H}$, 3D Flow $M_{0:H-1}$, actions $a_{1:H}$, robot states $s_{0:H}$, language instruction ℓ
Require: Model parameters θ and modality weights $\lambda_V, \lambda_M, \lambda_A$
$z^V \leftarrow E_V(I_{0:H})$
$z^A \leftarrow E_A(a_{1:H})$
for $t = 0, \dots, H-1$ **do**
    $z_t^M \leftarrow [E_P(\bar{P}_t); E_F(\bar{F}_t^{\Delta})]$
end for
Sample $\sigma \sim \mathcal{U}(0,1)$
for $m \in \{V, M, A\}$ **do**
    Sample $\epsilon^m \sim \mathcal{N}(0, I)$
    $z_\sigma^m \leftarrow (1-\sigma)z^m + \sigma\epsilon^m$
    $v^{m,*} \leftarrow \epsilon^m - z^m$
end for
Partition z_σ^V, z_σ^M, and z_σ^A into N temporally aligned prediction blocks
for $k = 1, \dots, N$ **do**
    $S_k \leftarrow [z_{\sigma,k}^V; z_{\sigma,k}^M; z_{\sigma,k}^A]$
    Jointly predict $v_{\theta,k}^V, v_{\theta,k}^M, v_{\theta,k}^A$ from S_k, $I_{\le t_k}$, s_{t_k}, and ℓ ▷ causal teacher forcing
end for
$\mathcal{L} \leftarrow \sum_{m \in \{V,M,A\}} \lambda_m \mathbb{E}\big[w(\sigma)\|v_\theta^m - v^{m,*}\|_2^2\big]$
Update θ using $\nabla_\theta \mathcal{L}$

Algorithm 2 3DFlowWAM Closed-Loop Inference

Require: Initial RGB observation I_0, robot state s_0, language instruction ℓ
Require: Trained model parameters θ
Initialize observed RGB history with I_0
Initialize the transformer KV cache
for $k = 1, \dots, N$ **do**
    Initialize $z_k^V, z_k^M, z_k^A \sim \mathcal{N}(0, I)$
    for solver step $j = 1, \dots, J$ **do**
        Predict $v_{\theta,k}^V, v_{\theta,k}^M, v_{\theta,k}^A$ conditioned on $I_{\le t_k}$, s_{t_k}, and ℓ
        Update z_k^V, z_k^M, z_k^A with the flow-matching solver
    end for
    Decode z_k^V to predicted RGB observations
    Decode z_k^M to predicted 3D Flow
    Decode z_k^A to action chunk $\hat{a}_k$
    Execute $\hat{a}_k$
    Observe new RGB frames and robot state
    Append the new RGB observations to $I_{\le t_k}$ and update the KV cache
end for