Learning 4D Geometric Priors for Inference-Efficient World Action Models

Jianjun Zhang[1,2,*] **Jian Zhu**[2,‡] **Taiyi Su**[2] **Chong Ma**[1,2] **Zitai Huang**[1,2]
Yi Xu[2,†] **Hanli Wang**[1,†]

[1]Tongji University [2]AIRC, Midea Group
[†]Corresponding Author [‡]Project Leader
Project Page: meco-wam.github.io

Abstract

World Action Models (WAMs) have shown strong potential for robotic manipulation by jointly modeling visual future dynamics and executable action sequences. However, existing video-action co-training methods primarily optimize appearance-oriented video latents, which may insufficiently capture the temporally evolving geometry required for precise manipulation. We propose **MECo-WAM**, a Multi-Expert Co-Training World Action Model that injects action-relevant 4D geometric priors into video-action representations while preserving the original lightweight inference graph. During training, MECo-WAM combines video and action experts with a lightweight 4D expert supervised by relational targets from a frozen VGGT encoder. Asymmetric expert visibility prevents non-causal shortcuts from auxiliary geometry to action generation. To transfer geometric knowledge into the deployed video-action pathway, we introduce *decayed 4D read-mask attention*, which provides restricted current-frame geometric guidance early in training and progressively removes this dependency. We further propose *action-aware temporal geometric distillation*, which aligns within-frame geometric relations and their temporal evolution while emphasizing visual regions most relevant to robot actions. At deployment, all auxiliary 4D components are removed. Experiments on LIBERO (98.2%), RoboTwin 2.0 (92.6%), and challenging real-world manipulation tasks show that MECo-WAM improves manipulation performance without increasing inference cost.

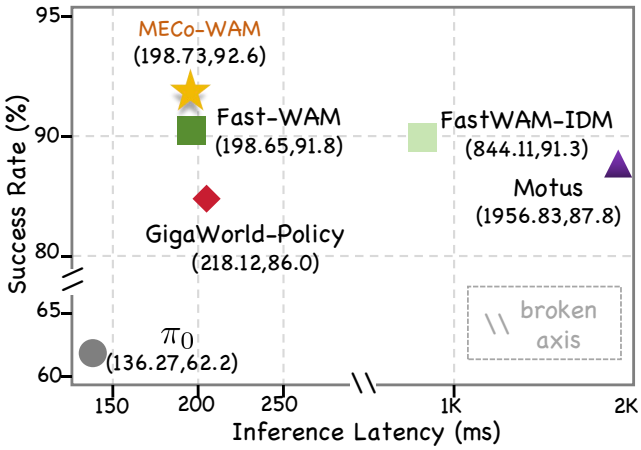


Figure 1: Comparison of MECo-WAM with Baselines in action-chunk inference latency and task success rate on RoboTwin.

Introduction

Robotic manipulation requires a policy to map visual observations and language instructions to precise action trajectories (Hu et al. 2025; Ma et al. 2026; Su et al. 2026; Wang et al. 2026; Su et al. 2025). World action models offer a promising formulation by jointly learning how visual states evolve under interaction and how robot actions should be generated (Ye et al. 2026c; Kim et al. 2026; Bi et al. 2026; Li et al. 2026b). Compared with direct action policies, video-action co-training can provide richer motion and interaction priors, enabling the policy to reason over changes that unfold beyond a single observation (Ye et al. 2026a; Yuan et al. 2026).

Despite this advantage, the visual representation learned by many WAMs remains dominated by appearance-oriented video prediction. A video latent can support plausible future synthesis without explicitly preserving the spatial relations that determine whether a grasp is reachable, whether an object is aligned with a target, or whether contact will cause a stable transition. These relations are not static: they evolve as the robot approaches, contacts, moves, and releases objects. Recent geometry-aware VLA and WAM studies therefore introduce 3D or 4D structure to strengthen spatial grounding for manipulation (Qu et al. 2025; Li et al. 2025a,b; Guo et al. 2026; Li et al. 2026c). Thus, manipulation-oriented WAMs require geometry-aware temporal representations beyond visual plausibility.

A direct approach is to introduce explicit 4D reconstruction or dense geometric prediction into the world action modeling pipeline (Guo et al. 2026; Li et al. 2026c). However, making geometry an explicit deployment-time output increases inference cost and may shift optimization toward geometric reconstruction that is only weakly coupled with action generation. More importantly, generic geometric supervision does not distinguish the visual relations that are causally relevant to the robot's current action. More importantly, generic geometric supervision may overlook action-relevant relations among manipulated objects, target regions,

[*]This work was completed during an internship at Midea AI Research Centers.

and their temporal interactions.

We therefore ask a focused question: *can a world action model acquire action-relevant temporal geometry during training while retaining the same lightweight video-action inference graph at deployment?* Our answer is **MECo-WAM**, a Multi-Expert Co-Training World Action Model. MECo-WAM adds a lightweight 4D expert only during training, where frozen VGGT features (Wang et al. 2025a) supervise temporal geometric prediction alongside video and action denoising. Rather than allowing unrestricted cross-expert communication, the deployed video-action pathway receives only restricted current-frame geometric guidance during early training, while future geometry remains loss-side supervision. This design transfers 4D priors without creating non-causal shortcuts or adding inference-time cost. As shown in Figure 1, this design achieves a strong task success rate on RoboTwin 2.0 (Mu et al. 2025) while keeping action-chunk inference latency low.

To transfer geometry without a permanent 4D dependency, we introduce *decayed 4D read-mask attention*, which exposes only the current-frame geometry token early in training and removes this access before deployment. We further propose *action-aware temporal geometric distillation*, aligning predicted 4D keyframes and their temporal relation changes with frozen VGGT targets while emphasizing action-relevant token pairs.

Our contributions are fourfold:

- We propose **MECo-WAM**, a multi-expert co-training framework that injects action-aware 4D geometric priors into WAM representations while preserving the original lightweight inference graph.
- We introduce **decayed 4D read-mask attention**, which provides early-stage geometric guidance and progressively removes the dependency on 4D tokens before deployment.
- We propose **action-aware temporal geometric distillation**, which aligns within-frame geometric relations and their temporal evolution while emphasizing action-relevant visual regions, enabling task-conditioned geometry learning.
- Extensive experiments on LIBERO, RoboTwin 2.0, and real-world manipulation tasks demonstrate consistent gains in task success and execution efficiency.

Related Work

World Action Models

World action models bridge two complementary embodied learning paradigms: policies that map observations and instructions to executable actions, and world models that predict how the environment evolves under interaction. Direct VLA policies, including the π model family (Black et al. 2024; Intelligence et al. 2025) and OpenVLA (Kim et al. 2025), provide strong reactive observation-to-action baselines. Recent WAMs adapt pretrained video priors or unified multimodal architectures to jointly model future observations, latent dynamics, and action chunks. DreamZero (Ye et al. 2026c), Mimic-Video (Pai et al. 2025), Cosmos Policy (Kim et al. 2026), Motus (Bi et al. 2026), and LingBot-VA (Li et al. 2026b) show that video-based temporal supervision can improve policy learning beyond direct behavior cloning by exposing the model to physical evolution and action-conditioned scene changes.

Some systems improve practical execution through causal attention, multi-chunk prediction, caching, asynchronous denoising, or action-centered interfaces (Ye et al. 2026a; Xu et al. 2026; Team et al. 2026; Guo et al. 2026). Fast-WAM (Yuan et al. 2026) makes this deployment-oriented view explicit: future video prediction remains useful during training, while test-time action generation can proceed without explicit future imagination. MECo-WAM follows this efficient WAM principle but studies a different source of supervision. Rather than adding rollout to the deployed policy, it uses a training-only 4D expert to transfer action-relevant temporal geometry into the shared video-action representation. The inference graph therefore remains the same lightweight observation-to-action path used by the base WAM.

Geometry-Aware Embodied Models

Geometry-aware embodied models improve manipulation by injecting 3D structure through depth, point clouds, spatial priors, or representation alignment. In VLA models, 3D-VLA connects 3D perception, reasoning, and action through a generative world model (Zhen et al. 2024). SpatialVLA (Qu et al. 2025) studies spatial representations for action prediction, BridgeVLA (Li et al. 2025a) aligns 3D inputs and heatmap-style outputs in a shared 2D space, 3DS-VLA (Li et al. 2025b) introduces 3D spatial constraints for robust multi-task manipulation, and GeoVLA (Sun et al. 2025) strengthens VLA policies with explicit 3D representations. Recent variants (Fan et al. 2026; Ni et al. 2026; Spiridonov et al. 2025; Zhang et al. 2026; Qian et al. 2026; Ye et al. 2026b) further improve robustness or data efficiency through diverse point clouds, lightweight spatiotemporal dynamics, manipulation data beyond action labels, spatial foundation priors, predictive kinematics with 3D Gaussian geometry, and spatially guided training. Spatial Forcing (Li et al. 2026a) further shows that spatial foundation priors can be transferred into VLA representations through implicit spatial alignment.

In the WAM setting, geometry has begun to move from direct policy inputs into future prediction and video-action co-training. X-WAM (Guo et al. 2026) unifies action execution with 4D world synthesis by predicting multi-view RGB-D futures, adding a lightweight depth branch to a pretrained video diffusion backbone, and using asynchronous denoising for efficient action decoding. WAM4D (Li et al. 2026c) studies fast 4D world action modeling with spatial register tokens and future-depth readouts, then removes the register branch for action inference. MECo-WAM follows this geometry-for-WAM direction but differs in where geometry lives: it treats frozen VGGT (Wang et al. 2025a) 4D structure as a training-time representation constraint, uses decayed read-mask attention to avoid permanent dependence, and transfers temporal geometry into the lightweight video-action path rather than requiring explicit 4D reconstruction during deployment.

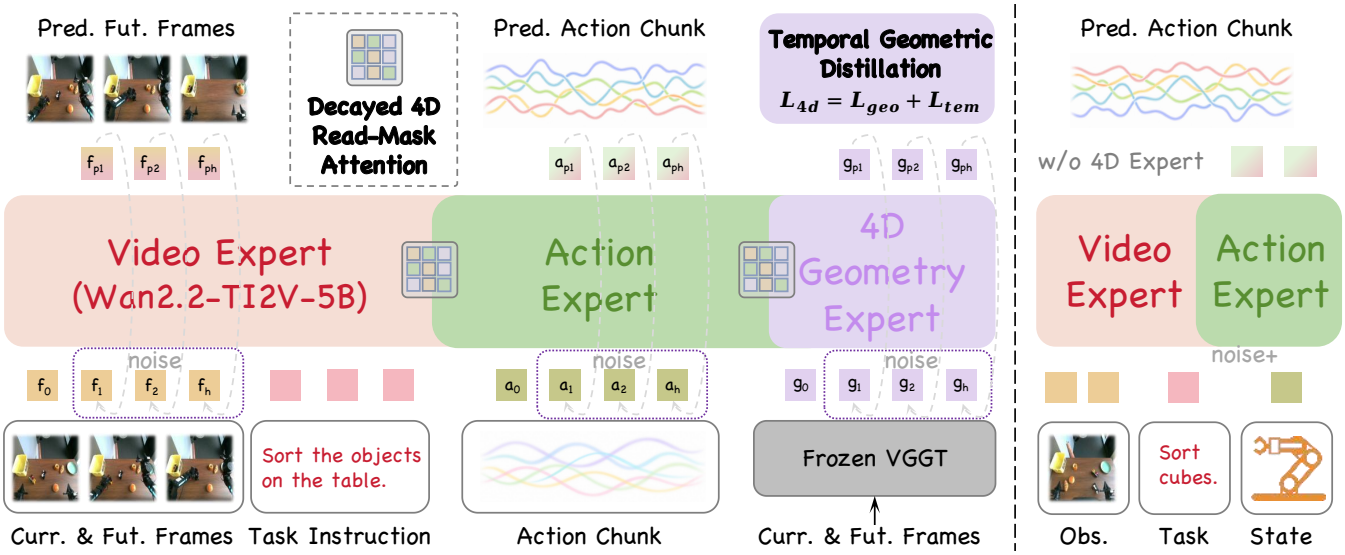


Figure 2: Overview of MECo-WAM. The left side illustrates the training process, while the right side shows the inference process. During training, video frames are encoded by the VAE for video denoising, while current and future frames are encoded by a frozen VGGT encoder to provide targets for 4D geometry denoising. The 4D expert takes g_0, g_1, g_2, and g_h as input slots, predicts g_{p1}, g_{p2}, and g_{ph}, and applies keyframe 4D losses on selected predictions. Decayed 4D read-mask attention transfers early current-frame geometric guidance to the video-action pathway. At inference, only the original video and action experts remain.

Methodology

Problem Formulation

Figure 2 gives an overview of MECo-WAM. Let o_0 denote the current visual observation, a_0 the current robot-state/proprioceptive context, ℓ a language instruction, and $a_{1:H}$ an action chunk of horizon H. The deployed policy models

$$p_\theta(a_{1:H} \mid o_0, a_0, \ell). \tag{1}$$

During training, the model observes visual trajectories and action chunks, and the base video-action pathway learns future visual dynamics together with action denoising. MECo-WAM adds a lightweight 4D expert only for training, using frozen VGGT (Wang et al. 2025a) features from current and future frames as geometric supervision. This auxiliary path constrains the shared representation through relational 4D losses, while the deployed policy remains an observation-to-action model.

Our design principle is that *4D geometry serves as a training-time representation constraint rather than an inference-time input or output.* At deployment, the 4D expert, frozen VGGT encoder, and alignment modules are removed, preserving the original observation-to-action interface.

Multi-Expert Co-Training Architecture

Expert tokens. The video expert uses the clean first-frame VAE token as visual context and denoises noisy future VAE target slots. Let $\bar{Y}_v = [\bar{f}_1, \bar{f}_2, \bar{f}_h]$ denote the clean future VAE targets. For flow-matching time r, we construct noisy future slots

$$\begin{aligned} [f_1, f_2, f_h] &= (1-r)\bar{Y}_v + r\epsilon_v, \\ X_v &= [f_0, f_1, f_2, f_h], \\ [f_{p1}, f_{p2}, f_{ph}] &= E_v(X_v, r; \ell), \end{aligned} \tag{2}$$

where f_{p1}, f_{p2}, and f_{ph} are the corresponding predicted future video outputs supervised against $\bar{Y}_v$ through the video loss. The action expert uses the same noise-slot convention, with a_0 serving as a clean robot-state/proprioceptive anchor rather than a prediction target. Let $\bar{Y}_a = [\bar{a}_1, \bar{a}_2, \bar{a}_h]$ denote clean future action targets:

$$\begin{aligned} [a_1, a_2, a_h] &= (1-r)\bar{Y}_a + r\epsilon_a, \\ X_a &= [a_0, a_1, a_2, a_h], \\ [a_{p1}, a_{p2}, a_{ph}] &= E_a(X_a, r; f_0, \ell). \end{aligned} \tag{3}$$

The 4D expert uses the same denoising convention, but obtains its tokens from a frozen VGGT encoder (Wang et al. 2025a) instead of the VAE. Given current and future RGB frames, frozen VGGT produces clean geometry targets g_0 and $\bar{Y}_g = [\bar{g}_1, \bar{g}_2, \bar{g}_h]$. The current geometry token remains clean, while future geometry slots are noisy:

$$\begin{aligned} [g_1, g_2, g_h] &= (1-r)\bar{Y}_g + r\epsilon_g, \\ X_{4\mathrm{d}} &= [g_0, g_1, g_2, g_h], \\ [g_{p1}, g_{p2}, g_{ph}] &= E_{4\mathrm{d}}(X_{4\mathrm{d}}, r). \end{aligned} \tag{4}$$

The 4D objective is applied only on selected keyframe predictions. Let $\mathcal{K} \subseteq \{1, 2, h\}$ denote the selected keyframe indices:

$$\widehat{G}_{\mathcal{K}} = \{g_{pk} \mid k \in \mathcal{K}\}, \qquad G_{\mathcal{K}} = \{\bar{g}_k \mid k \in \mathcal{K}\}. \tag{5}$$

Clean VGGT targets $G_{\mathcal{K}}$ are used only for training losses and never become input to the deployed policy.

The mixed-attention sequence is

$$X = [X_v, X_a, X_{4\mathrm{d}}]. \tag{6}$$

At each transformer layer, expert $e \in \{v, a, 4\mathrm{d}\}$ has separate query, key, and value projections:

$$Q_e = X_e W_e^Q, \qquad K_e = X_e W_e^K, \qquad V_e = X_e W_e^V. \tag{7}$$

After concatenating the expert-specific tensors, masked mixed attention produces

$$Y = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d}} + M\right) V, \tag{8}$$

where M is an expert-level attention mask. The mask defines the information paths that are allowed during co-training.

Decayed 4D Read-Mask Attention

MECo-WAM introduces decayed 4D read-mask attention to preserve video-geometry co-training benefits while decoupling the 4D path at inference. The mask prevents future-information shortcuts: current anchors f_0, a_0, and g_0 are self-only; future video tokens read the video branch, but not action tokens; and future action tokens read clean visual context plus the action branch, but not noisy future video slots. MECo-WAM then adds temporary read edges from future video/action queries to the current-frame geometry token g_0, which is safe because it is encoded from the current RGB frame only. Future 4D slots stay inside the 4D branch and serve only as auxiliary prediction targets.

Figure 3 visualizes the proposed visibility mask and highlights the decayed 4D attention edges used only during training. The read edges are stochastic and decay over optimization. Let s be the optimization step and let γ_s indicate whether the g_0 read edges are active:

$$\gamma_s \sim \mathrm{Bernoulli}\big(p_{4\mathrm{d}}(s)\big). \tag{9}$$

The activation probability follows a linear decay:

$$p_{4\mathrm{d}}(s) = \begin{cases} p_{\mathrm{start}} + (p_{\mathrm{end}} - p_{\mathrm{start}})\,\frac{s}{S_{\mathrm{decay}}}, & s < S_{\mathrm{decay}}, \\ p_{\mathrm{end}}, & s \geq S_{\mathrm{decay}}. \end{cases} \tag{10}$$

The read edges are present only when $\gamma_s = 1$. This schedule exposes the video-action path to current-frame geometry early in training and gradually removes the auxiliary dependency. At deployment, $p_{\mathrm{end}} = 0$, no 4D tokens are instantiated, and the original video-action graph is recovered.

Action-Aware Temporal Geometric Distillation

The distillation objective uses frozen VGGT as a geometry teacher, but supervises only the predictions of the training-time 4D expert. Its mechanism has three parts. First, relation matching transfers the 3D layout without requiring the student and teacher to share the same absolute feature coordinates. Second, action-aware weights identify which visual tokens are most coupled with the contemporaneous robot motion, so the loss focuses on manipulated objects, targets, and

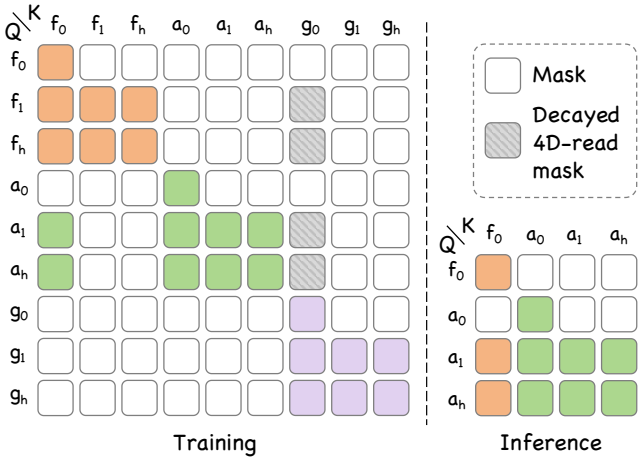


Figure 3: Decayed read-mask attention. Rows and columns denote query and key slots. Hatched cells are temporary reads from future video/action queries to current geometry g_0; white cells are masked. Future 4D slots are auxiliary training targets, and all 4D tokens are removed at inference.

contact regions. Third, temporal relation matching teaches how these action-relevant relations change across keyframes, capturing approach, contact, transport, and release dynamics rather than only static scene geometry.

VGGT-aligned geometry. For selected keyframes $k \in \mathcal{K}$, the student geometry features are

$$Z^k = P_{\mathrm{align}}(g_{pk}), \qquad k \in \mathcal{K}, \tag{11}$$

and the frozen VGGT encoder provides the corresponding clean target geometry $G_T^k = \bar{g}_k$. We represent geometry by pairwise feature relations:

$$\begin{aligned} R_{4\mathrm{d}}^k(i,j) &= \left\| Z_i^k - Z_j^k \right\|_2, \\ R_T^k(i,j) &= \left\| G_{T,i}^k - G_{T,j}^k \right\|_2. \end{aligned} \tag{12}$$

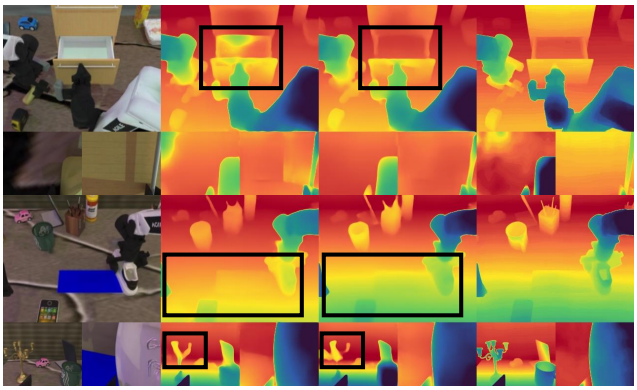
We normalize the valid entries of each relation matrix before alignment, denoting the normalized relations by $\widehat{R}_{4\mathrm{d}}^k$ and $\widehat{R}_T^k$. Relational matching avoids assuming a shared absolute feature coordinate system and focuses supervision on relative object-gripper-target structure.

Action-aware weights. Not all visual regions contribute equally to a given action. Let v_i^k be a selected video token and $\bar{a}_k$ a pooled action representation aligned to the same temporal segment. We compute

$$s_i^k = \frac{(W_v v_i^k)^\top (W_a \bar{a}_k)}{\tau \left\| W_v v_i^k \right\|_2 \left\| W_a \bar{a}_k \right\|_2}, \qquad r_i^k = \mathrm{softmax}_i(s_i^k). \tag{13}$$

The final token weights mix the learned relevance with a uniform prior:

$$w_i^k = N\left((1-\eta) r_i^k + \frac{\eta}{N}\right), \qquad w_{ij}^k = \sqrt{w_i^k w_j^k}, \tag{14}$$



Raw image Fast-WAM **MECo-WAM** Pseudo-GT

Figure 4: Depth probing of shared video-action representations. A matched DPT-style head is trained on tokens from the final four layers of Fast-WAM or MECo-WAM. The improved MECo-WAM depth structure indicates that 4D co-training enriches the deployed video-action representation. Pseudo-GT depth maps are generated by VDA (Chen et al. 2025).

where N is the number of selected tokens and η prevents collapse onto a single location. The within-frame loss is

$$\mathcal{L}_{\text{geo}}^{\text{act}} = \frac{\sum_{k\in\mathcal{K},i,j} w_{ij}^{k} \left| \widehat{R}_{\text{4d}}^{k}(i,j) - \widehat{R}_{T}^{k}(i,j) \right|}{\sum_{k\in\mathcal{K},i,j} w_{ij}^{k}}. \tag{15}$$

This weighting makes the 4D loss action-conditioned. High-relevance tokens contribute more strongly to pairwise geometry, while the uniform mixture prevents the loss from collapsing onto a single region and preserves useful scene context.

Temporal geometry. Manipulation also depends on how geometry changes. For consecutive selected keyframe pairs $(k, k^+) \in \mathcal{A}_{\mathcal{K}}$, we define normalized relation change:

$$\Delta(R_k, R_{k^+}) = \frac{R_{k^+} - R_k}{|R_{k^+}| + |R_k| + \epsilon}. \tag{16}$$

Adjacent-frame pair weights are defined as

$$w_{ij}^{k,k^+} = \sqrt{w_{ij}^{k} w_{ij}^{k^+}}. \tag{17}$$

For compactness, define

$$D_{ij}^{k,k^+} = \left| \Delta(\widehat{R}_{\text{4d}}^{k}, \widehat{R}_{\text{4d}}^{k^+}) - \Delta(\widehat{R}_{T}^{k}, \widehat{R}_{T}^{k^+}) \right|. \tag{18}$$

The temporal loss is

$$\mathcal{L}_{\text{tem}}^{\text{act}} = \frac{\sum_{(k,k^+)\in\mathcal{A}_{\mathcal{K}},i,j} w_{ij}^{k,k^+} D_{ij}^{k,k^+}}{\sum_{(k,k^+)\in\mathcal{A}_{\mathcal{K}},i,j} w_{ij}^{k,k^+}}. \tag{19}$$

This temporal objective complements static relation matching by supervising how action-relevant geometry evolves between keyframes.

Training Objective and Inference

The video and action experts follow the conditional flow-matching objective of the base WAM. For target y, Gaussian noise ϵ, and interpolation time r, we construct

$$y_r = (1-r)y + r\epsilon, \tag{20}$$

and optimize

$$\mathcal{L}_{\text{FM}}(y) = \mathrm{E}_{y,\epsilon,r}\left[\| u_\theta(y_r, r, o_0, a_0, \ell) - (\epsilon - y) \|_2^2 \right]. \tag{21}$$

Instantiating this loss for future video latents and action chunks yields $\mathcal{L}_{\text{video}}$ and $\mathcal{L}_{\text{action}}$. The 4D objective is

$$\mathcal{L}_{\text{4d}} = \alpha_{\text{geo}} \mathcal{L}_{\text{geo}}^{\text{act}} + \alpha_{\text{tem}} \mathcal{L}_{\text{tem}}^{\text{act}}, \tag{22}$$

and the total objective is

$$\mathcal{L}_{\text{total}} = \lambda_{\text{video}} \mathcal{L}_{\text{video}} + \lambda_{\text{action}} \mathcal{L}_{\text{action}} + \lambda_{\text{4d}} \mathcal{L}_{\text{4d}}. \tag{23}$$

During training, MECo-WAM uses the auxiliary 4D expert, frozen VGGT encoder, and decayed read mask, but these components are removed at inference, leaving only the original video and action experts. Thus, MECo-WAM adds no geometric decoder, sensor input, or extra denoising stage to the deployed policy. Figure 4 further suggests that the training-time 4D objective transfers geometric priors into the WAM representation.

Experiments

Experimental Setup

Implementation details. MECo-WAM uses Wan2.2-TI2V-5B (Wan et al. 2025) as the video backbone and follows the Fast-WAM observation-to-action deployment interface (Yuan et al. 2026). The action expert uses 30 DiT blocks, 24 attention heads, 128-dimensional heads, and hidden width $d_a = 1024$ (about 1B parameters). The auxiliary 4D expert uses $d_{\text{4d}} = 512$ (about 0.45B parameters), with supervision from a frozen VGGT-1B encoder. Decayed 4D read probability decreases linearly from 1.0 to 0 over the first half of training, and final-layer VGGT/MECo-WAM tokens are used for relational alignment.

Each training chunk contains 33 robot steps, corresponding to action horizon $H = 32$ and 9 video frames under a $4\times$ action-to-video temporal ratio. We use continuous flow matching with 1000 training timesteps and shift 5.0, AdamW with learning rate 1×10^{-4}, weight decay 0.01, cosine decay, bfloat16 mixed precision, and gradient clipping at 1.0. Training is conducted on 64 NVIDIA H20 96GB GPUs. Inference uses 10 denoising steps with CFG scale 1.0 on a single NVIDIA RTX 5090 32GB GPU, and real-world experiments are conducted with an ARX-R5 robotic arm.

Benchmarks. We evaluate MECo-WAM on LIBERO (Liu et al. 2023), RoboTwin 2.0 (Mu et al. 2025), and real-world tabletop manipulation. For simulation experiments, we follow the Fast-WAM evaluation configuration (Yuan et al. 2026). LIBERO covers four suites: Spatial, Object, Goal, and Long, each containing 10 tasks with 500 expert demonstrations. RoboTwin 2.0 evaluates bimanual manipulation under clean and randomized conditions, where randomization

Method	LIBERO				
	Spat.	Obj.	Goal	Long	Avg.
VLA Models					
π_0	96.8	98.8	95.8	85.2	94.1
π_0+FAST	96.4	96.8	88.6	60.2	85.5
OpenVLA	94.4	88.4	79.2	53.7	76.5
OpenVLA-OFT	97.6	98.4	97.9	94.5	97.1
DD-VLA	97.2	96.6	97.4	92.0	96.3
Uni-VLA	95.4	98.8	93.6	94.0	95.4
X-VLA	98.2	98.6	97.8	97.6	98.1
WAMs					
LingBot-VA (P.T.)	98.5	99.6	97.2	**98.5**	**98.5**
Motus (P.T.)	96.8	99.8	96.6	97.6	97.7
Fast-WAM (w/o P.T.)	98.2	**100.0**	97.0	95.2	97.6
MECo-WAM (w/o P.T.)	**98.8**	**100.0**	**98.2**	95.8	98.2

Table 1: LIBERO success rates (%). Spat./Obj./Avg. denote Spatial/Object/Average; P.T. denotes embodied-policy pretraining; bold/underline denote best/second-best.

Method	P.T.	Clean	Rand.	Average
$\pi_{0.5}$	✓	82.74	76.76	79.75
X-VLA	✓	72.80	72.84	72.82
Motus	✓	88.66	87.02	87.84
LingBot-VA	✓	92.90	91.50	92.20
Fast-WAM	✗	91.88	91.78	91.83
MECo-WAM	✗	**93.26**	**91.98**	**92.62**

Table 2: RoboTwin 2.0 success rates (%) under clean and randomized evaluation. P.T. denotes embodied-policy pretraining.

changes object poses, appearances, clutter, illumination, and tabletop layouts. For real-world evaluation, we use two tabletop tasks, stacking three sponge cubes vertically and sorting three cubes into a size-ordered line, and report success rate, progress rate, correction count, and completion time under an identical camera setup and execution budget.

Main Results

LIBERO. Table 1 compares MECo-WAM with VLA policies, including π_0 (Black et al. 2024), π_0+FAST (Pertsch et al. 2025), OpenVLA (Kim et al. 2025), OpenVLA-OFT (Kim, Finn, and Liang 2025), DD-VLA (Liang et al. 2025), Uni-VLA (Wang et al. 2025b), and X-VLA (Zheng et al. 2025), as well as WAM-style models, including LingBot-VA (Li et al. 2026b), Motus (Bi et al. 2026), and Fast-WAM (Yuan et al. 2026). Without embodied-policy pretraining, MECo-WAM reaches 98.2% average success, improving over Fast-WAM by 0.6 points and Motus by 0.5 points, while remaining close to the pretrained LingBot-VA result (98.5%). The gains are most apparent on geometry-sensitive suites: MECo-WAM obtains 98.8% on Spatial, 100.0% on Object, and 98.2% on Goal, improving Fast-WAM by 0.6 points on Spatial, matching it on Object, and adding 1.2 points on Goal. These results suggest that action-aware 4D co-training mainly strengthens spatial and object-centric reasoning, which is exactly where WAM policies benefit from better geometric priors.

RoboTwin 2.0. Table 2 reports aggregate RoboTwin 2.0 success rates and additionally compares with $\pi_{0.5}$ (Intelligence et al. 2025). MECo-WAM obtains the best overall average, improving the non-pretrained Fast-WAM baseline from 91.83% to 92.62% while keeping the same inference-time video-action graph. The improvement is larger in the clean setting (93.26% vs. 91.88%, +1.38 points) and remains positive under randomization (91.98% vs. 91.78%, +0.20 points), indicating that the learned geometric prior helps both nominal execution and visually perturbed scenes. MECo-WAM also slightly surpasses the strongest pretraining-based WAM average in the table, LingBot-VA (92.62% vs. 92.20%), showing that training-time 4D supervision can make a non-pretrained WAM competitive without adding geometry modules at deployment.

Real-world evaluation. The real-world tabletop study evaluates *Stack Cubes* and *Sort Cubes by Size* on an ARX-R5 robotic arm, with representative rollouts shown in Figure 5. Table 3 compares π_0, Fast-WAM, and MECo-WAM using success rate, progress rate, correction count, and completion time. On *Stack Cubes*, MECo-WAM matches Fast-WAM in SR/PR (60.0%/75.0%) but reduces corrections from 1.67 to 0.83 and shortens completion time from 27.06s to 25.71s, suggesting cleaner vertical alignment even when the final success rate is tied. On *Sort Cubes by Size*, MECo-WAM improves Fast-WAM from 60.0% to 70.0% SR and from 75.0% to 80.0% PR, while reducing corrections from 1.33 to 1.00 and completion time from 38.49s to 31.96s. Averaged over the two tasks, MECo-WAM gains 5.0 SR points and 2.5 PR points over Fast-WAM, with about 39% fewer corrections and 12% shorter completion time, indicating that the learned 4D prior improves real-robot spatial grounding rather than only simulated benchmark scores.

Representation probing. With the video-action backbone frozen, a matched DPT-style head (Ranftl, Bochkovskiy, and Koltun 2021) on tokens from the final four layers shows sharper depth for MECo-WAM (Figure 4) (Li et al. 2026a). The grasp comparison in Figure 6 further shows stronger sensitivity to action-relevant 3D position and pose. Both probes exclude auxiliary 4D tokens, indicating that geometry transfers into deployed video-action features.

Ablation Studies

Table 4 isolates the proposed 4D co-training components under identical training and inference settings. We evaluate each variant from three aspects: video quality, action prediction, and task success. Adding only an isolated 4D expert yields a small task gain over Fast-WAM (91.83% to 91.87%) but does not improve action MSE (0.032 to 0.034), suggesting that geometric supervision is weak if it remains confined to an auxiliary branch. With decayed 4D read access, average success rises to 92.14% and action MSE drops to 0.026, showing that temporary geometry-to-policy transfer is important. The spatial and temporal relation losses

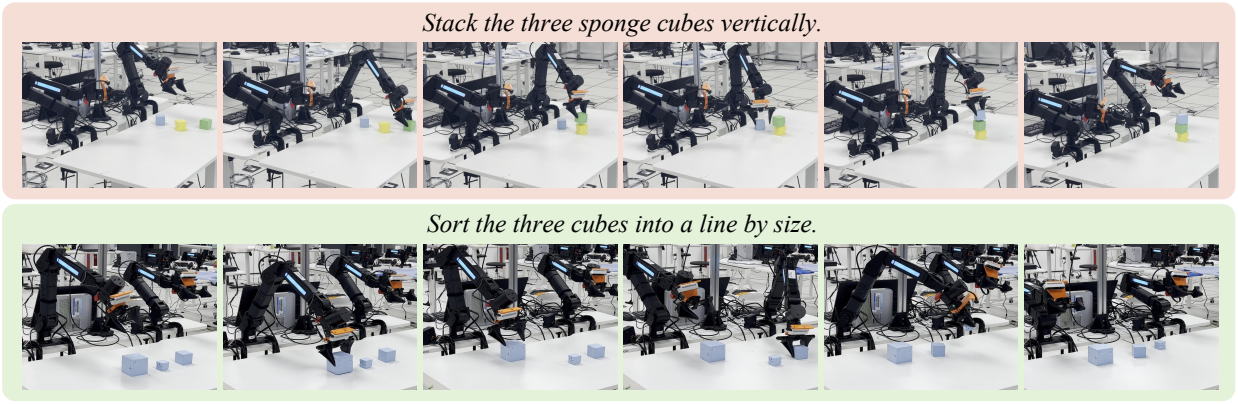
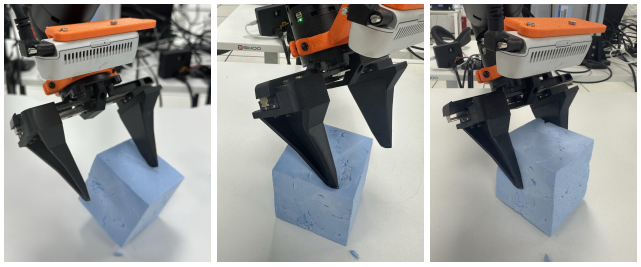


Figure 5: Representative real-robot tabletop experiments on cube stacking and size-based cube sorting.

Method	SR (%) ↑				PR (%) ↑				CR (#) ↓				CT (s) ↓			
	Test 1	Test 2	Test n	Avg	Test 1	Test 2	Test n	Avg	Test 1	Test 2	Test n	Avg	Test 1	Test 2	Test n	Avg
Stack the three sponge cubes vertically.																
π_0	0	100	0	40.0	50	100	50	55.0	–	1	–	**0.75**	–	32.62	–	25.98
Fast-WAM	100	0	100	**60.0**	100	50	100	**75.0**	3	–	2	1.67	28.15	–	30.65	27.06
MECo-WAM	100	0	100	**60.0**	100	50	100	**75.0**	0	–	1	0.83	25.69	–	29.62	**25.71**
Sort the three cubes into a line by size.																
π_0	100	0	100	40.0	100	50	100	60.0	1	–	1	1.50	22.88	–	25.04	**30.82**
Fast-WAM	100	100	100	60.0	100	100	100	75.0	0	4	0	1.33	26.03	57.94	25.23	38.49
MECo-WAM	100	100	100	**70.0**	100	100	100	**80.0**	0	2	3	**1.00**	26.35	28.15	55.47	31.96

Table 3: Real-world tabletop results on two tasks (SR/PR/CR/CT: success rate/progress rate/correction count/completion time).



π_0 Fast-WAM **MECo-WAM**

Figure 6: 3D position and pose sensitivity during real-robot grasping. MECo-WAM better preserves action-relevant cube position, grasp pose, and contact geometry.

Variant	Video			Action	Task
	PSNR ↑	SSIM ↑	LPIPS ↓	MSE ($\times 10$) ↓	Avg SR ↑
Fast-WAM	29.55	0.936	0.038	0.032	91.83
+ 4D expert	29.81	0.935	0.039	0.034	91.87
+ decayed read	30.06	0.939	0.038	0.026	92.14
+ $\mathcal{L}_{\text{geo}}^{\text{act}}$	29.97	0.938	0.037	0.022	92.22
+ $\mathcal{L}_{\text{tem}}^{\text{act}}$	30.42	0.940	0.039	0.019	92.25
Full w/o aware	30.31	0.942	0.038	0.017	92.38
MECo-WAM	30.72	0.942	0.037	0.013	92.62

Table 4: Ablation results on RoboTwin.

further reduce MSE to 0.022 and 0.019, respectively, and raise task success above 92.2%. Full MECo-WAM obtains the best PSNR (30.72), the lowest action MSE (0.013), and the highest average success (92.62%), a 0.79-point gain over Fast-WAM. The uniform-weight variant reaches 92.38%, below the full model, indicating that the gain comes not only from additional 4D supervision but also from emphasizing action-relevant geometry.

Conclusion

We presented MECo-WAM, an inference-efficient WAM that injects 4D geometry only during training. A training-only 4D expert, decayed read-mask attention, and action-aware temporal geometric distillation transfer frozen VGGT priors into the deployed video-action representation, while all auxiliary geometry modules are removed at inference. Results on LIBERO, RoboTwin 2.0, and ARX-R5 real-world tasks, together with ablations, show that action-relevant 4D supervision improves geometric reasoning without increasing deployment cost.

References

Bi, H.; Tan, H.; Xie, S.; Wang, Z.; Huang, S.; Liu, H.; Zhao, R.; Feng, Y.; Xiang, C.; Rong, Y.; et al. 2026. Motus: A unified latent action world model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 35101–35113.

Black, K.; Brown, N.; Driess, D.; Esmail, A.; Equi, M.; Finn, C.; Fusai, N.; Groom, L.; Hausman, K.; Ichter, B.; et al. 2024. π_0: A Vision-Language-Action Flow Model for General Robot Control. *arXiv preprint arXiv:2410.24164*.

Chen, S.; Guo, H.; Zhu, S.; Zhang, F.; Huang, Z.; Feng, J.; and Kang, B. 2025. Video Depth Anything: Consistent Depth Estimation for Super-Long Videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22831–22840.

Fan, X.; Deng, S.; Wu, X.; Lu, Y.; Li, Z.; Yan, M.; Zhang, Y.; Zhang, Z.; Wang, H.; and Zhao, H. 2026. Any3D-VLA: Enhancing VLA Robustness via Diverse Point Clouds. *arXiv preprint arXiv:2602.00807*.

Guo, J.; Li, Q.; Li, P.; Chen, Z.; Sun, N.; Su, Y.; Wang, H.; Zhang, Y.; Li, X.; and Liu, H. 2026. Unified 4D world action modeling from video priors with asynchronous denoising. *arXiv preprint arXiv:2604.26694*.

Hu, Y.; Guo, Y.; Wang, P.; Chen, X.; Wang, Y.-J.; Zhang, J.; Sreenath, K.; Lu, C.; and Chen, J. 2025. Video Prediction Policy: A Generalist Robot Policy with Predictive Visual Representations. In *International Conference on Machine Learning*, volume 267, 24328–24346.

Intelligence, P.; Black, K.; Brown, N.; Darpinian, J.; Dhabalia, K.; Driess, D.; Esmail, A.; Equi, M.; Finn, C.; Fusai, N.; et al. 2025. $\pi_{0.5}$: A Vision-Language-Action Model with Open-World Generalization. *arXiv preprint arXiv:2504.16054*.

Kim, M. J.; Finn, C.; and Liang, P. 2025. Fine-Tuning Vision-Language-Action Models: Optimizing Speed and Success. *arXiv preprint arXiv:2502.19645*.

Kim, M. J.; Gao, Y.; Lin, T.-Y.; Lin, Y.-C.; Ge, Y.; Lam, G.; Liang, P.; Song, S.; Liu, M.-Y.; Finn, C.; et al. 2026. Cosmos Policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*.

Kim, M. J.; Pertsch, K.; Karamcheti, S.; Xiao, T.; Balakrishna, A.; Nair, S.; Rafailov, R.; Foster, E. P.; Sanketi, P. R.; Vuong, Q.; Kollar, T.; Burchfiel, B.; Tedrake, R.; Sadigh, D.; Levine, S.; Liang, P.; and Finn, C. 2025. OpenVLA: An Open-Source Vision-Language-Action Model. In *Conference on Robot Learning*, volume 270, 2679–2713.

Li, F.; Song, W.; Zhao, H.; Wang, J.; Ding, P.; Wang, D.; Zeng, L.; and Li, H. 2026a. Spatial Forcing: Implicit Spatial Representation Alignment for Vision-Language-Action Model. In *International Conference on Learning Representations*.

Li, L.; Zhang, Q.; Luo, Y.; Yang, S.; Wang, R.; Han, F.; Yu, M.; Gao, Z.; Xue, N.; Zhu, X.; et al. 2026b. Causal World Modeling for Robot Control. *arXiv preprint arXiv:2601.21998*.

Li, P.; Chen, Y.; Wu, H.; Ma, X.; Wu, X.; Huang, Y.; Wang, L.; Kong, T.; and Tan, T. 2025a. BridgeVLA: Input-Output Alignment for Efficient 3D Manipulation Learning with Vision-Language Models. In *Conference on Neural Information Processing Systems*.

Li, X.; Heng, L.; Liu, J.; Shen, Y.; Gu, C.; Liu, Z.; Chen, H.; Han, N.; Zhang, R.; Tang, H.; Zhang, S.; and Dong, H. 2025b. 3DS-VLA: A 3D Spatial-Aware Vision Language Action Model for Robust Multi-Task Manipulation. In Lim, J.; Song, S.; and Park, H.-W., eds., *Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, 2344–2359. PMLR.

Li, Y.; Wei, X.; Cao, J.; Wang, H.; Chi, X.; Bai, C.; Sun, Q.; Li, J.; Zhang, X.; Tang, J.; et al. 2026c. WAM4D: Fast 4D World Action Model via Spatial Register Tokens. *arXiv preprint arXiv:2606.14048*.

Liang, Z.; Li, Y.; Yang, T.; Wu, C.; Mao, S.; Pei, L.; Nian, T.; Zhou, S.; Yang, X.; Pang, J.; et al. 2025. Discrete Diffusion VLA: Bringing Discrete Diffusion to Action Decoding in Vision-Language-Action Policies. *arXiv preprint arXiv:2508.20072*.

Liu, B.; Zhu, Y.; Gao, C.; Feng, Y.; Liu, Q.; Zhu, Y.; and Stone, P. 2023. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36: 44776–44791.

Ma, T.; Zheng, J.; Wang, Z.; Jiang, C.; Cui, A.; Liang, J.; and Yang, S. 2026. DiT4DiT: Jointly Modeling Video Dynamics and Actions for Generalizable Robot Control. *arXiv preprint arXiv:2603.10448*.

Mu, Y.; Chen, T.; Chen, Z.; Peng, S.; Lan, Z.; Gao, Z.; Liang, Z.; Yu, Q.; Zou, Y.; Xu, M.; et al. 2025. RoboTwin: Dual-arm robot benchmark with generative digital twins. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 27649–27660.

Ni, C.; Chen, C.; Wang, X.; Zhu, Z.; Zheng, W.; Wang, B.; Chen, T.; Zhao, G.; Li, H.; Dong, Z.; Zhang, Q.; Ye, Y.; Wang, Y.; Huang, G.; and Mei, W. 2026. SwiftVLA: Unlocking Spatiotemporal Dynamics for Lightweight VLA Models at Minimal Overhead. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13474–13485.

Pai, J.; Achenbach, L.; Montesinos, V.; Forrai, B.; Mees, O.; and Nava, E. 2025. mimic-video: Video-action models for generalizable robot control beyond vlas. *arXiv preprint arXiv:2512.15692*.

Pertsch, K.; Stachowicz, K.; Ichter, B.; Driess, D.; Nair, S.; Vuong, Q.; Mees, O.; Finn, C.; and Levine, S. 2025. FAST: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*.

Qian, J.; Han, B.; Shi, C.; Xiao, L.; Yang, L.; Shi, S.; and Jiang, L. 2026. GeoPredict: Leveraging Predictive Kinematics and 3D Gaussian Geometry for Precise VLA Manipulation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13529–13539.

Qu, D.; Song, H.; Chen, Q.; Yao, Y.; Ye, X.; Ding, Y.; Wang, Z.; Gu, J.; Zhao, B.; Wang, D.; et al. 2025. SpatialVLA: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*.

Ranftl, R.; Bochkovskiy, A.; and Koltun, V. 2021. Vision Transformers for Dense Prediction. In *IEEE/CVF International Conference on Computer Vision*, 12179–12188.

Spiridonov, A.; Zaech, J.-N.; Nikolov, N.; Van Gool, L.; and Paudel, D. P. 2025. Generalist Robot Manipulation beyond Action Labeled Data. In *Conference on Robot Learning*.

Su, T.; Zhu, J.; Li, Y.; Ma, C.; Zhang, J.; Huang, Z.; Wang, H.; and Xu, Y. 2025. Towards high-consistency embodied world model with multi-view trajectory videos. *arXiv preprint arXiv:2511.12882*.

Su, T.; Zhu, J.; Wang, T.; He, Y.; Huang, Z.; Zhang, J.; Ma, C.; Wang, H.; Zhang, T.; Yin, M.; et al. 2026. DeMaVLA: A Vision-Language-Action Foundation Model for Generalizable Deformable Manipulation. *arXiv preprint arXiv:2605.31286*.

Sun, L.; Xie, B.; Liu, Y.; Shi, H.; Wang, T.; and Cao, J. 2025. GeoVLA: Empowering 3D Representations in Vision-Language-Action Models. *arXiv preprint arXiv:2508.09071*.

Team, M.; Xiang, C.; Bao, F.; Liu, H.; Tan, H.; Bi, H.; Li, J.; Liu, J.; Pang, J.; Jing, K.; et al. 2026. MotuBrain: An advanced world action model for robot control. *arXiv preprint arXiv:2604.27792*.

Wan, T.; Wang, A.; Ai, B.; Wen, B.; Mao, C.; Xie, C.-W.; Chen, D.; Yu, F.; Zhao, H.; Yang, J.; et al. 2025. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*.

Wang, J.; Chen, M.; Karaev, N.; Vedaldi, A.; Rupprecht, C.; and Novotny, D. 2025a. VGGT: Visual Geometry Grounded Transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5294–5306.

Wang, S.; Shi, J.; Fu, Z.; He, X.; Liu, F.; Yang, C.; Zhou, Y.; Fei, Z.; Gong, J.; Fu, J.; et al. 2026. World Action Models: The Next Frontier in Embodied AI. *arXiv preprint arXiv:2605.12090*.

Wang, Y.; Li, X.; Wang, W.; Zhang, J.; Li, Y.; Chen, Y.; Wang, X.; and Zhang, Z. 2025b. Unified Vision-Language-Action Model. *arXiv preprint arXiv:2506.19850*.

Xu, G.; Zhang, Q.; Zhou, J.; Zhu, X.; Shen, Y.; Yang, X.; and Xu, Y. 2026. Next Forcing: Causal World Modeling with Multi-Chunk Prediction. *arXiv preprint arXiv:2606.11187*.

Ye, A.; Wang, B.; Ni, C.; Huang, G.; Zhao, G.; Li, H.; Li, H.; Li, J.; Lv, J.; Liu, J.; et al. 2026a. GigaWorld-Policy: An Efficient Action-Centered World–Action Model. *arXiv preprint arXiv:2603.17240*.

Ye, J.; Wang, F.; Gao, N.; Yu, J.; Zhu Yangkun; Wang, B.; Zhang, J.; Jin, W.; Fu, Y.; Zheng, F.; Chen, Y.; and Pang, J. 2026b. Spatially Guided Training for Vision-Language-Action Model. In *International Conference on Learning Representations*.

Ye, S.; Ge, Y.; Zheng, K.; Gao, S.; Yu, S.; Kurian, G.; Indupuru, S.; Tan, Y. L.; Zhu, C.; Xiang, J.; et al. 2026c. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*.

Yuan, T.; Dong, Z.; Liu, Y.; and Zhao, H. 2026. Fast-WAM: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*.

Zhang, Z.; Li, H.; Dai, Y.; Zhu, Z.; Zhou, L.; Liu, C.; Wang, D.; Tay, F. E. H.; Chen, S.; Liu, Z.; Liu, Y.; Li, X.; and Zhou, P. 2026. From Spatial to Actions: Grounding Vision-Language-Action Model in Spatial Foundation Priors. In *International Conference on Learning Representations*.

Zhen, H.; Qiu, X.; Chen, P.; Yang, J.; Yan, X.; Du, Y.; Hong, Y.; and Gan, C. 2024. 3D-VLA: A 3D Vision-Language-Action Generative World Model. In *International Conference on Machine Learning*, volume 235, 61229–61245.

Zheng, J.; Li, J.; Wang, Z.; Liu, D.; Kang, X.; Feng, Y.; Zheng, Y.; Zou, J.; Chen, Y.; Zeng, J.; et al. 2025. X-VLA: Soft-prompted Transformer as Scalable Cross-Embodiment Vision-Language-Action Model. *arXiv preprint arXiv:2510.10274*.