M-WAM: MOTION-CENTRIC WORLD ACTION MODEL

Anonymous authors
Paper under double-blind review

ABSTRACT

World-action models learn robot control alongside future prediction, but their predictive objectives commonly emphasize visual appearance. We investigate dense 3D scene motion as a supervision target for action learning. We introduce M-WAM, which predicts pixel-aligned 3D point trajectories organized into image-like motion maps. A pretrained image-generation backbone models these trajectories, while motion and action objectives jointly optimize the scene features used by the policy. In training, we reconstruct dense trajectories on demand from the per-frame simulator geometry. This avoids precomputing every frame pair and reduces export and storage costs from quadratic to linear in episode length. At deployment, the policy generates actions directly without predicting future motion. In RoboTwin ablations, 3D motion improves success by 10.9 percentage points over last-frame RGB prediction and 6.9 points over 2D motion. M-WAM's mean success rate on LIBERO-Plus is 1.7 points higher than ImageWAM's. Across three real-robot tasks, it achieves twice the mean success rate of LingBot-VA. These results support explicit 3D motion prediction as an effective training objective for manipulation policies. All code and data will be publicly released.

1 INTRODUCTION

Vision–language–action (VLA) models transfer visual and language priors to robot control (Brohan et al., 2023; Kim et al., 2024; Black et al., 2024; Bjorck et al., 2025; Shukor et al., 2025; Team et al., 2025). Video-based policies incorporate visual plans (Du et al., 2023; Zhou et al., 2024; Du et al., 2024; Bharadhwaj et al., 2024) or predictive features (Hu et al., 2024; Feng et al., 2025; Pai et al., 2025), while world–action models (WAMs) jointly learn future observation prediction and action generation (Li et al., 2025b; Zhu et al., 2025; Ye et al., 2026; Cheang et al., 2024; Cen et al., 2025b). Predictive co-training can benefit control without generating future observations at deployment (Yuan et al., 2026), making the prediction target a central design choice.

Recent WAMs predict future RGB observations, visual latents (Li et al., 2026a; Bi et al., 2025), or edited images (Zhang et al., 2026). These targets combine object motion with texture, lighting, and background appearance. Visual reconstruction leaves 3D point displacement implicit. Yet a small displacement relative to the gripper can determine manipulation success while leaving most pixels unchanged. This motivates our question: *can dense 3D scene motion provide a more effective prediction target for action learning?*

Point trajectories explicitly represent scene displacement (Yuan et al., 2024; Huang et al., 2026; Lee et al., 2026), including motion along the viewing direction and spatial relations between objects and the gripper. Prior models incorporate geometry through depth (Zhen et al., 2025; Guo et al., 2026), pointmaps (Xu et al., 2026), future-depth readouts (Li et al., 2026b), or trajectory-based feature alignment (Yang et al., 2026). We instead make **dense 3D trajectories the generative future-prediction target in joint motion–action learning**.

We introduce M-WAM, which jointly learns dense 3D motion prediction and robot action generation (Figure 1(d)). Pixel-aligned, image-like motion maps let a pretrained PixelDiT (Yu et al., 2025) backbone model object and gripper trajectories. Each source pixel stores the future 3D trajectory of its corresponding scene point. Motion and action objectives jointly optimize current-scene features, which the policy combines with language and proprioception to generate actions without predicting future motion at deployment.

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

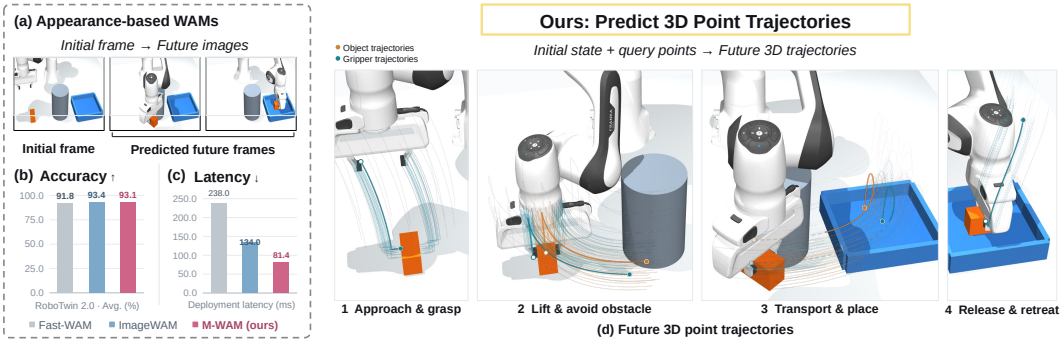


Figure 1: **From appearance prediction to 3D motion supervision.** (a) Future-image prediction in appearance-based WAMs. (b,c) Mean RoboTwin 2.0 success and model forward-pass latency for Fast-WAM (Yuan et al., 2026), ImageWAM (Zhang et al., 2026), and M-WAM (Tables 1 and 7). (d) Illustration of object (orange) and gripper (teal) trajectories through approach, obstacle avoidance, placement, and retreat.

We reconstruct trajectory targets on demand from per-frame simulator geometry under a piecewise-rigid scene model. The same records support different source frames and temporal offsets, reducing simulator export and offline storage from quadratic to linear in episode length compared with pre-computing all frame pairs.

Experiments on RoboTwin 2.0 (Chen et al., 2025), LIBERO (Liu et al., 2023), LIBERO-Plus (Fei et al., 2025), and three real-robot tasks evaluate both the prediction target and the resulting policy. M-WAM achieves competitive results on standard RoboTwin and LIBERO evaluations and exceeds ImageWAM by 1.7 points on LIBERO-Plus. Figure 1(b,c) summarizes RoboTwin success and model forward-pass latency. We further evaluate motion–action consistency and robustness to camera shifts.

Our contributions are threefold:

◇ **Motion-centric world–action modeling.** We introduce M-WAM, which represents future scene change as pixel-aligned, source-frame 3D point trajectories. Motion prediction shapes the scene representation used for action generation without requiring generated motion at inference.

◇ **Scalable 3D motion supervision.** We reconstruct requested trajectories on demand from per-frame geometry under a piecewise-rigid scene model, avoiding pairwise flow rendering and keeping simulator export and offline storage linear in trajectory length.

◇ **Extensive validation.** In RoboTwin ablations, 3D motion improves success by 10.9 percentage points over last-frame RGB and 6.9 over 2D motion. M-WAM also exceeds ImageWAM's mean success on LIBERO-Plus and doubles LingBot-VA's across three real-robot tasks.

2 Related Work

2.1 Vision–Language–Action and World–Action Models

VLA models adapt vision–language backbones to robot control (Brohan et al., 2023; Kim et al., 2024; Black et al., 2024; Intelligence et al., 2025; Bjorck et al., 2025; Wen et al., 2025); video-based policies incorporate generated plans (Du et al., 2023; Zhou et al., 2024; Liang et al., 2024; Ko et al., 2024) or predictive features (Feng et al., 2025; Hu et al., 2024). WAMs jointly learn future visual dynamics and actions (Li et al., 2025b; Zhu et al., 2025; Ye et al., 2026; Cen et al., 2025b;a; Won et al., 2025). Related approaches transfer video-generation pretraining to control (Wu et al., 2023; Cheang et al., 2024; Kim et al., 2026; Liang et al., 2025). Latent prediction also supervises policy representations (Zheng et al., 2025b; Sun et al., 2026), while predicted visual goals guide action generation (Zhao et al., 2025). Fast-WAM retains the benefits of predictive co-training without generating future observations at deployment (Yuan et al., 2026). ImageWAM replaces video generation with instruction-conditioned image editing and conditions actions on editing representa-

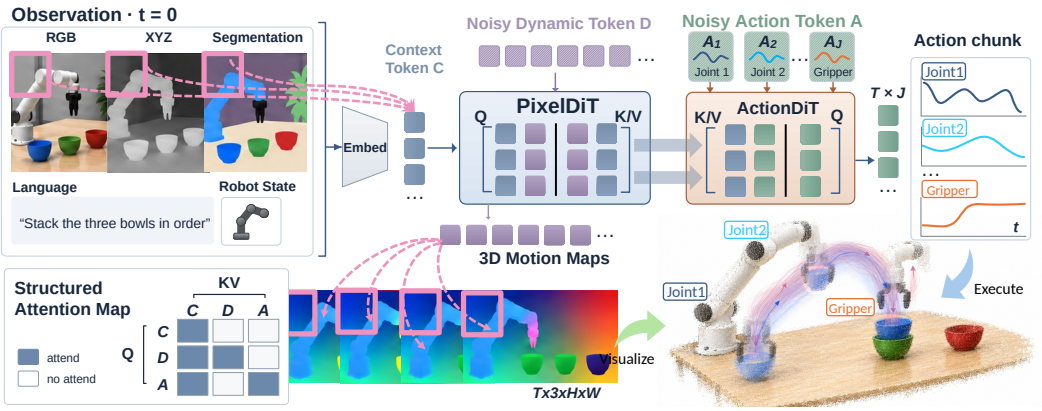


Figure 2: **M-WAM architecture.** M-WAM takes current RGB, XYZ, and segmentation observations, a language instruction, and the robot state as inputs. PixelDiT predicts pixel-aligned 3D motion maps, with T future 3D displacement vectors stored at each source pixel. ActionDiT uses J tokens, one per control dimension; each token encodes a full T-step trajectory, producing a $T \times J$ action chunk. The attention mask allows dynamic (D) and action (A) tokens to read context (C) tokens, while preventing action tokens from reading dynamic tokens.

tions (Zhang et al., 2026). Their prediction targets emphasize visual appearance or latent change; M-WAM instead predicts dense 3D motion to train scene features for action generation.

2.2 3D Motion and Geometric World Modeling

Policies incorporate spatial and motion cues through spatial encoding (Qu et al., 2025; Yuan et al., 2025), visual traces (Zheng et al., 2025a), and feature alignment (Li et al., 2025a). General Flow uses dense motion as a reusable affordance representation (Yuan et al., 2024). TesserAct predicts RGB, depth, and surface-normal sequences (Zhen et al., 2025), PointWorld models manipulation dynamics through 3D point flow (Huang et al., 2026), and Kinema4D uses kinematic 4D pointmaps (Xu et al., 2026). These methods model geometry or motion without using dense trajectories as the prediction target of a jointly trained world–action policy.

Within WAMs, WAM4D adds geometric supervision through spatial register tokens (Li et al., 2026b), X-WAM adds RGB-D prediction and 3D reconstruction (Guo et al., 2026), and 4D-WAM aligns features with trajectory fields while retaining visual prediction (Yang et al., 2026). μ_0 predicts sparse semantic 3D interaction traces for a separate action expert (Lee et al., 2026). M-WAM instead uses dense, pixel-aligned 3D trajectories as the generative prediction target, rather than an auxiliary geometric readout, feature-alignment signal, or sparse semantic trace. Motion and action objectives jointly train the scene representation used by the policy.

3 Method

3.1 Overview and Problem Formulation

We consider language-conditioned robotic manipulation from current multiview observations. At time t, the model receives

$$O_t = \{I_t^v, \mathbf{P}_t^v, \mathbf{S}_t^v\}_{v \in \mathcal{V}}, \tag{1}$$

where $\mathcal{V}$ is the set of available views, and I_t^v, $\mathbf{P}_t^v$, and $\mathbf{S}_t^v$ denote the RGB observation, per-pixel XYZ coordinates, and segmentation encoding of view v, respectively. The XYZ coordinates are expressed in the source reference frame defined in Section 3.2. The formulation does not assume a fixed number or semantic type of cameras. Together with the robot proprioceptive state $\mathbf{q}_t$ and task instruction l, the policy predicts an action chunk

$$\mathbf{A}_t = (\mathbf{a}_t, \ldots, \mathbf{a}_{t+H-1}) \in \mathbb{R}^{H \times J}, \qquad \mathbf{A}_t \sim \pi_\theta(\cdot \mid O_t, l, \mathbf{q}_t), \tag{2}$$

where H is the action horizon and J is the number of control dimensions.

M-WAM jointly learns future scene motion and robot action from the current scene (Figure 2). Let $Z_C = E_C(O_t)$ denote condition tokens extracted from the current observations. The dynamics (D) and action (A) branches predict flow-matching velocity fields for a noisy 3D motion target and a noisy action chunk at noise levels σ_D and σ_A, respectively:

$$\widehat{\mathbf{v}}_D = F_D\Big(\widetilde{\mathbf{M}}_{t,\sigma_D}, Z_C, l, \sigma_D\Big), \qquad \widehat{\mathbf{v}}_A = F_A\Big(\widetilde{\mathbf{A}}_{t,\sigma_A}, Z_C, l, \mathbf{q}_t, \sigma_A\Big). \tag{3}$$

The action branch does not consume a generated motion map. Instead, motion prediction serves as a world-modeling objective that shapes the condition representation on which action generation relies. At deployment, actions are generated directly from the current observations, language instruction, and proprioceptive state.

3.2 Source-Frame 3D Motion Maps

We represent future scene change with pixel-aligned 3D point trajectories. For each sampled source frame t, we designate one observation view $r \in \mathcal{V}$ as the reference view and fix its camera coordinate system at time t as the source reference frame $\mathcal{F}_t^{\mathrm{ref}}$. All current XYZ maps and future 3D positions in the sample are expressed in this frame. No shared simulator coordinate convention is required across platforms.

For source pixel p in view v, let $\mathbf{x}^v_{t\to s}(p) \in \mathbb{R}^3$ denote the position, expressed in $\mathcal{F}_t^{\mathrm{ref}}$, of the physical point selected by (v, p) at source time t when that point is transported to time s. The input XYZ maps are expressed in the same source reference frame. For a sampled clip, let $\Delta_t = (\delta_1, \ldots, \delta_K)$ denote its ordered sequence of positive future offsets. These offsets may be fixed within a training configuration or changed across temporal settings without changing the stored simulation data. The motion target is

$$\mathbf{M}^v_t(p, k) = \mathbf{x}^v_{t\to t+\delta_k}(p) - \mathbf{x}^v_{t\to t}(p), \qquad k = 1, \ldots, K. \tag{4}$$

The notation anchors point identity at the source frame: p does not denote a pixel in a future observation. Under the piecewise-rigid scene model used to construct simulation supervision, the same physical point can be transported geometrically to future times even when it becomes occluded or leaves the camera view. The supervision therefore tracks represented source points through future occlusion.

For each view, the K displacement vectors are stored at their source-pixel locations, yielding

$$\mathbf{M}^v_t \in \mathbb{R}^{h_v \times w_v \times 3K}. \tag{5}$$

Here, h_v and w_v are the image height and width of view v. Each output pixel represents the future 3D trajectory of its corresponding source point, while patch-level dynamic tokens encode full-horizon motion over their associated spatial regions. We use $\mathbf{M}_t$ to denote the collection of per-view motion maps in one training sample; operations on $\mathbf{M}_t$ are applied to all available views. XYZ inputs and 3D motion targets are standardized independently using training-set statistics for each dataset. This routine normalization accommodates differences in coordinate ranges and motion scales across platforms.

3.3 Motion-Supervised World–Action Model

M-WAM uses an image-generation backbone for 3D motion prediction and a diffusion-transformer action branch for control. The image-like motion maps allow the dynamics branch to build on a pretrained PixelDiT (Yu et al., 2025). We adapt the input embeddings for RGB, normalized XYZ, segmentation, and noisy motion, together with the output head for the $3K$-channel motion target. The remaining backbone components follow PixelDiT.

3.3.1 Asymmetric Condition–Dynamics Attention

Within the dynamics branch, condition tokens Z_C and noisy dynamic tokens Z_D form a joint sequence

$$Z_{CD} = [Z_C; Z_D], \tag{6}$$

but use asymmetric masked self-attention. The subscripts C, D, and A identify condition, dynamic, and action tokens, respectively; Q, K, and V denote their query, key, and value projections. Suppressing the layer index, the attention sublayer computes

$$Z_C^+ = \mathrm{Attn}(Q_C, K_C, V_C), \qquad Z_D^+ = \mathrm{Attn}(Q_D, [K_C; K_D], [V_C; V_D]). \tag{7}$$

These equations describe the attention outputs; other transformer operations are omitted. Condition tokens therefore remain independent of noisy dynamic tokens in the forward pass, whereas dynamic tokens use the current scene to predict future motion. Because Z_D^+ depends on keys and values computed from Z_C, the motion-prediction loss backpropagates through the condition representation. The asymmetric mask thus lets motion supervision shape scene features without allowing sampled or noisy future motion to contaminate them.

3.3.2 Trajectory-Wise Action Representation

The action branch represents an action chunk by control dimension rather than by individual time step. For the j-th control dimension, one token encodes its complete H-step trajectory:

$$\mathbf{z}_{A,j} = E_A\left(\widetilde{\mathbf{A}}_{t,\sigma_A}[:, j]\right) + \mathbf{e}_j, \qquad j = 1, \ldots, J, \tag{8}$$

where $E_A : \mathbb{R}^H \to \mathbb{R}^{d_{\text{emb}}}$ is a learnable projection, d_{emb} is the action-token embedding dimension, and $\mathbf{e}_j$ identifies the control dimension. Let $Z_A = [\mathbf{z}_{A,1}; \ldots; \mathbf{z}_{A,J}]$. Self-attention among these tokens models coordination across control dimensions while each token retains the full temporal evolution of one control variable. At corresponding layers, action queries read action and condition features:

$$Z_A^+ = \text{Attn}\left(Q_A, [K_C; K_A], [V_C; V_A]\right). \tag{9}$$

The action branch does not read dynamic tokens. Its scene context is provided by condition features jointly optimized by the motion and action objectives. A final projection maps the action-token features to an $H \times J$ flow-matching velocity prediction. Language, proprioception, and noise-level conditioning are applied throughout the action branch.

3.3.3 Training Strategies

Following Fast-WAM (Yuan et al., 2026), we train both branches with flow matching (Lipman et al., 2022), independently perturbing the motion and action targets and minimizing the sum of their mean squared velocity errors.

High-noise-biased sampling. Motion maps and action chunks contain strong low-frequency trajectory structure. During inference, the earliest denoising steps start close to pure noise and establish the coarse global layout of the predicted trajectories. We therefore reweight the noise-level distribution for both branches toward this regime. Specifically, the training noise-level distribution p_σ^{train} satisfies $\Pr_{\sigma \sim p_\sigma^{\text{train}}}[\sigma \in [0.93, 1]] = 0.20$. The remaining probability mass is renormalized over lower noise levels. This increases supervision where the model must recover global motion and action structure from nearly pure noise, while retaining coverage of lower-noise refinement steps.

Robust geometric and segmentation conditioning. The XYZ conditions are constructed from depth, but we perturb the input depth during training before back-projection and transformation to the source reference frame. Only the conditioning geometry is perturbed; motion supervision is constructed from the unperturbed recorded geometry. This exposes the policy to imperfect geometric observations while preserving accurate trajectory targets. Segmentation conditions are generated automatically from RGB observations using Grounding DINO and SAM 2 rather than simulator-provided ground-truth masks.

3.4 Scalable Motion Supervision from Per-Frame Geometry

The motion target in Eq. (4) can be generated without rendering pair-specific flow. Simulator-specific exporters convert each episode into a common per-frame representation containing metric depth, camera intrinsics and poses, per-pixel rigid-entity associations, and the poses of rigid objects and robot links. Static scene regions are represented as entities with fixed poses. The entity associations are used only to construct supervision and are not provided to the policy.

Consider a source pixel p observed by camera v at time t, and let $b = b_t^v(p)$ denote its associated rigid entity. For readability, we omit (v, p) from intermediate point variables. Let $T_v^{\text{cam}}(t)$ and $T_r^{\text{cam}}(t)$ map coordinates from the source camera and the designated reference camera, respectively, to simulator coordinates, and let $T_b^{\text{ent}}(s)$ map entity-local coordinates to simulator coordinates at

time s. All transforms are in $\mathrm{SE}(3)$; homogeneous-coordinate notation is omitted. We first back-project the source pixel to simulator coordinates,

$$\mathbf{y}_t = T_v^{\mathrm{cam}}(t)\ \mathrm{BackProj}\left(p, d_t^v(p); \mathbf{K}_t^v\right), \tag{10}$$

where $d_t^v(p)$ and $\mathbf{K}_t^v$ are the stored depth and camera intrinsics. The same physical point is then transported with its rigid entity and expressed in the reference-camera coordinate system fixed at source time t:

$$\mathbf{x}_{t\to s} = T_r^{\mathrm{cam}}(t)^{-1} T_b^{\mathrm{ent}}(s) T_b^{\mathrm{ent}}(t)^{-1} \mathbf{y}_t. \tag{11}$$

Here, $T_b^{\mathrm{ent}}(t)^{-1}$ maps the source point into the entity frame, $T_b^{\mathrm{ent}}(s)$ moves it to time s, and $T_r^{\mathrm{cam}}(t)^{-1}$ expresses the result in $\mathcal{F}_t^{\mathrm{ref}}$. Thus, $\mathbf{x}_{t\to s}$ is precisely $\mathbf{x}_{t\to s}^v(p)$ defined in Section 3.2. Setting $s = t + \delta_k$ gives the motion target $\mathbf{M}_t^v(p,k) = \mathbf{x}_{t\to t+\delta_k} - \mathbf{x}_{t\to t}$. Because the point is propagated through entity motion rather than recovered from a target image, its trajectory remains defined through future occlusion and does not depend on target-frame visibility.

For an N-frame trajectory, pre-rendering pairwise flow requires $O(NK)$ targets for a fixed set of K offsets and can grow to $O(N^2)$ when arbitrary source–target pairs must be supported. Our pipeline instead stores one geometry record per frame and constructs only the motion targets requested by the dataloader. Simulator-side export and offline geometry storage therefore remain $O(N)$, independent of the temporal offsets used for training or ablation.

4 Experiments

We evaluate dense 3D motion supervision through manipulation benchmarks, predictive-target comparisons, visual perturbations, and motion diagnostics. Architectural and training ablations examine which design choices contribute to action performance and inference efficiency.

4.1 Experimental Setup

Sim & Real Benchmark. We evaluate on RoboTwin 2.0 (Chen et al., 2025), LIBERO (Liu et al., 2023), LIBERO-Plus (Fei et al., 2025), and an ARX5 dual-arm robot. RoboTwin training uses 20,000 demonstrations across 50 tasks, with 200 episodes per task under each of the Clean and Randomized conditions. All RoboTwin closed-loop success evaluations, including ablations, follow the Fast-WAM protocol with 10,000 test episodes: 100 per task per condition, evaluated with RoboTwin's task-specific success checks. LIBERO covers the Spatial, Object, Goal, and Long suites. LIBERO-Plus evaluates the same trained policies under seven perturbation categories, using only the original LIBERO demonstrations for training. The ARX5 tasks are block insertion (T1), cube stacking (T2), and drawer loading (T3), each with 200 training demonstrations and 20 evaluation trials under randomized initial object placements. We construct real-robot 3D motion supervision from depth and point trajectories estimated using VGGT-Ω and SpatialTrackerV2.

Baseline & Metrics. We compare with Fast-WAM (Yuan et al., 2026), Motus (Bi et al., 2025), LingBot-VA (Li et al., 2026a), and ImageWAM (Zhang et al., 2026). Table 1 reports unweighted averages of the displayed component scores. ImageWAM reports 2,500 Clean and 25,000 Randomized RoboTwin training demonstrations, whereas M-WAM uses the balanced 20,000-episode set described above. The LIBERO-Plus comparison uses ImageWAM's FLUX.2 4B variant. We report success rate (SR) in percent and absolute differences in percentage points. For motion evaluation, Avg. δ (Doersch et al., 2022) averages tracking accuracy over error thresholds and future frames, using ground-truth-visible points. Endpoint error (EPE) measures the mean Euclidean distance between predicted and demonstration endpoints. Motion–Action Discrepancy (MAD) measures the same distance using endpoints induced by executing the corresponding action chunk as the reference. EPE and MAD include all queried points, irrespective of endpoint visibility. Both distances are measured in image-plane pixels; lower is better. For each fixed model, motion metrics are reported as mean $\pm$ standard deviation over repeated sampling with different random noise initializations.

4.2 Manipulation Performance

Simulation performance. Table 1 compares performance across the three simulation benchmarks. On RoboTwin 2.0 Clean, M-WAM achieves the highest success rate, exceeding ImageWAM by 1.70

Table 1: **Success on simulation benchmarks (%, ↑).** Best and second-best results are bold and underlined; shaded rows denote our method. "-" denotes unavailable results.

	RoboTwin 2.0 (Chen et al., 2025)			LIBERO (Liu et al., 2023)					LIBERO-Plus (Fei et al., 2025)							
Method	Clean	Rand.	Avg.	Spa.	Obj.	Goal	Long	Avg.	Cam.	Robot	Lang.	Light	Bkg.	Noise	Layout	Avg.
Fast-WAM	91.88	91.78	91.83	98.2	**100.0**	97.0	95.2	97.6	16.4	44.5	68.9	78.2	53.7	37.7	60.7	51.4
Motus	88.66	87.02	87.84	96.8	99.8	96.6	97.6	97.7	-	-	-	-	-	-	-	-
LingBot-VA	92.90	91.50	92.20	**98.5**	99.6	97.2	**98.5**	**98.5**	13.8	35.7	24.1	28.6	20.7	21.4	13.8	22.6
ImageWAM	93.20	**93.56**	**93.38**	97.2	99.2	**98.8**	98.4	98.4	80.8	50.3	91.4	**98.1**	85.5	**93.8**	80.5	82.9
M-WAM (ours)	**94.90**	91.30	93.10	97.8	99.0	97.9	**98.5**	98.3	**81.0**	**59.9**	**93.8**	94.3	**89.9**	90.2	**83.0**	**84.6**

Table 2: **Denoising steps and SR (%).**

Denoising steps	1	2	3	5	10	20
SR (%)	88.21	89.57	91.35	**93.10**	92.03	91.06

Table 3: **Predictive targets.**

Target	Success Rate % ↑	MAD ↓
Last-frame RGB	82.17	4.23 ± 0.89
2D motion	86.22	2.42 ± 0.41
3D motion (ours)	**93.10**	**1.15** ± 0.07

Table 4: **Predicted and executed motion.** Avg. δ (%); EPE and MAD (px), mean ± SD.

	Prediction		Alignment	
Method	Avg. δ ↑	EPE ↓	Avg. δ ↑	MAD ↓
Fast-WAM	91.26 ± 0.27	1.48 ± 0.04	**91.66** ± 0.35	1.26 ± 0.06
Motus	91.55 ± 0.21	1.67 ± 0.11	90.65 ± 0.26	1.32 ± 0.06
LingBot-VA	89.26 ± 0.33	3.85 ± 0.09	88.19 ± 0.64	5.32 ± 0.55
ImageWAM	N/A	3.51 ± 0.06	N/A	1.76 ± 0.07
M-WAM	**93.06** ± 0.20	**1.01** ± 0.02	91.64 ± 0.30	**1.15** ± 0.07

percentage points. Averaged over Clean and Randomized conditions, M-WAM reaches 93.10%, 1.27 points above Fast-WAM and 0.28 points below ImageWAM. On LIBERO, its 98.3% average is within 0.2 points of the highest-scoring baseline, LingBot-VA.

The larger visual and layout variations in LIBERO-Plus produce clearer differences between methods. M-WAM achieves an average of 84.6%, improving over ImageWAM by 1.7 points and Fast-WAM by 33.1 points. It exceeds ImageWAM in five of the seven categories, with the largest gains on Robot (+9.6 points) and Background (+4.4 points). ImageWAM retains higher scores on Light and Noise. Thus, methods with similar success on standard LIBERO exhibit substantially different performance under the broader LIBERO-Plus evaluation.

Real-robot performance. M-WAM achieves the highest success rate on all three ARX5 tasks (Table 5). It succeeds in 8, 11, and 9 of the 20 trials for block insertion, cube stacking, and drawer loading, respectively. On each task, this is four more successful trials, or 20 percentage points, than the strongest baseline. Its mean success rate is 46.7%, twice the 23.3% achieved by LingBot-VA, the strongest baseline on average.

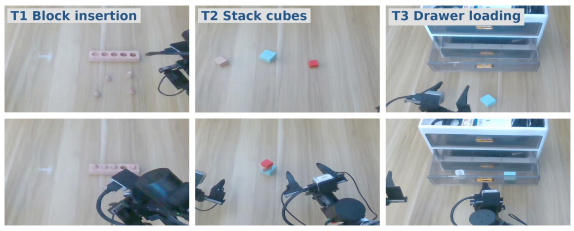
Inference efficiency. We measure model forward-pass costs on RoboTwin inputs using an NVIDIA RTX 4090. Latency covers the model computation needed to produce an action chunk from prepared inputs; observation preprocessing is outside the timed region. M-WAM has the lowest compute cost, latency, and peak memory among the methods in Table 7. Its forward pass requires 673.84 GFLOPs and 81.39 ms, with 3.54 GB peak memory. Fast-WAM and ImageWAM take 2.92 and 1.65 times as long, respectively.

4.3 Which Future Representation Supports Action?

We isolate the predictive target by keeping the input modalities, model, and training setup fixed and comparing last-frame RGB, 2D motion, and 3D motion (Table 3). Dense 3D motion achieves the highest success rate among these targets: 93.10%, improving over last-frame RGB by 10.93 percentage points and 2D motion by 6.88 points. Prediction–execution agreement improves alongside task success: MAD decreases from 4.23 pixels for RGB to 2.42 for 2D motion and 1.15 for 3D motion. Relative to 2D motion, the 3D target reduces MAD by 52.5%, supporting its use for learning both successful actions and motion predictions that agree with their execution.

Table 5: **Real-robot success rate.** ARX5 dual-arm robot; 20 trials per task.

Method	T1	T2	T3	Avg.
Motus (Bi et al., 2025)	0%	15%	5%	6.7%
Fast-WAM (Yuan et al., 2026)	20%	25%	20%	21.7%
ImageWAM (Zhang et al., 2026)	10%	20%	20%	16.7%
LingBot-VA (Li et al., 2026a)	10%	35%	25%	23.3%
M-WAM (ours)	**40%**	**55%**	**45%**	**46.7%**



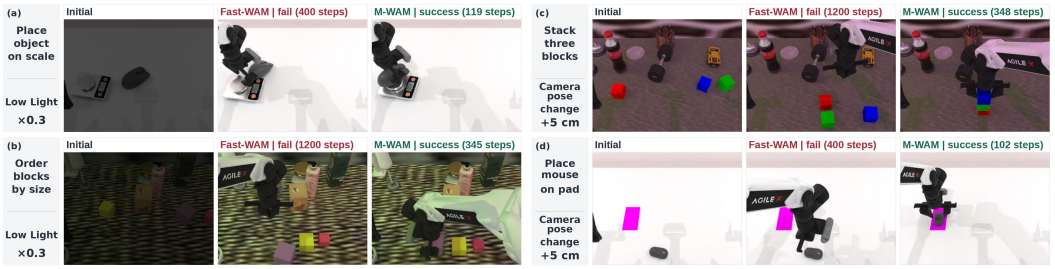


Figure 3: **Selected rollouts under visual perturbations.** (a,b) Brightness $\times 0.3$. (c,d) Camera raised by 5 cm. Each triplet shows the initial observation, a failed Fast-WAM rollout, and a successful M-WAM rollout from the same initial state. Headers report rollout step counts.

Method	Ref.	Camera change	Low light
Fast-WAM	91.83	52.64 (–39.19)	80.46 (–11.37)
Motus	87.84	41.45 (–46.39)	76.10 (–11.74)
LingBot-VA	92.20	35.67 (–56.53)	85.11 (–7.09)
ImageWAM	**93.38**	47.58 (–45.80)	86.36 (–7.02)
M-WAM (ours)	93.10	**86.46** (–6.64)	**87.01** (–6.09)

Table 6: **RoboTwin robustness (SR, %).**

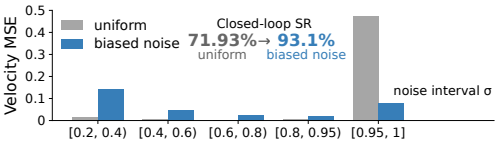


Figure 4: **Noise sampling ablation.**

4.4 Robustness to Camera and Lighting Changes

We test robustness by modifying observations in the original RoboTwin evaluation episodes while retaining the same trained policies, initial states, and task dynamics. In Table 6, Ref. denotes performance on the unmodified test conditions. For the perturbed evaluations, we raise the camera rig by 5 cm or scale image brightness by 0.3. After the camera shift, we recompute the calibration used for depth back-projection and express XYZ in the moved head-camera coordinate frame. Parenthesized values report changes in success rate, in percentage points, relative to each method's unperturbed reference.

M-WAM achieves the highest success rate and the smallest decrease under both perturbations. Under the camera shift, its success decreases by 6.64 percentage points to 86.46%, whereas the baselines lose 39.19–56.53 points. M-WAM exceeds the strongest baseline under this shift, Fast-WAM, by 33.82 points. Under low light, M-WAM achieves 87.01%, exceeding ImageWAM, the strongest baseline, by 0.65 points. Its decrease of 6.09 points is also smaller than the baseline drops of 7.02–11.74 points.

Figure 3 illustrates the effect of these perturbations on individual rollouts. Across the four selected cases, M-WAM completes the task while Fast-WAM fails from the same initial state, under either reduced brightness or a shifted camera.

4.5 Do Predicted and Executed Motions Agree?

For prediction accuracy, we compare generated motion with held-out demonstration trajectories. For prediction–execution alignment, we start from the same simulator state and execute a single generated action chunk over the prediction horizon, without replanning. Execution endpoints are obtained from simulator geometry. Queries cover all source-image points. We project simulator trajectories and predicted 3D trajectories into the image plane, and use CoTracker3 (Karaev et al.,

Table 7: **Inference cost.**

Method	GFLOPs	ms	GB
Fast-WAM	2849.30	238.00	13.66
ImageWAM	4838.24	133.96	10.13
M-WAM	**673.84**	**81.39**	**3.54**

Table 8: **Attention ablation.**

Attn.	Future	SR	ms
$C \leftrightarrow D$	Joint	89.14	128.44
$C \rightarrow D$	Omit	**93.10**	**81.39**

Table 9: **Action tokenization.**

Tokens	SR	MAD ↓
Time-wise	91.52	2.72 ± 0.41
Trajectory-wise	**93.10**	**1.15** ± 0.07

2024) to extract point correspondences from RGB predictions. For ImageWAM, CoTracker3 establishes correspondences between the current and predicted endpoint frames; we therefore report EPE and MAD, with full-horizon Avg. δ marked N/A. For policies deployed without future prediction, we enable the predictive branch for this diagnostic while retaining the same action output.

M-WAM achieves the highest prediction Avg. δ and the lowest prediction EPE in Table 4. Relative to Fast-WAM, its full-horizon prediction accuracy improves by 1.80 percentage points, and its endpoint EPE decreases from 1.48 to 1.01 pixels, a 31.8% reduction.

For alignment with executed actions, M-WAM achieves the lowest MAD: 1.15 pixels, compared with 1.26 for Fast-WAM, an 8.7% reduction. Their full-horizon Avg. δ scores are nearly identical (91.64% and 91.66%). M-WAM therefore improves motion forecasting and endpoint agreement with executed actions, while full-horizon alignment remains comparable to Fast-WAM.

4.6 Architectural and Training Ablations

We evaluate the architectural and training choices on the full RoboTwin 2.0 test set, using the same 10,000-episode protocol as the main evaluation.

Attention ablation. The attention structure determines whether action inference depends on future denoising. With bidirectional attention ($C \leftrightarrow D$), context features depend on noisy dynamic tokens, requiring joint future and action generation. One-way conditioning ($C \rightarrow D$) prevents dynamic tokens from feeding back into the context used by the action branch, allowing future denoising to be omitted. Table 8 compares these two structures with their corresponding inference procedures. The asymmetric model achieves 93.10% success, 3.96 percentage points above the bidirectional model, while reducing latency from 128.44 to 81.39 ms (36.6%).

Action tokenization. Trajectory-wise tokenization assigns one token to each of the J control dimensions, with each token encoding a complete H-step trajectory. Time-wise tokenization instead assigns one token to each of the H time steps, encoding a J-dimensional action. In Table 9, trajectory-wise tokenization improves success by 1.58 percentage points and reduces MAD from 2.72 to 1.15 pixels, a 57.7% reduction.

Denoising steps. Five denoising steps achieve the highest success (Table 2). Success rises from 88.21% with one step to 93.10% with five, a gain of 4.89 percentage points. Increasing to 10 or 20 steps does not improve success.

High-noise-biased sampling. Figure 4 plots motion flow-matching velocity MSE by noise level. Uniform sampling produces a sharp error increase at $\sigma \in [0.95, 1]$. Motion is largely smooth across space and time and dominated by low-frequency structure; finer high-frequency details are harder to reconstruct. At low and moderate noise, much of the dominant structure remains recoverable, making these cases relatively easy. Uniform sampling therefore allocates most updates to easier cases, which can leave near-pure-noise denoising undertrained. The biased strategy in Section 3.3.3 reduces high-noise error, although errors at lower noise levels increase. Closed-loop success improves from 71.93% to 93.10%, a gain of 21.17 percentage points.

5 Conclusion

We introduced M-WAM, which jointly learns robot actions and dense 3D point trajectories represented as pixel-aligned motion maps. Motion supervision shapes the scene features used by the policy, enabling direct action generation without predicting future motion at deployment. Trajectories are reconstructed on demand from per-frame geometry, keeping simulator export and offline storage linear in episode length. Target ablations and motion–action diagnostics support this choice of supervision, while simulation and real-robot evaluations demonstrate its utility for manipulation. These findings highlight the prediction target as an important design choice for world–action models and establish dense 3D motion as an effective supervision signal for action learning.

Ethics Statement

This work studies robotic manipulation in simulation and controlled real-robot experiments. We identify no specific ethical concerns associated with the methods, data, or experiments presented in this paper.

Reproducibility Statement

Sections 3 and 4 describe our model, motion-supervision pipeline, datasets, and evaluation protocols. We will open-source all code and data following the public release of this paper.

AI Use Statement

Generative AI tools were used to aid and polish the writing of this manuscript, primarily to improve clarity, fluency, and conciseness.

References

Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2Act: Human video generation in novel scenarios enables generalizable robot manipulation. *arXiv preprint arXiv:2409.16283*, 2024. URL https://arxiv.org/abs/2409.16283.

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, 2025. URL https://arxiv.org/abs/2512.13030.

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. URL https://arxiv.org/abs/2503.14734.

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. URL https://arxiv.org/abs/2410.24164.

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

Jun Cen, Siteng Huang, Yuqian Yuan, Kehan Li, Hangjie Yuan, Chaohui Yu, Bohan Hou, Yuming Jiang, Jiayan Guo, Xin Li, Hao Luo, Fan Wang, Deli Zhao, and Hao Chen. RynnVLA-002: A unified vision-language-action and world model. *arXiv preprint arXiv:2511.17502*, 2025a. URL https://arxiv.org/abs/2511.17502.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. WorldVLA: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025b. URL https://arxiv.org/abs/2506.21539.

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, Hanbo Zhang, and Minzhao Zhu. GR-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024. URL https://arxiv.org/abs/2410.06158.

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, 2025. URL https://arxiv.org/abs/2506.18088.

Carl Doersch, Ankush Gupta, Larisa Markeeva, Adrià Recasens, Lucas Smaira, Yusuf Aytar, João Carreira, Andrew Zisserman, and Yi Yang. TAP-Vid: A benchmark for tracking any point in a video, 2022. URL https://arxiv.org/abs/2211.03726.

Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B. Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation, 2023. URL https://arxiv.org/abs/2302.00111.

Yilun Du, Sherry Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Brian Ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B. Tenenbaum, Leslie Pack Kaelbling, Andy Zeng, and Jonathan Tompson. Video language planning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=9pKtcJcMP3.

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: In-depth robustness analysis of vision-language-action models, 2025. URL https://arxiv.org/abs/2510.13626.

Yao Feng, Hengkai Tan, Xinyi Mao, Guodong Liu, Shuhe Huang, Chendong Xiang, Hang Su, and Jun Zhu. Vidar: Embodied video diffusion model for generalist bimanual manipulation. *arXiv preprint*, 2025. URL https://arxiv.org/abs/2507.12898.

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

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. URL https://arxiv.org/abs/2412.14803.

Wenlong Huang, Yu-Wei Chao, Arsalan Mousavian, Ming-Yu Liu, Dieter Fox, Kaichun Mo, and Li Fei-Fei. PointWorld: Scaling 3D world models for in-the-wild robotic manipulation. *arXiv preprint arXiv:2601.03782*, 2026. URL https://arxiv.org/abs/2601.03782.

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. URL https://arxiv.org/abs/2504.16054.

Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker3: Simpler and better point tracking by pseudo-labelling real videos, 2024. URL https://arxiv.org/abs/2410.11831.

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

Moo Jin Kim, 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. URL https://arxiv.org/abs/2601.16163.

Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B. Tenenbaum. Learning to act from actionless videos through dense correspondences. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Mhb5fpA1T0.

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. URL https://arxiv.org/abs/2606.13769.

Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial forcing: Implicit spatial representation alignment for vision-language-action model. *arXiv preprint arXiv:2510.12276*, 2025a. URL https://arxiv.org/abs/2510.12276.

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

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. *arXiv preprint arXiv:2503.00200*, 2025b. URL https://arxiv.org/abs/2503.00200.

Ying Li, Xiaobao Wei, Jiajun Cao, Hao Wang, Xiaowei Chi, Chengyu Bai, Qianpu Sun, Jiajun Li, Xiaojie Zhang, Jian Tang, et al. WAM4D: Fast 4D world action model via spatial register tokens. *arXiv preprint arXiv:2606.14048*, 2026b. URL https://arxiv.org/abs/2606.14048.

Junbang Liang, Ruoshi Liu, Ege Ozguroglu, Sruthi Sudhakar, Achal Dave, Pavel Tokmakov, Shuran Song, and Carl Vondrick. Dreamitate: Real-world visuomotor policy learning via video generation. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=InT87E5sr4.

Junbang Liang, Pavel Tokmakov, Ruoshi Liu, Sruthi Sudhakar, Paarth Shah, Rares Ambrus, and Carl Vondrick. Video generators are robot policies. *arXiv preprint arXiv:2508.00795*, 2025. URL https://arxiv.org/abs/2508.00795.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling, 2022. URL https://arxiv.org/abs/2210.02747.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-1939. URL https://papers.nips.cc/paper_files/paper/2023/hash/8c3c666820ea055a77726d66fc7d447f-Abstract-Datasets_and_Benchmarks.html.

Jonas Pai, Liam Achenbach, Victoriano Montesinos, Benedek Forrai, Oier Mees, and Elvis Nava. mimic-video: Video-action models for generalizable robot control beyond VLAs. *arXiv preprint arXiv:2512.15692*, 2025. URL https://arxiv.org/abs/2512.15692.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. SpatialVLA: Exploring spatial representations for visual-language-action model. *arXiv preprint*, 2025. URL https://arxiv.org/abs/2501.15830.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. SmolVLA: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025. URL https://arxiv.org/abs/2506.01844.

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. VLA-JEPA: Enhancing vision-language-action model with latent world model. *arXiv preprint arXiv:2602.10098*, 2026. URL https://arxiv.org/abs/2602.10098.

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

Junjie Wen, Yichen Zhu, Jinming Li, Zhibin Tang, Chaomin Shen, and Feifei Feng. DexVLA: Vision-language model with plug-in diffusion expert for general robot control. *arXiv preprint arXiv:2502.05855*, 2025. URL https://arxiv.org/abs/2502.05855.

John Won, Kyungmin Lee, Huiwon Jang, Dongyoung Kim, and Jinwoo Shin. Dual-stream diffusion for world-model augmented vision-language-action model, 2025. URL https://arxiv.org/abs/2510.27607.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation, 2023. URL https://arxiv.org/abs/2312.13139.

Mutian Xu, Tianbao Zhang, Tianqi Liu, Zhaoxi Chen, Xiaoguang Han, and Ziwei Liu. Kinema4D: Kinematic 4D world modeling for spatiotemporal embodied simulation. *arXiv preprint arXiv:2603.16669*, 2026. URL https://arxiv.org/abs/2603.16669.

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

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, 2026. URL https://arxiv.org/abs/2602.15922.

Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. PixelDiT: Pixel diffusion transformers for image generation, 2025. URL https://arxiv.org/abs/2511.20645.

Chengbo Yuan, Chuan Wen, Tong Zhang, and Yang Gao. General Flow as Foundation Affordance for Scalable Robot Learning, 2024. URL https://arxiv.org/abs/2401.11439.

Tianyuan Yuan, Yicheng Liu, Chenhao Lu, Zhuoguang Chen, Tao Jiang, and Hang Zhao. DepthVLA: Enhancing vision-language-action models with depth-aware spatial reasoning. *arXiv preprint arXiv:2510.13375*, 2025. URL https://arxiv.org/abs/2510.13375.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-WAM: Do World Action Models Need Test-time Future Imagination?, 2026. URL https://arxiv.org/abs/2603.16666.

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?, 2026. URL https://arxiv.org/abs/2606.19531.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, Ankur Handa, Ming-Yu Liu, Donglai Xiang, Gordon Wetzstein, and Tsung-Yi Lin. CoT-VLA: Visual chain-of-thought reasoning for vision-language-action models, 2025. URL https://arxiv.org/abs/2503.22020.

Haoyu Zhen, Qiao Sun, Hongxin Zhang, Junyan Li, Siyuan Zhou, Yilun Du, and Chuang Gan. Learning 4D embodied world models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5337–5347, 2025. URL https://arxiv.org/abs/2504.20995.

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 *The Thirteenth International Conference on Learning Representations*, 2025a.

Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loic Magne, Avnish Narayan, You Liang Tan, Guanzhi Wang, Qi Wang, Jiannan Xiang, Yinzhen Xu, Seonghyeon Ye, Jan Kautz, Furong Huang, Yuke Zhu, and Linxi Fan. FLARE: Robot learning with implicit world modeling, 2025b. URL https://arxiv.org/abs/2505.15659.

Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. RoboDreamer: Learning compositional world models for robot imagination. *arXiv preprint arXiv:2404.12377*, 2024. URL https://arxiv.org/abs/2404.12377.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets, 2025. URL https://arxiv.org/abs/2504.02792.