

Trace-WAM: Learning to Act via 3D Motion Prediction for World Action Models

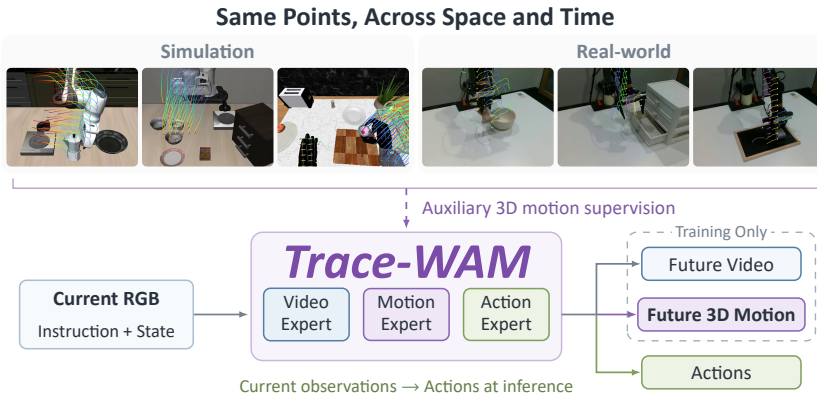
Xu Hu^{1,2} Changjie Wu² Linzhe Shi² Hang Zhang² Chen Yang^{2,3}
Lingjun Zhang^{2,3} Guowen Zhang¹ Mu Xu² Junran Peng⁵ Qing Li^{1,†} Zhaoxiang Zhang^{4,†}

¹The Hong Kong Polytechnic University ²Amap, Alibaba Group ³Tsinghua University

⁴NLPR, Institute of Automation, Chinese Academy of Sciences (CASIA) ⁵University of Science and Technology Beijing

Project Page: <https://trace-wam.github.io/>

(a) Learning from 3D Motion



(b) Performance

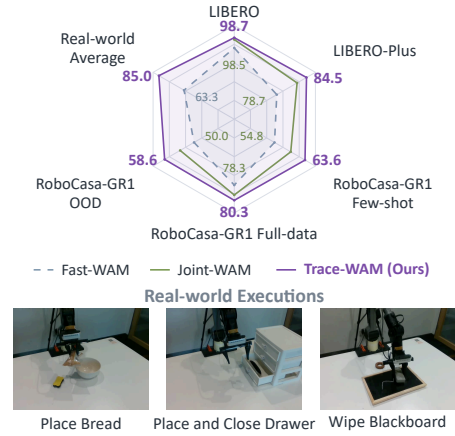


Figure 1. **Overview of Trace-WAM.** (a) Motion annotations from simulation and real-world demonstrations provide auxiliary 3D motion supervision for Trace-WAM. The model jointly learns video, motion, and action prediction; future video and motion prediction are used only during training. (b) Benchmark performance and real-world policy executions.

Abstract

World Action Models (WAMs) learn representations for robot control by coupling future visual prediction with action generation. Successful manipulation depends primarily on how the robot and task-relevant objects evolve, whereas future-video prediction must also reconstruct the appearance and background of the entire scene. We therefore propose Trace-WAM, which augments video-action learning with future 3D motion prediction as an auxiliary objective. This objective supervises temporal point correspondence together with image-plane displacement and depth change, directly describing the interaction dynamics and spatial changes involved in manipulation. To this end, we adopt a three-branch Mixture-of-Transformer (Tri-MoT) architecture to jointly learn video prediction, motion prediction, and action generation. The motion expert pre-

dicts future trajectories from current visual features, allowing motion supervision to shape the representations used for action generation. Because the action expert does not consume predicted trajectories, inference retains the current-observation policy path: a single visual forward pass followed by action denoising. Experiments on RoboCasa-GR1, LIBERO, and LIBERO-Plus demonstrate strong in-distribution performance and out-of-distribution generalization, with particularly clear gains over Fast-WAM under limited demonstrations and distribution shifts.

1. Introduction

Vision-Language-Action (VLA) models [14, 16, 33] adapt pretrained vision-language representations to robot control, learning to generate actions from visual observations and language instructions. Despite their strong policy-learning capabilities, most VLAs focus on action prediction without

[†]Corresponding authors.

explicitly modeling how the scene evolves during execution. World Action Models (WAMs) incorporate this future evolution into policy learning by combining future visual prediction with action generation [12, 48]. Built on pre-trained video models, WAMs transfer rich spatiotemporal priors to robot manipulation by learning from future observations together with action sequences. Yet successful manipulation depends primarily on how the robot and task-relevant objects change over time, whereas future RGB prediction must also model texture, illumination, and background appearance across the entire scene. This difference motivates a complementary target that emphasizes interaction dynamics while retaining the rich visual context learned from future RGB. Recent approaches enrich policy representations by augmenting visual prediction with auxiliary supervision for geometry, semantics, and interaction structure. These objectives include depth and geometric prediction, semantic feature learning, and supervision of affordances and goal regions [12, 24, 30, 47, 49, 53], encouraging representations to capture spatial layout and task-relevant interaction constraints. A per-frame or region-level target, however, does not by itself identify how the same physical locations evolve throughout an interaction, a distinction also emphasized by 4D-WAM [42]. This temporal correspondence connects the current scene to the sequence of changes required to complete a manipulation task.

We therefore focus on **3D motion** as an auxiliary prediction target for WAM representation learning, motivated by two complementary properties. First, each trajectory connects a point in the current observation to its positions at successive future steps, revealing the coordinated motion of the robot and task-relevant objects (Fig. 1(a)). Second, depth change complements image-plane displacement by describing motion toward or away from the camera. In our annotation census, among point tracks moving by at least 2% of the image diagonal over 32 steps, 79.3% in LIBERO and 32.9% in Piper exhibit depth changes of at least 5% of their initial depth (Fig. 2(a)). Depth variation therefore accompanies a substantial fraction of visible motion in both simulation and real-world annotations. Its value for policy learning is supported by our depth ablation: retaining the depth component improves LIBERO-Plus success from 82.4% to 84.5% (Fig. 2(b)). These properties describe the temporal and spatial changes involved in manipulation.

Based on this formulation, we propose **Trace-WAM**, which jointly learns future video prediction, future 3D motion prediction, and action generation. We represent motion using trajectories anchored to a fixed grid in the current image, combining image-plane displacement with depth change relative to each point’s initial position. Whereas 4D-WAM [42] transfers trajectory knowledge by aligning temporal feature changes and source-to-destination relations with those of a pretrained 4D tracking model, Trace-WAM

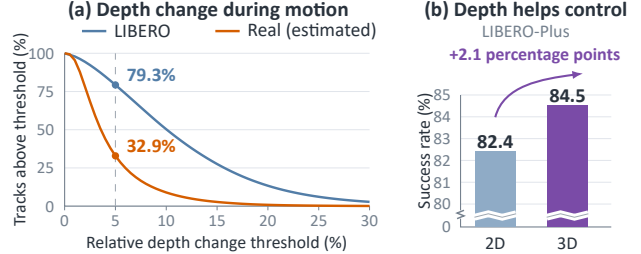


Figure 2. **Depth complements image-plane motion.** (a) Depth-change survival curves over 32-step point tracks whose maximum 2D displacement is at least 2% of the image diagonal. Depth change is $\max_k |d_{t+k} - d_t|/d_t$. LIBERO uses simulator depth; Real uses estimated relative depth. (b) Retaining depth in motion supervision improves LIBERO-Plus success (Tab. 4).

directly predicts motion trajectories under an auxiliary objective. This objective adds explicit supervision for subsequent robot and object motion to video-action representation learning. Concretely, Trace-WAM uses video, motion, and action experts with a structured attention mask (Fig. 3). During training, the video, motion, and action objectives jointly optimize the model, using future 3D motion as an additional learning signal for the representations used in action generation. At inference, we retain the Fast-WAM [48] policy path: the video backbone processes current RGB observations once, and the action expert denoises an action chunk using cached visual features, language, and proprioception. As summarized in Fig. 1(a), Trace-WAM requires no future-video or motion prediction at inference.

We evaluate Trace-WAM on RoboCasa-GR1, LIBERO, and LIBERO-Plus, covering both in-distribution control and out-of-distribution generalization. Trace-WAM achieves strong performance across these settings, with particularly clear gains over Fast-WAM when demonstrations are limited and under distribution shifts (Fig. 1(b)). Experiments on a real robot further show higher success rates across three physical manipulation tasks. These results support 3D motion prediction as an effective auxiliary objective for WAM policy learning. Our contributions are:

- We introduce Trace-WAM, which jointly learns future video prediction, future 3D motion prediction, and action generation within a three-branch architecture.
- Future 3D motion provides auxiliary supervision for video-action representation learning without requiring future video or motion prediction at inference.
- Experiments on three manipulation benchmarks demonstrate strong in-distribution performance and out-of-distribution generalization, while real-world evaluations show consistent gains across three physical tasks.

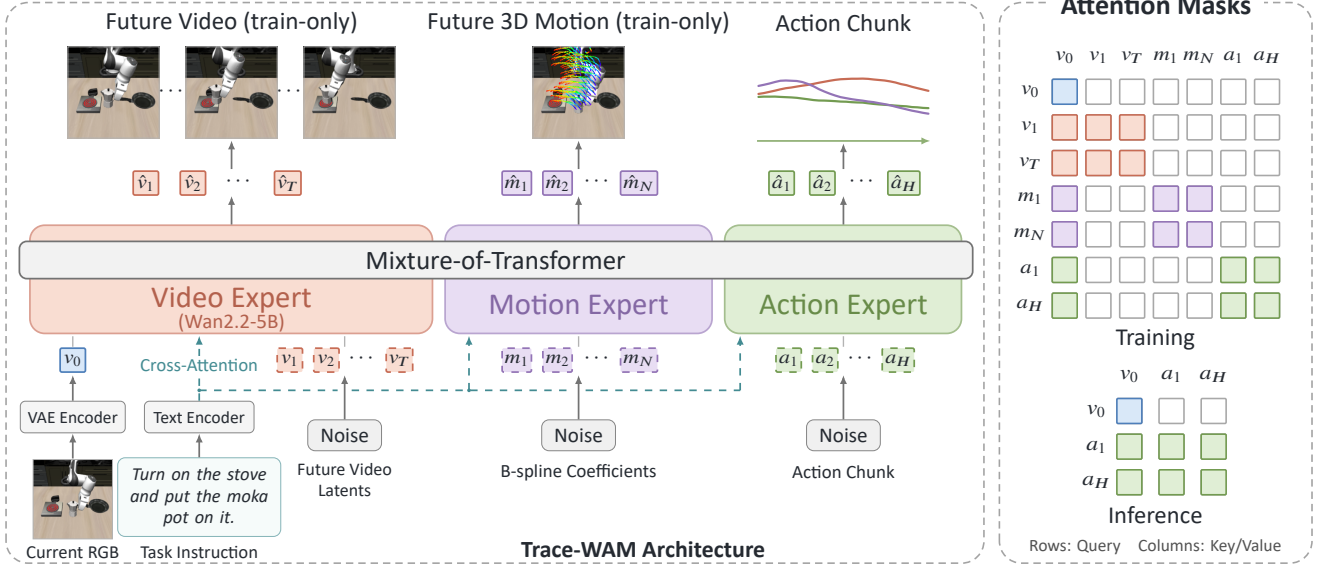


Figure 3. **Overview of Trace-WAM.** Left: three Transformer experts jointly learn future video prediction, 3D motion prediction, and action generation. Right: colored cells allow attention among the illustrated video, motion, and action streams. At inference, future-video and motion prediction are removed, and action denoising uses cached visual keys and values.

2. Related Work

2.1. World Action Models

Early approaches formulate manipulation as visual planning, using generative models to propose future observations and learned controllers to translate the resulting visual plans into executable actions [4, 10]. More recent World Action Models integrate video priors more directly into policy learning, either conditioning actions on predictive video features [13, 25] or jointly modeling future visual observations and action sequences [1, 18, 22, 31, 34, 41, 46].

Beyond RGB prediction, recent methods extend predictive supervision to semantic features, depth and 3D geometry, and motion, encouraging policy representations to capture object identity, spatial layout, and scene dynamics [21, 30, 38, 40, 41, 47]. Complementary supervision of affordances, object-centric interaction configurations, and goal regions emphasizes which scene elements are involved in the task and where the robot should interact [49, 53]. Trace-WAM follows this direction by using auxiliary future 3D motion prediction to supervise temporal point correspondence and depth change jointly with future RGB prediction and action generation.

2.2. Motion for Policy Learning

Motion representations connect action generation to the spatial and temporal structure of manipulation by describing how the robot and objects move throughout an interaction. Incorporating depth further captures displacement along the viewing direction, complementing image-plane

motion with spatial information needed for reaching, grasping, and object transport. Existing approaches couple action learning with optical flow or 2D point-motion prediction, using motion features or predicted motion to guide control [8, 37, 43], while others incorporate depth and 3D motion to model spatial scene dynamics for action generation [20, 39, 52]. A complementary line of work uses motion supervision to improve policy representations, through distillation from pretrained tracking models [36, 42] or auxiliary 3D motion prediction during human-robot co-training [21]. Trace-WAM directly predicts future 3D point trajectories alongside RGB prediction and action generation, using numerical motion targets rather than tracker-feature alignment. The motion objective shapes video-action representations during training; deployed actions use current visual context, language, and proprioception without future-video or motion prediction.

3. Method

Trace-WAM jointly learns future video prediction, future 3D motion prediction, and action generation. Future 3D motion provides auxiliary supervision for video-action representation learning, while the deployed policy generates actions from current observations. Figure 3 summarizes the Tri-MoT architecture and its attention masks.

3.1. Problem Formulation

We consider language-conditioned robot manipulation. At each decision step, the policy receives current RGB obser-

uations I_0 , a language instruction ℓ , and the robot’s proprioceptive state s_0 . It generates an action chunk

$$A \sim \pi_\theta(\cdot \mid I_0, \ell, s_0), \quad A \in \mathbb{R}^{H_A \times D_A}, \quad (1)$$

where H_A is the action horizon and D_A is the action dimension. The robot executes the first few actions before receiving a new observation and replanning.

We follow Fast-WAM [48] in training the policy with future video prediction while generating actions from current observations. Trace-WAM adds future 3D motion prediction as an auxiliary objective. Future RGB and motion labels are used only during training.

3.2. 3D Motion Representation

Motion Targets. Following the grid-based motion representation of LaMP [39], we place Q anchors on a uniform grid covering the current main-camera image and associate each anchor with its positions over H_M future steps. Our reference configuration uses a 20×20 grid ($Q = 400$) and $H_M = 32$, giving 33 positions per anchor when the current frame is included. Anchors follow a fixed row-major order. Although the policy can receive multiple camera views, the motion grid is defined only in the main-camera image.

For anchor q at step t , let $p_{q,t} = (u_{q,t}, v_{q,t}, d_{q,t})$ contain image coordinates normalized to $[-1, 1]$ and depth, measured in meters for simulator annotations and in an aligned relative scene scale for TraceForge annotations. Motion is displacement relative to its current-image position:

$$\Delta p_{q,t} = p_{q,t} - p_{q,0}, \quad t = 0, \dots, H_M, \quad (2)$$

where $\Delta p_{q,0} = 0$. The resulting target $\Delta P \in \mathbb{R}^{Q \times (H_M+1) \times 3}$ combines image-plane displacement with depth change. Every displacement is measured from the current position, rather than accumulated from adjacent frames. All trajectories are expressed in a unified reference-camera frame before conversion to (u, v, d) , compensating for camera motion. This screen-aligned representation retains the spatial information provided by depth without depending on a particular robot embodiment, while its (u, v) coordinates remain compatible with 2D trajectory formats. For simulated data, motion labels are generated from simulator geometry, depth, and entity poses, whereas real-world videos are processed by the TraceForge pipeline from TraceGen [19] to estimate depth, camera motion, and point tracks; the supplementary material describes the complete annotation and validation procedure.

B-Spline Representation. Following μ_0 [20], we represent each motion sequence with cubic B-spline control points. After scaling each displacement channel using training-set statistics, we fit K control points to the valid positions, including the initial zero displacement:

$$\widetilde{\Delta P}_q \approx BC_q, \quad C_q \in \mathbb{R}^{K \times 3}, \quad (3)$$

where $\widetilde{\Delta P}_q$ is the scaled motion sequence and B is a fixed spline basis matrix. The reference configuration uses $K = 10$ control points per anchor. The fitted coefficients provide a compact target for the motion expert. Multiplying predicted coefficients by B and restoring the channel scales recovers the displacement sequence. This representation preserves the correspondence between each current-image anchor and its future motion.

3.3. Model Architecture

Tri-MoT Architecture. Trace-WAM uses three Transformer experts with separate parameters and input/output projections. A frozen causal video VAE encodes current and future RGB as latent tokens. The video expert processes clean current tokens V_0 and, during training, noisy future tokens V_f . The action expert embeds each step of a noisy action chunk as a token. A trainable projection embeds the current state as proprioceptive tokens; a frozen text encoder supplies instruction embeddings via cross-attention.

Motion Expert. The motion expert operates on noisy B-spline coefficients. We arrange the $3K$ coefficients of each anchor as channels on the spatial grid, then embed them with a convolution whose kernel and stride are both 2×2 . With the reference settings, this maps the 20×20 grid to 100 motion tokens. Each token contains four complete future trajectories, preserving their individual coefficients. This packs the motion horizon into token channels instead of creating a token for every future time step.

We encode each patch center with two-dimensional Fourier features and add their learned projection to its token embedding. Noise-time embeddings modulate the Transformer normalization and gated residual updates. At every layer, motion tokens attend to one another and to current video features, while the instruction and proprioceptive tokens provide task and state conditions. The output head predicts $4 \times K \times 3$ velocity components per token, unpacked into per-anchor coefficient velocities.

Structured Attention. At each layer, the experts project their queries, keys, and values to a common attention dimension. The attention masks in Fig. 3 allow current video tokens to attend only to one another, while future-video, motion, and action tokens attend to their own group and to current video. Attention within future video and within the action chunk is bidirectional across time.

The video, motion, and action objectives are jointly optimized, using future 3D motion as an additional learning signal for the representations used in action generation. Because neither current video nor action tokens attend to motion or future video, the deployed policy does not depend on either prediction branch.

3.4. Training Objective

Flow Matching. We jointly train the three experts with flow matching [26]. Let x_r denote the clean target for branch $r \in \{V, M, A\}$: future video latents, motion coefficients, or an action chunk. For each branch, Gaussian noise ϵ_r and noise level $\sigma_r \in [0, 1]$ define

$$x_{r,\sigma_r} = (1 - \sigma_r)x_r + \sigma_r\epsilon_r, \quad v_r^* = \epsilon_r - x_r. \quad (4)$$

The expert predicts velocity $v_{\theta,r}$ from this noisy target, the noise level, and the conditions allowed by the attention mask. Its basic loss is

$$\mathcal{L}_r^{\text{FM}} = \mathbb{E} \left[w_r(\sigma_r) \|v_{\theta,r} - v_r^*\|^2 \right], \quad (5)$$

where w_r is the noise-level-dependent weight. We use this loss for $\mathcal{L}_V$ and $\mathcal{L}_A$, with motion errors balanced below.

Motion Supervision. Static anchors can greatly outnumber moving anchors. For each sample, we compute mean coefficient-velocity errors ℓ_{mov} and ℓ_{stat} for moving and static anchors separately and combine them:

$$\mathcal{L}_M = \mathbb{E} [w_M(\sigma_M) (\alpha \ell_{\text{mov}} + (1 - \alpha) \ell_{\text{stat}})]. \quad (6)$$

Here, α controls the two groups' contributions.

We additionally penalize residual motion at static anchors in the decoded trajectory space. From the predicted velocity, we estimate clean coefficients as $\hat{C}_0 = C_{\sigma_M} - \sigma_M v_{\theta,M}$ and decode them with the spline basis and channel scales to obtain $\hat{\Delta p}_{q,t}$. The static-motion loss is

$$\mathcal{L}_{\text{zero}} = \mathbb{E}_{\text{stat}} \left[\text{mean}_{q \in \mathcal{S}} \text{mean}_{t \in \mathcal{T}_q} \rho_\beta \left(\hat{\Delta p}_{q,t} \right) \right], \quad (7)$$

where $\mathcal{S}$ and $\mathcal{T}_q$ index static anchors and future times, respectively, and ρ_β is a channel-averaged Smooth-L1 penalty. It directly encourages stationary points to remain stationary after decoding.

Joint Training. The full objective combines video prediction, balanced motion prediction, static-motion regularization, and action prediction:

$$\mathcal{L} = \lambda_V \mathcal{L}_V + \lambda_M \mathcal{L}_M + \lambda_{\text{zero}} \mathcal{L}_{\text{zero}} + \lambda_A \mathcal{L}_A, \quad (8)$$

where the λ terms set the loss weights. Training begins with a warmup using the video and motion objectives, including the static-motion loss. We then add the action objective and jointly optimize all three experts. The VAE and text encoder remain frozen; the three experts and proprioceptive-token projection are trainable.

Table 1. **In-distribution performance on RoboCasa-GR1.** Success rates (%) in the few-shot and full-data settings. P.T. indicates additional robotic pretraining. DIAL is trained without human data; Best and second-best results are shown in bold and underlined, respectively.

Method	P.T.	Few-shot	Full
GR00T N1.6 [2]	✓	—	47.6
StarVLA-OFT [9]	×	—	48.8
StarVLA- α [45]	×	—	53.8
ABot-M0 [44]	✓	—	58.3
RLDX-1 [15]	✓	—	58.7
JoyAI-RA [50]	✓	—	63.2
DIAL [7]	×	58.3	70.2
Being-H0.7 [28]	✓	—	49.2
DiT4DiT [31]	✓	—	50.8
LDA-1B [29]	✓	—	55.4
LAWA w/ ego [23]	✓	65.6	80.8
LAWA w/o ego [23]	×	59.7	76.3
Fast-WAM [48]	×	44.8	74.6
Joint-WAM [48]	×	54.8	78.3
Trace-WAM (Ours)	×	<u>63.6</u>	<u>80.3</u>

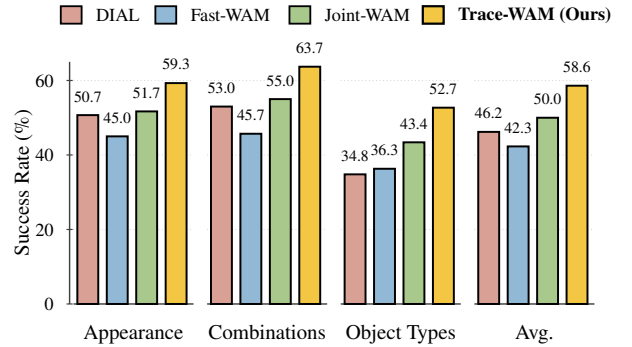


Figure 4. **Few-shot OOD generalization on RoboCasa-GR1.**

4. Experiments

4.1. Experimental Setup

Benchmarks. We evaluate Trace-WAM on RoboCasa-GR1 [32], LIBERO [27], and LIBERO-Plus [11] using task success rate. RoboCasa-GR1 contains 24 tabletop rearrangement and articulated-object manipulation tasks. We evaluate both full-data (24,000 trajectories, 1,000 per task) and few-shot settings (10% subset), and test the few-shot policy on Unseen Appearance, Unseen Combinations, and Unseen Object Types following DIAL [7]. LIBERO comprises four task suites: Spatial, Object, Goal, and Long. All models train solely on original LIBERO demonstrations and transfer zero-shot to LIBERO-Plus under seven perturbations: camera, robot initial state, language, lighting, back-

Table 2. **Quantitative comparison on LIBERO [27].** Success rates (%) across four task suites.

Method	P.T.	Task Suite				Avg.
		Spatial	Object	Goal	Long	
OpenVLA [16]	✓	84.7	88.4	79.2	53.7	76.5
OpenVLA-OFT [17]	✓	97.6	98.4	97.9	94.5	97.1
UniVLA [5]	✓	96.5	96.8	95.6	92.0	95.2
π_0 [3]	✓	96.8	98.8	95.8	85.2	94.2
π_0 -FAST [33]	✓	96.4	96.8	88.6	60.2	85.5
$\pi_{0.5}$ [14]	✓	98.8	98.2	98	92.4	96.9
LingBot-VA [22]	✓	98.5	99.6	97.2	98.5	98.5
Motus [1]	✓	96.8	<u>99.8</u>	96.6	97.6	97.7
LaWAM [6]	✓	<u>99</u>	96	97.2	93.4	96.4
IDM-WAM [48]	×	98.6	99.4	98.4	96.4	98.2
Fast-WAM [48]	×	97.2	100	97	95.2	97.6
Joint-WAM [48]	×	99.6	99.4	98.2	96.8	98.5
ImageWAM [51]	×	97.2	99.2	98.8	<u>98.4</u>	98.4
GIFT-WAM-Fast [53]	×	97.8	99.6	<u>98.6</u>	<u>94.8</u>	97.7
GeoSem-WAM [30]	×	<u>99.0</u>	100.0	98.2	97.0	<u>98.6</u>
Trace-WAM (Ours)	×	98.2	<u>99.8</u>	99.2	97.6	98.7

ground, sensor noise, and layout. No augmented LIBERO-Plus data are used for fine-tuning.

Implementation Details. We build on the Fast-WAM [48] codebase and training settings, using pretrained Wan2.2-5B [35] as the video backbone. The action and motion experts each follow the video Transformer’s architecture with a hidden dimension of 1,024. Both action and motion horizons are set to 32, and videos are temporally downsampled by $4\times$ to nine frames per chunk. For LIBERO and RoboCasa-GR1 few-shot, we train for 26,000 steps on 16 NVIDIA H20 GPUs with a global batch size of 128. For RoboCasa-GR1 full-data, we train for 50,000 steps on 32 NVIDIA H20 GPUs with a global batch size of 1,024. Each schedule starts with 2,000 steps of video–motion experts warmup, followed by joint training with the action expert for the remaining steps.

4.2. Main Results

Performance on RoboCasa. As shown in Table 1, Trace-WAM achieves success rates of 63.6% and 80.3% in the few-shot and full-data settings, outperforming Fast-WAM and Joint-WAM in both. The gains are especially pronounced with limited demonstrations, reaching 18.8 and 8.8 percentage points over the two baselines, respectively. Figure 4 further shows consistent generalization across unseen appearances, object combinations, and object types. Trace-WAM leads all three comparison methods in every setting and reaches 58.6% average success, exceeding DIAL, Fast-WAM, and Joint-WAM by 12.4, 16.3, and 8.6 percentage

points, respectively. Together, these results show that future 3D motion supervision supports both in-distribution performance and OOD generalization.

Performance on LIBERO. Trace-WAM achieves an average success rate of 98.7% across the four LIBERO suites (Table 2), compared with 97.6% for Fast-WAM and 98.5% for Joint-WAM. This further demonstrates strong in-distribution performance across diverse manipulation tasks.

Zero-shot Robustness on LIBERO-Plus. Without further adaptation, Trace-WAM reaches 84.5% success on LIBERO-Plus (Table 3), exceeding Fast-WAM and Joint-WAM by 18.3 and 5.8 percentage points, respectively. It improves over Fast-WAM across all seven perturbation types. In particular, success rates under Robot and Background perturbations reach 87.4% and 76.1%, improving over Fast-WAM by 18.9 and 9.2 points, respectively. These gains support the value of motion supervision for learning interaction-relevant representations: temporal correspondence and depth change describe how the robot and objects move during manipulation, encouraging features that remain useful under altered robot initial states and task-irrelevant background changes.

4.3. Ablation Study and Analysis

Motion Supervision Ablation. Table 4 shows the contributions of the two motion-supervision components. The *w/o Motion FM-split* variant removes the separate balancing of moving and static anchors in the motion flow-matching loss, achieving 80.4% average success. The *w/o Static-Motion Penalty* variant achieves 83.6%. Trace-WAM reaches 84.5%, improving over these variants by 4.1 and 0.9 percentage points, respectively. These gains highlight the importance of balancing supervision from moving and static anchors, with static-motion regularization providing a further improvement in task success.

Motion Conditioning Ablation. The *Motion Inject* variant injects future 3D motion latents from each motion-expert layer into the corresponding action-expert layer, analogous to the layerwise video–action connection in Joint-WAM, and achieves 81.6% average success. The *Video–Motion Bid-Attention* variant instead allows the video and motion streams to attend to each other at every layer and achieves 80.3%. Trace-WAM keeps future 3D motion as a jointly optimized auxiliary objective and reaches 84.5%, improving over these variants by 2.9 and 4.2 percentage points, respectively. These results favor auxiliary 3D motion supervision within video–action representation learning, without bidirectional video–motion coupling or action conditioning on future motion latents.

Table 3. **Zero-shot transfer from LIBERO to LIBERO-Plus [11].** Success rates (%) across seven perturbation types.

Method	P.T.	Perturbation Type							Avg.
		Camera	Robot	Language	Light	Background	Noise	Layout	
OpenVLA [16]	✓	0.8	3.5	23.0	8.1	34.8	15.2	28.5	15.6
OpenVLA-OFT [17]	✓	56.4	31.9	79.5	88.7	93.3	75.8	74.2	69.6
UniVLA [5]	✓	1.8	46.2	69.6	69.0	81.0	21.2	31.9	42.9
π_0 [3]	✓	13.8	6.0	58.8	85.0	81.4	79.0	68.8	53.6
π_0 -FAST [33]	✓	65.1	21.6	61.0	73.2	73.2	74.4	68.8	61.6
Fast-WAM [48]	×	40	68.5	65.6	89.8	66.9	63.3	76.8	66.2
Joint-WAM [48]	×	59.4	<u>78.3</u>	92.7	95.7	65.2	75.6	<u>84.5</u>	78.7
ImageWAM [51]	×	80.8	50.3	91.4	98.1	<u>85.5</u>	93.8	80.5	<u>83.1</u>
GIFT-WAM-Fast [53]	×	37.4	75.3	79.5	94.5	70.3	78.7	78.8	72.6
4D-WAM [42]	×	45.15	64.26	90.63	94.29	57.71	69.08	79.21	71.01
LAWA [23]	×	<u>69.2</u>	66.7	62.8	96.2	64.5	85.5	78.6	74.4
Trace-WAM (Ours)	×	66	87.4	91.9	<u>97.2</u>	76.1	<u>88.6</u>	85.7	84.5

Table 4. **Ablation Studies on LIBERO-Plus.** Success rates (%) across seven perturbation types. (1) Motion supervision ablates moving/static balancing in $\mathcal{L}_M$ and the static-motion penalty $\mathcal{L}_{\text{zero}}$. (2) Motion conditioning compares auxiliary supervision with Motion Inject and bidirectional video-motion attention. (3) Motion representation compares 2D/3D motion and raw-motion/B-spline targets. Best and second-best results are shown in bold and underlined, respectively.

Method	Camera	Robot	Language	Light	Background	Noise	Layout	Avg.
1) Motion Supervision Ablation								
w/o Motion FM-split	55.7	84.5	89.2	96.1	70.5	85.1	83.7	80.4
w/o Static-Motion Penalty	60.9	<u>86.9</u>	<u>93.4</u>	97.3	<u>75.3</u>	<u>88.1</u>	85	<u>83.6</u>
2) Motion Conditioning Ablation								
Motion Inject	61.7	82.8	<u>93.4</u>	96.3	73.1	81	84.9	81.6
Video-Motion Bid-Attention	54.4	83.5	94	96.2	68.3	82.3	84.8	80.3
3) Motion Representation Ablation								
w/o Depth (2D Motion)	<u>65.1</u>	86.3	91.9	95.4	70.3	82.6	85.8	82.4
w/o B-spline (Raw Motion)	64.2	85.5	91.3	<u>97.2</u>	75.1	86.8	85.4	83.4
Trace-WAM (Ours)	66	87.4	91.9	<u>97.2</u>	76.1	88.6	<u>85.7</u>	84.5

Motion Representation Ablation. Trace-WAM outperforms *w/o Depth (2D Motion)* and *w/o B-spline (Raw Motion)*, which achieve 82.4% and 83.4% average success, respectively. Adding depth yields a 2.1-point gain, with improvements of 5.8 and 6.0 points under Background and Noise perturbations. This supports the value of 3D motion supervision beyond image-plane displacement for OOD generalization. B-spline targets improve success by 1.1 points over direct per-step displacements, supporting compact motion prediction more effective for representation learning.

Motion Prediction Accuracy. We compare predicted motion with the motion realized in the simulator after executing the policy actions on LIBERO-Plus. Table 5 reports image-plane average displacement error (ADE) and depth mean absolute error (MAE) over all valid query-time

Table 5. **Motion Prediction Accuracy on LIBERO-Plus.** We aggregate 140 rollout episodes at each horizon; lower is better.

Motion Representation	8 Steps	16 Steps	32 Steps
<i>Image-Plane ADE (pixels, ↓)</i>			
Raw Motion (Direct32)	0.92	1.25	1.87
B-spline Motion	0.67	0.95	1.43
<i>Depth MAE (mm, ↓)</i>			
Raw Motion (Direct32)	1.634	2.297	3.427
B-spline Motion	1.277	1.808	2.768

pairs at 8-, 16-, and 32-step horizons. Both representations closely match the executed trajectories. Even at 32 steps, Raw Motion attains an image-plane ADE of 1.87 pixels and a depth MAE of 3.427 mm, while B-spline Motion reaches 1.43 pixels and 2.768 mm. The B-spline representation con-

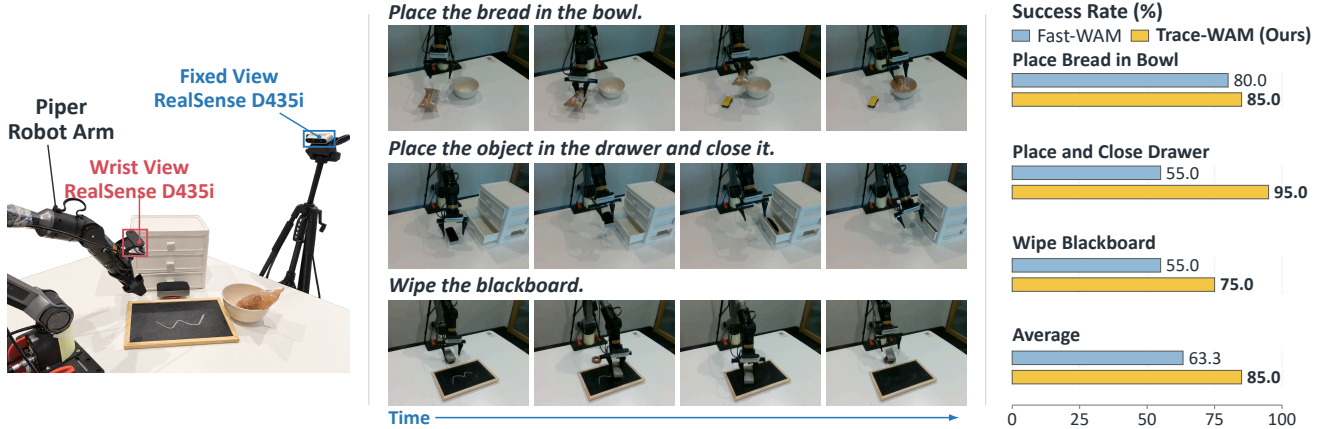


Figure 5. **Real-world Experiments.** Left: a Piper arm with fixed-view and wrist-view RealSense D435i cameras. Center: four observed frames from a successful Trace-WAM execution of each task. Right: task success rates over 20 trials per task.

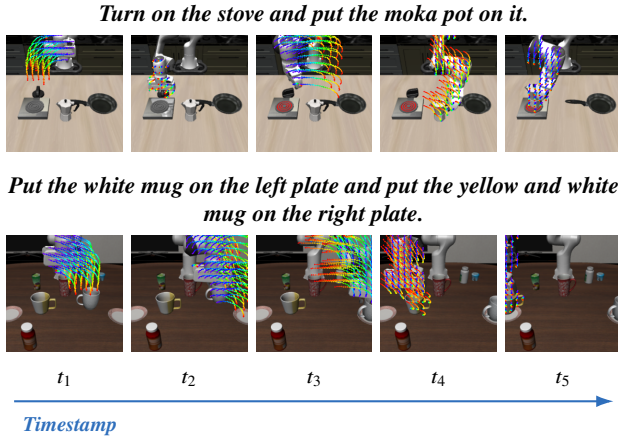


Figure 6. **Qualitative future 3D motion predictions on LIBERO-Plus.** The overlays visualize B-spline motion predictions over a 32-step horizon.

sistently lowers both errors across all three horizons, showing that its compact trajectory parameterization more accurately captures the realized 3D motion. Figure 6 further visualizes B-spline predictions at five ordered replanning steps for two tasks. The predicted trajectories change with the ongoing robot-object interaction as each rollout progresses. Together with its higher task success in Table 4, these results support B-spline motion as the more effective representation for future 3D motion prediction and downstream action generation.

4.4. Real-world Experiments

Setup. The experiments are conducted on the Piper platform with a gripper and two Intel RealSense D435i cameras providing fixed and wrist views (Figure 5). We evaluate three tasks: *Place Bread* requires placing bread in a bowl;

Place and Close Drawer requires placing an object in the drawer and closing it; and *Wipe Blackboard* requires erasing the chalk markings. Each task uses 50 demonstrations, and each method is evaluated over 20 trials per task.

Results. Trace-WAM achieves 85.0% average success across the three physical tasks, compared with 63.3% for Fast-WAM (Figure 5). The largest gain occurs on Place and Close Drawer, where success rises from 55.0% to 95.0%. Wipe Blackboard improves from 55.0% to 75.0%, and Place Bread in Bowl from 80.0% to 85.0%. These results extend the benefits of auxiliary 3D motion supervision to physical manipulation, including sequential placement and closing as well as sustained surface contact.

5. Conclusion

We presented Trace-WAM, which uses auxiliary future 3D motion prediction to focus video-action representation learning on the scene dynamics most relevant to manipulation. The video, motion, and action objectives are jointly optimized during training, while the deployed policy generates actions from current observations, language, and proprioception without running future-video or motion prediction. Across LIBERO, zero-shot LIBERO-Plus, and RoboCasa-GR1, Trace-WAM achieves strong in-distribution and out-of-distribution performance, with clear gains in the few-shot and OOD settings. Real-world experiments on a Piper robot further demonstrate improved success across three manipulation tasks. These results establish future 3D motion prediction as an effective training signal for learning robust action representations in WAMs.

References

- [1] Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chen-

- dong 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*, pages 35101–35113, 2026. 3, 6
- [2] 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. 5
- [3] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0 : A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. 6, 7
- [4] 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*, pages 33431–33452, 2024. 3
- [5] Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025. 6, 7
- [6] Jialei Chen, Kai Wang, Kang Chen, Shuaihang Chen, Feng Gao, Wenhao Tang, Zhiyuan Li, Weilin Liu, Zhuoyu Yao, Boxun Li, et al. Lawam: Latent world action models for efficient dynamics-aware robot policies. *arXiv preprint arXiv:2606.15768*, 2026. 6
- [7] Yi Chen, Yuying Ge, Hui Zhou, Mingyu Ding, Yixiao Ge, and Xihui Liu. Dial: Decoupling intent and action via latent world modeling for end-to-end vla. *arXiv preprint arXiv:2603.29844*, 2026. 5
- [8] 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*, 2026. 3
- [9] StarVLA Community. Starvla: A lego-like codebase for vision-language-action model developing. *arXiv preprint arXiv:2604.05014*, 2026. 5
- [10] 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. 3
- [11] 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. 5, 7
- [12] 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. 2
- [13] 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. *arXiv preprint arXiv:2412.14803*, 2024. 3
- [14] 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. 1, 6
- [15] Dongyoung Kim, Huiwon Jang, Myungkyu Koo, Suhyeok Jang, Taeyoung Kim, Beomjun Kim, Byungjun Yoon, Changsung Jang, Daewon Choi, Dongsu Han, et al. Rldx-1 technical report. *arXiv preprint arXiv:2605.03269*, 2026. 5
- [16] 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, 2024. [URL https://arxiv.org/abs/2406.09246](https://arxiv.org/abs/2406.09246), 1(2):4, 2024. 1, 6, 7
- [17] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025. 6, 7
- [18] Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, et al. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026. 3
- [19] Seungjae Lee, Yoonkyo Jung, Inkook Chun, Yao-Chih Lee, Zikui Cai, Hongjia Huang, Aayush Talreja, Tan Dao, Yongyuan Liang, Jia-Bin Huang, et al. Tracen: World modeling in 3d trace space enables learning from cross-embodiment videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20721–20731, 2026. 4
- [20] Seungjae Lee, Yoonkyo Jung, Jusuk Lee, Jonghun Shin, Amir Hossein Shahidzadeh, Yao-Chih Lee, H Jin Kim, Jia-Bin Huang, and Furong Huang. μ_0 : A scalable 3d interaction-trace world model. *arXiv preprint arXiv:2606.13769*, 2026. 3, 4
- [21] Baoyu Li, Xinchun Yin, Mengying Lin, Yixin Zhang, and Danfei Xu. Egowam: World action models beyond pixels with in-the-wild egocentric human data. In *Robot World Models*, 2026. 3
- [22] 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*, 2026. 3, 6
- [23] Xiang Li, Yupeng Zheng, Songen Gu, Huailiang Ma, Feng Yu, Xian Nie, Shanshuai Yuan, Yujie Zang, Weize Li, Shuai Tian, et al. Latent action as intention enables efficient future imagination for world action models. *arXiv preprint arXiv:2608.24882*, 2026. 5, 7
- [24] Ying Li, Xiaobao Wei, Jiajun Cao, Hao Wang, Xiaowei Chi, Chengyu Bai, Qianpu Sun, Jiajun Li, Xiaojie Zhang, Peidong Jia, et al. Wam4d: Fast 4d world action model via spatial register tokens. *arXiv preprint arXiv:2606.14048*, 2026. 2
- [25] Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Si Liu, Jianlan

- Luo, Liliang Chen, et al. Genie envisioner: A unified world foundation platform for robotic manipulation. In *International Conference on Learning Representations*, pages 88446–88463, 2026. 3
- [26] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 5
- [27] 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. 5, 6
- [28] 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. 5
- [29] Jiangran Lyu, Kai Liu, Xuheng Zhang, Haoran Liao, Yusen Feng, Wenxuan Zhu, Tingrui Shen, Jiayi Chen, Jiazhao Zhang, Yifei Dong, et al. Lda-1b: Scaling latent dynamics action model via universal embodied data ingestion. *arXiv preprint arXiv:2602.12215*, 2026. 5
- [30] Fulong Ma, Daojie Peng, Wenjun Yue, Jiahang Cao, Bintao Wang, Qiang Zhang, and Jun Ma. Geosem-wam: Geometry- and semantic-aware world action models. *arXiv preprint arXiv:2606.03188*, 2026. 2, 3, 6
- [31] Teli Ma, Jia Zheng, Zifan Wang, Chunli Jiang, Andy Cui, Junwei Liang, and Shuo Yang. Dit4dit: Jointly modeling video dynamics and actions for generalizable robot control. *arXiv preprint arXiv:2603.10448*, 2026. 3, 5
- [32] Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. *arXiv preprint arXiv:2406.02523*, 2024. 5
- [33] 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. 1, 6, 7
- [34] MotuBrain Team, Chendong Xiang, Fan Bao, Haitian Liu, Hengkai Tan, Hongzhe Bi, James Li, Jiabao Liu, Jingrui Pang, Kiro Jing, et al. Motubrain: An advanced world action model for robot control. *arXiv preprint arXiv:2604.27792*, 2026. 3
- [35] 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. 6
- [36] 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*, 2026. 3
- [37] Jiayu Wang, Bin Zhu, Yue Yu, and Jingjing Chen. Mowam: Explicit future motion prediction for efficient world action models. *arXiv preprint arXiv:2609.20709*, 2026. 3
- [38] Mingxin Wang, Bin Hu, Bin Qian, Kaitao Jiang, Haoning Wu, Feng Yan, Bowen Jing, Ruiyang Hao, Enyi Wang, Kangning Niu, et al. St-wam: Semantic-temporal world action model for robust manipulation under visual distribution shifts. *arXiv preprint arXiv:2607.28993*, 2026. 3
- [39] 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*, pages 227–244. Springer, 2026. 3, 4
- [40] Ge Yan, Jinghao Liu, Yuzhi Fan, Lei Cai, Minwen Liao, Jesse Zhang, and Dieter Fox. Flex- π : A multi-stream world-action model with compute flexibility, 2026. URL <https://arxiv.org/abs/2608.10860>. 3
- [41] Haodong Yan, Junfeng Li, Junjie He, Zhide Zhong, Mingming Yu, Wenxuan Song, Jiaguan Zhu, Yangyang Zheng, Yuqiao Du, Jiadi You, et al. Robust-wam: Bridging generative pretraining and semantic foresight in world-action models. *arXiv preprint arXiv:2608.05903*, 2026. 3
- [42] Lishan Yang, Wenxuan Song, Xi Wang, Pingyue Sheng, Zheng Fang, Ziyang Zhou, Junjie He, Haodong Yan, Jiayi Chen, Nan Sun, et al. 4d-wam: Infusing spatiotemporal awareness into world action models through trajectory fields. *arXiv preprint arXiv:2608.08023*, 2026. 2, 3, 7
- [43] Yiguang Yang, Jiankun Peng, Xiaoming Wang, Yiran Zhang, and Zhibo Fang. Mt-wam: Reorienting the one-pass predictive representation toward action generation. *arXiv preprint arXiv:2609.21474*, 2026. 3
- [44] 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*, 2026. 5
- [45] 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*, 2026. 5
- [46] 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*, 2026. 3
- [47] 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*, 2026. 2, 3
- [48] 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. 2, 4, 5, 6, 7
- [49] Mengqi Zhang, Sahil Khose, Simar Kareer, Yuchen Song, Unnat Jain, and Judy Hoffman. Deva: Decoupled video-action model with physical guidance for robot policy learning. *arXiv preprint arXiv:2607.24159*, 2026. 2, 3
- [50] Tianle Zhang, Zhihao Yuan, Dafeng Chi, Peidong Liu, Dongwei Li, Kejun Hu, Likui Zhang, Junnan Nie, Ziming

- Wei, Zengjue Chen, et al. Joyai-ra 0.1: A foundation model for robotic autonomy. *arXiv preprint arXiv:2604.20100*, 2026. [5](#)
- [51] 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*, 2026. [6](#), [7](#)
- [52] Haoyu Zhao, Xingyue Zhao, Siteng Huang, Xin Li, Deli Zhao, and Zhongyu Li. Rynnworld-4d: 4d embodied world models for robotic manipulation. *arXiv preprint arXiv:2607.06559*, 2026. [3](#)
- [53] Yupeng Zheng, Xiang Li, Songen Gu, Yuhang Zheng, Shuai Tian, Weize Li, Linbo Wang, Chaoyue Li, Qichao Zhang, Haoran Li, et al. Gift: Guided intermediate feature training via action-oriented structural supervision for robotic manipulation. *arXiv preprint arXiv:2609.04193*, 2026. [2](#), [3](#), [6](#), [7](#)