JAM: EFFECTIVE COUPLING OF WORLD AND ACTION IN WORLD-ACTION MODEL PRETRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

World Action Models (WAMs), which jointly predict future observations and robot actions, have emerged as a promising paradigm for generalist robot control. Their effectiveness is closely tied to how world and action predictions are coupled during pretraining. Existing WAMs condition one modality on the other, merge both into a shared token sequence, or use multi-stream joint attention between world, action, and text. However, these designs restrict information to one direction, let the high-dimensional world signal dominate the compact action signal, or entangle direct world–action coupling with semantic understanding. We introduce **JAM** (**J**oint World–**A**ction **M**odeling), a WAM pretraining framework where world and action streams exchange information and gradients through MM-DiT-style joint attention, while text conditions both streams. JAM further uses 2D motion tracks to provide an embodiment-agnostic image-space target that makes world–action correspondence explicit and extends supervision to action-free human videos. Pretrained primarily on action-labeled robot data, JAM scales consistently with compute and model capacity. Evaluations across seven simulation benchmark families and real-world experiments demonstrate that the same pretrained JAM model adapts to single-arm, bimanual, mobile, and humanoid embodiments and supports action-only inference without generating future observations. Controlled ablations further show that our two-stream world–action coupling outperforms one-way cross-attention and a shared-residual joint-attention baseline, and that motion-track supervision improves downstream performance. Our code and demos are available at this anonymous repository.

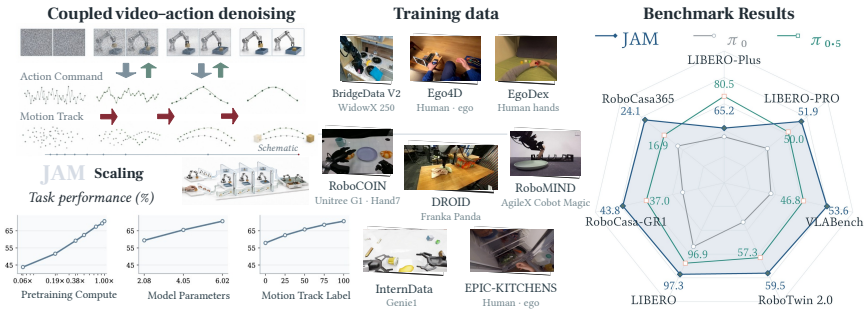

Figure 1: **JAM overview.** World and action streams are jointly denoised with two-way coupling and 2D motion-track supervision (**top left**). JAM pretrained on robot and human data (**middle**) scales with compute and parameters (**bottom left**), and transfers across seven benchmarks (**right**).

1 INTRODUCTION

Video world models learn how visual scenes evolve over time (Assran et al., 2025; Zhen et al., 2025; Chen et al., 2026c; Huang et al., 2026) and are increasingly used in robotics as action-conditioned simulators, policy evaluators, and planners (Jiang et al., 2025; Li et al., 2025c; Quevedo et al., 2026; Kim et al., 2026). World Action Models (WAMs) extend this idea by jointly predicting future observations and robot actions within a single pretrained model (Zhu et al., 2025; Zhou et al., 2026; Kim et al., 2026; Mi et al., 2026; Yuan et al., 2026). Compared to Vision-Language-Action (VLA) policies that map visual and language context directly to actions (Pertsch et al., 2025; Liu et al., 2026; Wang et al., 2026; Black et al., 2025a; Qu et al., 2025; Kim et al., 2025a; Wen et al., 2025), WAMs

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.

additionally learn the effect of actions on the world and have achieved competitive performance (Kim et al., 2026; Yuan et al., 2026; Bi et al., 2025; Li et al., 2026). Joint prediction in WAMs is motivated by the reciprocal relationship between world and action, in which an action constrains how the world evolves and an observed change in the world constrains which actions could have caused it. Pretraining should therefore effectively couple world and action predictions so that action supervision shapes world representations and world prediction shapes action representations. The benefit of joint prediction depends substantially on how this coupling is realized during pretraining.

Existing WAMs couple world and action through one-way conditioning, shared-residual joint attention, or multi-stream joint attention that also includes text in the joint token exchange. Cross-attention and adaptive normalization mainly use one modality as a signal for the other (Huang et al., 2026; Fang et al., 2025), while unified-token models share projections and residual states across modalities (Zhu et al., 2025; Kim et al., 2026; Ye et al., 2026b). MM-DiT and MoT instead assign each modality its own projections and residual computation while attending globally over the combined tokens (Esser et al., 2024; Liang et al., 2025b). Motus (Bi et al., 2025) adopts this MM-DiT-style pattern for world, action, and vision-language understanding together, so text participates directly in the attention path that couples world and action, entangling direct world–action coupling with language understanding. We introduce **JAM** (**J**oint World–**A**ction **M**odeling), which restricts multi-stream joint attention to the world and action streams while supplying text as a condition to both. This isolates reciprocal world–action exchange from the text stream while preserving instruction conditioning. Independently corrupted world and action variables interact at every layer, and both losses remain connected through the opposite stream so that action prediction shapes world representations and world prediction shapes action representations.

While this reciprocal world–action coupling objective balances how world and action streams exchange information, the model must still learn how actions in a robot's native control coordinates map to changes in future observations. This correspondence is typically supervised only implicitly through joint world–action prediction, making it challenging to learn, especially when pretraining across embodiments with incompatible action spaces. To establish explicit world–action correspondences during pretraining, we additionally predict 2D motion tracks, which encode image-space displacement, as an embodiment-agnostic prediction target alongside native actions. Motion tracks further allow us to use egocentric human videos with hand motion during pretraining (Hoque et al., 2026; Luo et al., 2026; Li et al., 2025a; Jang et al., 2025). To summarize our contributions:

- We propose JAM, a WAM pretraining framework that restricts MM-DiT-style joint attention to the world and action streams, separating direct world–action coupling from text understanding. World and action variables are corrupted independently with cross-stream gradient paths preserved, so that action supervision shapes world representations and world prediction shapes action representations.
- We demonstrate the effectiveness of motion tracks as an embodiment-agnostic prediction target in WAM pretraining, tying native robot actions to their visual effects in the world and improving cross-embodiment transfer.
- We present scaling trends for JAM pretrained mostly on action-labeled robot data, with downstream performance improving with compute and model capacity, and evaluate the pretrained model on seven simulation benchmark families and in real-world experiments, where it effectively adapts across embodiments and supports action-only inference without generating future observations.

2 Related Work

2.1 World Action Models (WAMs) and world-action coupling

Large-scale video pretraining enables models to effectively learn how scenes change over time, either by predicting future representations or by generating future frames (Team Wan et al., 2025; Assran et al., 2025; Zhen et al., 2025). Embodied world models then make scene dynamics controllable by conditioning future observations on language, actions, or trajectories (Jiang et al., 2025; Chen et al., 2026c; Feng et al., 2025; Li et al., 2025d; Guo et al., 2026b). WAMs extend this by learning future observations and actions within one pretraining system through video-action diffusion, latent actions, modality-independent corruption, and flexible noise schedules (Zhu et al., 2025; Zhou et al., 2026; Li et al., 2025b; Bi et al., 2025). More recent WAMs use predicted video as policy context, center world learning on actions, or decode actions without test-time future frame generation (Mi et al., 2026; Ye

et al., 2026b;a; Yuan et al., 2026), enabling them to turn generated dynamics into planning context, policy supervision, action evaluation, and synthetic experience (Chen et al., 2026a; Liang et al., 2025a; Jang et al., 2025; Gu et al., 2026). Their architectures differ in how information and gradients move between modalities. Some systems condition a target stream through cross-attention or adaptive normalization (Huang et al., 2026; Fang et al., 2025); others concatenate video and action tokens under shared or causal self-attention (Zhu et al., 2025; Kim et al., 2026; Ye et al., 2026b). MM-DiT introduced modality-specific projections and residual streams with joint attention across the combined token set (Esser et al., 2024), while Mixture-of-Transformers extends this principle by separating attention, normalization, and feed-forward parameters by modality under global self-attention (Liang et al., 2025b). Motus applies related expert-specific projections and shared multi-stream attention to latent-action world modeling (Bi et al., 2025). JAM adopts this established multi-stream joint-attention pattern rather than proposing a new attention operator. Its contribution is a WAM pretraining formulation in which independently corrupted world, native-action, and Motion Track targets share the block at every layer, both prediction losses retain gradient paths through the opposite stream, and partial Motion Track supervision connects executable robot controls with visible robot and human motion across embodiments.

Table 1: **Comparison with representative WAMs.** w, a, and tr denote future observations, robot actions, and motion tracks. ✓ marks a met criterion and $\times$ an unmet one. We include full comparison criteria in Appendix D.

Method	Attention mechanism	Embodied pretraining	Cross-emb.	Joint w, a	Motion loss	Joint w, a, tr
Motion Image Diffusion (Fang et al., 2025)	Cross-attn.	✓	×	×	✓	×
A2World (Huang et al., 2026)	Cross-view attn.	✓	✓	×	×	×
Cosmos Policy (Kim et al., 2026)	Joint self-attn.	×	×	✓	×	×
UWM (Zhu et al., 2025)	Joint self-attn.	✓	×	✓	×	×
DreamZero (Ye et al., 2026b)	Causal self-attn.	✓	×	✓	×	×
LingBot-VA (Li et al., 2026)	Causal self-attn.	✓	×	✓	×	×
Motus (Bi et al., 2025)	Multi-stream self-attn.	✓	✓	✓	✓	×
JAM (ours)	**MM-DiT joint attn.**	✓	✓	✓	✓	✓

2.2 Visual motion representations in embodied learning

An action trajectory is a continuous sequence in a robot's native control coordinates, whereas a Motion Track is the image-plane path of a visible point over time. Visual motion representations have progressed from dense two-frame optical flow (Fischer et al., 2015; Sun et al., 2018; Teed & Deng, 2020; Xu et al., 2022) to long-horizon point trajectories that retain correspondence through deformation and occlusion (Doersch et al., 2022; Harley et al., 2022; Doersch et al., 2023; Wang et al., 2023; Karaev et al., 2024), and to spatial tracks and articulation fields that express motion in three dimensions (Xiao et al., 2024; Eisner et al., 2022). Embodied systems use these representations throughout the learning pipeline. Flow fields and trajectory sketches guide planning at inference time (Xu et al., 2024; Gu et al., 2023); tracked correspondences support few-shot imitation and residual policy adaptation (Vecerik et al., 2023; Bharadhwaj et al., 2024); and pretraining learns point dynamics, motion images, latent actions, or hand motion from video before downstream control (Wen et al., 2024; Fang et al., 2025; Bi et al., 2025; Hoque et al., 2026). JAM builds on this progression by generating Motion Tracks as a shared pretraining target for human and robot video while retaining native robot actions for execution. Our controlled studies measure how track form, density, and supervision coverage shape downstream control.

2.3 Cross-embodiment pretraining

Generalist policies extend action learning across bodies and tasks through robot data, simulation, and heterogeneous control interfaces (NVIDIA et al., 2025; Gemini Robotics Team et al., 2025; AgiBot-World-Contributors et al., 2025; Chen et al., 2025; 2026b). Human and egocentric demonstrations broaden this experience through visible hand and object motion (Luo et al., 2025; 2026; Cai et al., 2025; Li et al., 2025a). Cross-embodiment world-action models further use hands, skeletons, handheld

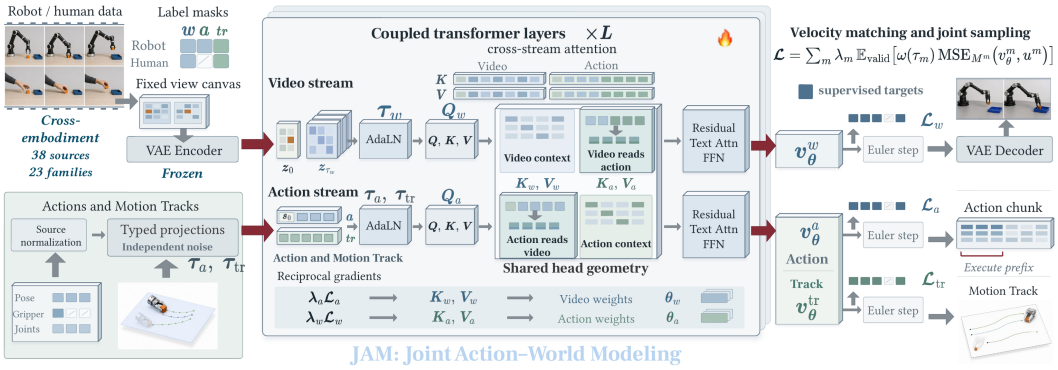

Figure 2: **How action learning enters world pretraining.** Source masks select available video, action, and Motion Track targets. Independently corrupted future-video, action, and Motion Track tokens exchange information through MM-DiT-style joint attention while attached cross-stream gradients couple their objectives and the current observation remains clean. Masked velocity matching jointly predicts future video, actions, and Motion Tracks.

views, active perception, tactile signals, and force-grounded interaction to identify structure that survives changes in embodiment (Hoque et al., 2026; Wu & Gao, 2026; Rayyan et al., 2026; Zeng et al., 2025; Wu et al., 2026; Engelbracht et al., 2026). JAM factors this problem into native motor actions for execution and Motion Tracks for shared visual change. Table 1 compares the resulting pretraining objective and attention mechanism with representative systems.

3 Reciprocal World and Action Pretraining Across Embodiments

3.1 Prediction and Coupling Preliminaries

Let $h = (o_0, \ell, s)$ collect the current observation, language instruction, and proprioceptive state, using a learned null token when state is absent. Encoded future observations z, native motor-action chunks a, and image-space Motion Tracks tr define $p_\theta(z, a \mid h)$ or $p_\theta(z, a, \mathrm{tr} \mid h)$ according to label availability.

For world and action tokens H_w and H_a, cross-attention updates one stream from the other as $\mathrm{CA}(a \leftarrow w) = \mathrm{Attn}(Q_a, K_w, V_w)$. Bidirectional cross-attention adds the reverse direction as another sublayer. Joint attention instead applies $\mathrm{JA}(w, a) = \mathrm{Attn}(Q_{[w;a]}, K_{[w;a]}, V_{[w;a]})$, placing both modalities in one residual sequence. MM-DiT and MoT retain modality-specific projections and residual computation while applying attention over the combined token set (Esser et al., 2024; Liang et al., 2025b). JAM adopts this multi-stream joint-attention operator for world and action tokens; the WAM-specific design lies in the prediction variables, corruption process, gradient routing, and heterogeneous supervision described below.

For each generated modality $m \in \{w, a, \mathrm{tr}\}$, let $x^w = z$, $x^a = a$, and $x^{\mathrm{tr}} = \mathrm{tr}$. Independent standard Gaussian noise ϵ^m and corruption level τ_m define rectified-flow training:

$$x^m_{\tau_m} = (1 - \tau_m)x^m + \tau_m \epsilon^m, \qquad u^m = \epsilon^m - x^m. \tag{1}$$

JAM predicts every available velocity from the corrupted variables and clean conditions:

$$(v^w_\theta, v^a_\theta, v^{\mathrm{tr}}_\theta) = F_\theta(z_{\tau_w}, a_{\tau_a}, \mathrm{tr}_{\tau_{\mathrm{tr}}}; h, \tau_w, \tau_a, \tau_{\mathrm{tr}}). \tag{2}$$

Independent corruption covers distinct uncertainty combinations across modalities (Zhu et al., 2025), while reciprocal coupling turns their prediction objectives into a shared representation.

3.2 Reciprocal Computation Across Streams

JAM instantiates the MM-DiT-style block with two generative streams. The world stream retains a pretrained video transformer's latent embeddings, blocks, and output projection. The action stream contains a state token followed by action and Motion Track tokens, with separate input projections and noise-dependent adaptive normalization for each token type. Both streams receive the language

condition. Their residual widths may differ, while their attention projections share a common head geometry.

At every coupled layer, each stream forms its own queries, keys, and values. World queries attend to world and action-stream keys, and action-stream queries attend to action-stream and world keys. For stream m and the other stream $\bar{m}$,

$$H'_m = H_m + P_m \ \mathrm{Attn}\big(Q_m, [K_m; K_{\bar{m}}], [V_m; V_{\bar{m}}]\big), \tag{3}$$

where P_m maps the attention output to the residual width. Each block then applies language cross-attention and a feed-forward update. Motion Track tokens remain in the action stream and participate in the same reciprocal exchange. Figure 2 summarizes the architecture.

Both cross-stream routes remain attached to the computation graph. Action loss therefore updates world parameters through the world keys and values read by the action stream, while world loss has a reciprocal route into action-stream parameters. This makes reciprocity a property of information access and representation learning during WAM pretraining, rather than a claim of a new attention primitive. The current observation forms a clean prefix in the world sequence. Prefix queries read the clean prefix, while generated world tokens read the prefix and the action stream. The state token provides clean action context and is excluded from the generation loss.

3.3 Motion Track pretraining across embodiments

JAM factors behavior into native actions for execution and Motion Tracks for sharing across embodiments. Source adapters map available end-effector, gripper, and joint coordinates into a typed layout while preserving rotation geometry and a common gripper convention. Motion Tracks encode the visible effects of robot and human motion in image space.

A Motion Track expresses displacement from an observation-time anchor. For point k at future time t, let $q_{t,k}$ be its image-normalized position, $q_{0,k}$ its anchor, and $b_{t,k}$ its observability. The target is

$$\mathrm{tr}_{t,k} = \big[g(q_{t,k} - q_{0,k}),\ 2b_{t,k} - 1\big], \tag{4}$$

where g is a fixed displacement gain. Robot examples combine actuator-aligned points with scene probes supplied by point tracking (Karaev et al., 2024); human examples project hand poses through camera calibration and combine pose confidence with visibility (Hoque et al., 2026). Actuator points anchor at their observed position and scene probes at their current-frame query. Image normalization places both on a fixed-view canvas, while coordinate masks distinguish unavailable tracks from observed points with low visibility.

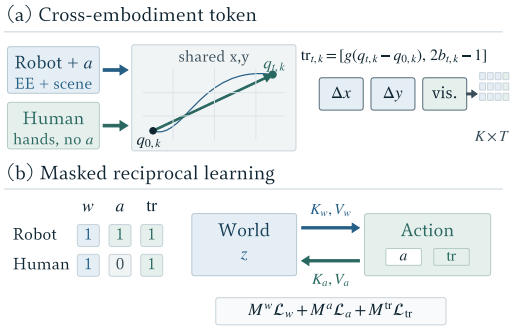

Figure 3: Motion Track pretraining interface. Robot and human motion share image-space track tokens while native controls retain source-specific coordinates. Masks route available targets through reciprocal world and action streams.

Motion Track tokens use dedicated projections and adaptive normalization within the action stream. Video, action, and track variables are corrupted independently. At every reciprocal layer, action and track queries read world keys and values, while world queries read the combined action-stream keys and values; attached gradients couple both representations. The masked objective in Equation 5 trains robot batches with video, native actions, and available tracks, and human batches with video and visible-motion tracks. This factorization learns a shared account of visual change while retaining native coordinates for execution, so downstream adapters can transfer motion-to-visual-change correspondence across cameras, action spaces, and embodiments. Figure 3 summarizes the interface.

3.4 Pretraining under partial supervision

Each example carries coordinate masks M^m for its available labels. We compute MSE_{M^m} over valid coordinates and let $\mathcal{B}_m$ contain examples with supervision for modality m. The objective is

$$\mathcal{L} = \sum_{m \in \{w,a,\mathrm{tr}\}} \lambda_m \frac{1}{|\mathcal{B}_m|} \sum_{i \in \mathcal{B}_m} \omega(\tau_{m,i})\ \mathrm{MSE}_{M_i^m}\left(v_{\theta,i}^m, u_i^m\right), \tag{5}$$

where λ_m controls each modality and ω weights its noise level. Only examples with valid coordinates enter the corresponding average. Robot batches supervise future video and native actions together with available Motion Tracks. Human batches supervise future video and available Motion Tracks. The same reciprocal model therefore learns from every supplied target without translating human motion into robot control coordinates. Section 4 specifies the data mixture, optimization, refinement conditions, and downstream adaptation protocol.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

JAM pretrains on quality-filtered robot demonstrations, simulated trajectories, and egocentric videos. Each window pairs language and the current observation with future frames plus available state, native action, and Motion Track targets. Frozen video and text encoders provide representations, while source adapters map embodiment-specific controls and image-space tracks into typed layouts. Source-specific microbatches, temperature sampling, and masked objectives balance the mixture and optimize observed targets. Simulation contributes 46.97% of the pretraining sampling allocation. The mixture combines real-robot, simulation, and egocentric data; Appendix A summarizes its dataset composition. Downstream adaptation retains reciprocal training and follows each benchmark's demonstrations, evaluation splits, and metrics. Modality-specific integration schedules support joint future sampling and action-only inference. We compare JAM with VLA models NORA (Hung et al., 2025), UniVLA (Bu et al., 2025), π_0 (Black et al., 2025b), π_0-FAST (Pertsch et al., 2025), $\pi_{0.5}$ (Black et al., 2025a), OpenVLA-OFT (Kim et al., 2025a), X-VLA (Zheng et al., 2025), StarVLA (StarVLA Community, 2026), MolmoAct (Lee et al., 2025), ABot-M0 (Yang et al., 2026a), and GR00T-N1.5 (NVIDIA et al., 2025; Bjorck et al., 2025); world-action models Mantis (Yang et al., 2026b), Fast-WAM (Yuan et al., 2026), DreamVLA (Zhang et al., 2025), Motus (Bi et al., 2025), AHA-WAM (Cai et al., 2026), X-WAM (Guo et al., 2026a), GigaWorld-Policy (Ye et al., 2026a), and UWM (Zhu et al., 2025); and the task-specific Diffusion Policy (Chi et al., 2025). Appendix A reports the training and evaluation protocol, while Appendix B describes the backbones and baselines.

VLABench

Model	Success	Progress	Intention
π_0	29.4	44.1	55.0
π_0-FAST	39.8	49.5	58.6
X-VLA	30.7	51.2	**70.2**
$\pi_{0.5}$	46.8	61.4	64.3
Motus	36.5	45.3	59.2
JAM	**53.6**	**63.9**	65.1

Table 2: **VLABench.**

RoboTwin 2.0

Model	Clean	Random	Mean
StarVLA	46.5	3.1	24.8
π_0	45.1	15.8	30.4
X-VLA	68.0	20.9	44.4
$\pi_{0.5}$	69.5	**45.2**	57.3
AHA-WAM	64.3	3.2	33.7
Fast-WAM	70.2	1.2	35.7
ABot-M0	57.4	30.4	43.9
X-WAM	70.0	25.8	47.9
JAM	**85.5**	33.6	**59.5**

Table 3: **RoboTwin 2.0.**

LIBERO-PRO

Model	Goal	Spatial	Long	Object	Mean
π_0	41.8	47.2	34.7	39.4	40.8
$\pi_{0.5}$	**52.5**	48.8	44.8	54.0	50.0
MolmoAct	35.4	41.1	30.9	45.3	38.2
NORA	33.7	42.3	27.7	41.4	36.3
X-VLA	40.9	45.0	38.9	46.2	42.8
JAM	49.0	**52.4**	**47.8**	**58.3**	**51.9**

Table 4: **LIBERO-PRO.**

4.2 MAIN RESULT

Tables 2–7 adapt one foundation checkpoint to benchmark-specific observations, native actions, and demonstrations across progressively different embodiments.

The benchmarks span long-horizon language tasks (VLABench), randomized dual-arm control (RoboTwin 2.0), controlled perturbations of a fixed-base Franka (LIBERO-PRO and LIBERO-Plus), and mobile and humanoid control (RoboCasa365 and RoboCasa-GR1). Standard LIBERO is reported in Appendix C, as it is near saturation (JAM averages 97.3%). Across these benchmarks, JAM consistently ranks among the top-performing methods: it performs competitively with state-of-the-art VLA policies and world-action models, achieving the highest success on VLABench, RoboTwin 2.0 clean, and RoboCasa365 and the best LIBERO-PRO mean, while staying within a few points of the strongest baseline on RoboCasa-GR1 and remaining below the strongest VLA policies on LIBERO-Plus.

LIBERO-Plus	
Model	Mean
NORA	38.6
UniVLA	42.1
π_0	51.6
X-VLA	68.8
StarVLA	71.7
$\pi_{0.5}$	**80.5**
Mantis	38.5
Fast-WAM	47.3
DreamVLA	48.1
JAM	65.2

Table 5: **LIBERO-Plus.**

RoboCasa365				
Model	Atomic	Seen	Unseen	Mean
DP	15.7	0.2	1.2	6.1
π_0	34.6	6.1	1.1	14.7
$\pi_{0.5}$	39.6	7.1	1.2	16.9
GR00T-N1.5	43.0	9.6	4.4	19.9
GigaWorld-Policy	44.4	11.8	2.9	20.6
JAM	**48.6**	**13.5**	**7.0**	**24.1**

Table 6: **RoboCasa365 mobile manipulation.** Atomic skills and seen/unseen compositions.

RoboCasa-GR1	
Model	Success Rate
π_0	13.6
$\pi_{0.5}$	37.0
GR00T-N1.5	**48.0**
StarVLA	47.8
UWM	14.2
JAM	43.8

Table 7: **RoboCasa-GR1 humanoid.** Over all 24 tabletop tasks.

4.3 Reciprocal and Motion Track ablations

Figure 4 tests the two mechanisms that improve action–world coupling. Panel A is an ablation over attention mechanisms with matched data, parameter, optimization, and adaptation settings. Cross adds a separate other-stream-only attention sublayer after within-stream self-attention. Joint represents the unified-sequence family exemplified by UWM and Cosmos Policy (Zhu et al., 2025; Kim et al., 2026). JAM uses the MM-DiT-style multi-stream joint-attention pattern (Esser et al., 2024; Liang et al., 2025b): it preserves stream-specific projections and residual states while both streams jointly read their own and the other stream's keys and values. Panels B and C isolate Motion Track supervision. Coverage varies the proportion of labelled samples, while form varies which points are tracked at fixed labelled exposure. The gains from greater coverage and the peak reached by end-effector plus moderate-grid tracks show that shared geometric supervision benefits from both sufficient exposure and selective spatial support.

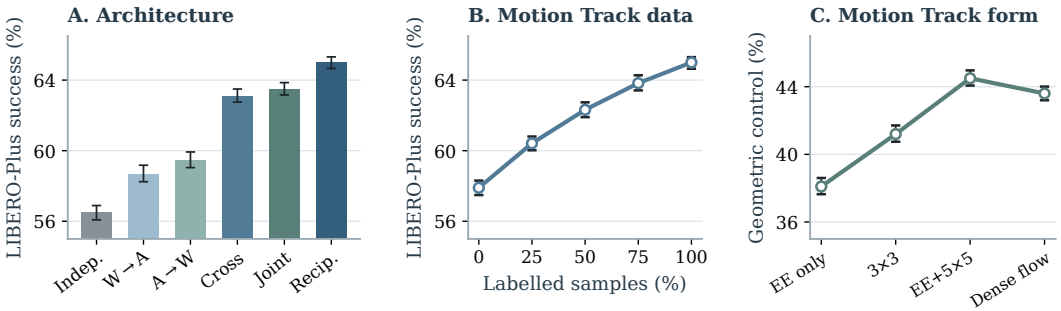

Figure 4: **Controlled ablations of reciprocal pretraining. A** varies only the world–action attention mechanism under matched pretraining and adaptation protocols. **B** varies the percentage of pretraining samples carrying Motion Track labels. **C** varies Motion Track form and density at fixed labelled exposure; geometric control averages LIBERO-Plus camera and layout success with RoboTwin randomized success.

4.4 Scaling behavior of reciprocal pretraining

Following Kaplan et al. (2020), Figure 5 varies pretraining compute (top; 6.02B parameters) and model capacity (bottom; 2.08B, 4.05B, 6.02B) independently under a matched full-data recipe. Task performance P is the mean success rate across the following six evaluation benchmarks: LIBERO, LIBERO-Plus, LIBERO-PRO, VLABench, RoboTwin 2.0 Clean, and RoboCasa-GR1. Along both axes, the residual error $100 - P$ follows a power law, with fitted exponents of 0.24 for compute and 0.30 for parameters. Neither curve saturates at the largest scale we trained, so longer training and additional capacity provide complementary gains.

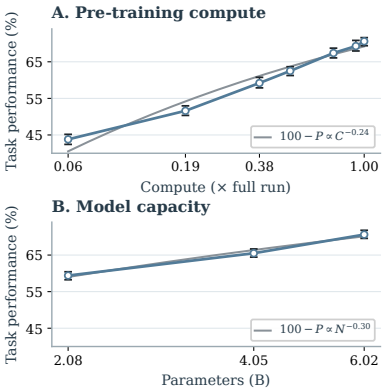

Figure 5: **Scaling pretraining.**

4.5 Action-only inference and held-out transfer

On the four standard LIBERO suites (Liu et al., 2023), Figure 6 (top) drops the generated-video keys and values read by the action stream with probability p during matched adaptation. Joint inference samples both modalities, while action-only inference omits future-video generation. Success averages the official completion outcome over 20 rollouts for each of 40 tasks. With p=0, action-only inference fails; moderate dropout (p=0.5) brings it close to joint inference while retaining joint video–action supervision, whereas p=1 lowers joint success. The matched video-prior baseline uses the same backbone and target-task adaptation data as JAM, but initializes from future-video prediction pretraining without reciprocal action-stream coupling. The bottom panel adapts both with ten demonstrations per task on ALOHA tasks in RoboTwin 2.0, mobile manipulation in RoboCasa365, and the 24 GR1 tabletop tasks in RoboCasa-GR1 (Chen et al., 2025; Nasiriany et al., 2026; NVIDIA, 2025). Bars report mean task-averaged success under each benchmark's protocol; JAM improves over the video prior in all three settings.

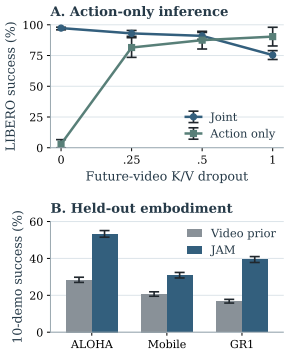

Figure 6: **Efficient deployment and held-out transfer.**

4.6 Inference speed

Table 8 compares end-to-end latency under each method's deployment configuration. JAM jointly denoises video and action for ten steps, predicts 32 actions, and executes the first 16, averaging 1.08 seconds per request. This is 28% lower latency than Motus (Bi et al., 2025) and 79% lower than LingBot-VA (Li et al., 2026) under the listed settings. Because the comparison retains each model's native denoising schedule and action horizon, it measures deployment throughput rather than isolated backbone compute.

Model	Video/action steps	Generated actions	Executed actions	Latency (s/request)
LingBot-VA	25/50	32	16	5.23
Motus	10/10	16	16	1.49
JAM	10/10	32	16	**1.08**

Table 8: **Inference speed** on RTX A5000.

4.7 Real-world experiments

Model	Bread	Ring	Stack	Grill	Noodle	Total
$\pi_{0.5}$	14/20	9/20	2/20	15/20	12/20	52/100
Fast-WAM	0/20	0/20	0/20	3/20	0/20	3/100
LingBot-VA	**18/20**	0/20	0/20	17/20	14/20	49/100
JAM	17/20	**12/20**	**5/20**	**19/20**	**18/20**	**71/100**

Table 9: **Real-world results.** Successes out of 20 trials per task on a Franka Emika arm.

Table 9 evaluates five tabletop tasks on a Franka Emika arm with a wrist camera and a third-person camera: pick-and-place (bread, noodle), precise placement (ring onto a peg), long-horizon stacking (three blocks), and articulated manipulation (grill lid). Figure 7 shows the workspace, in-progress JAM rollouts on each task, and randomized initial states. Each model is fine-tuned on the same 50 teleoperated demonstrations per task, 45 minutes of all tasks, and evaluated over 20 trials with randomized object placements. JAM succeeds in 71 of 100 trials, versus 52 for $\pi_{0.5}$ (Black et al., 2025a), 49 for LingBot-VA (Li et al., 2026), and 3 for Fast-WAM (Yuan et al., 2026). Fast-WAM was originally trained on 60 hours of demonstrations for a single task (Yuan et al., 2026), and it rarely succeeds with our 50 demonstrations per task. JAM outperforms $\pi_{0.5}$ on every task. JAM and LingBot-VA perform comparably on pick-and-place and the grill lid, but only JAM and $\pi_{0.5}$ succeed on ring hanging and block stacking, with JAM highest on both (12/20 and 5/20). These results show that JAM transfers to a physical robot from a small demonstration budget, with its largest gains on tasks that require precise alignment and multi-step execution.

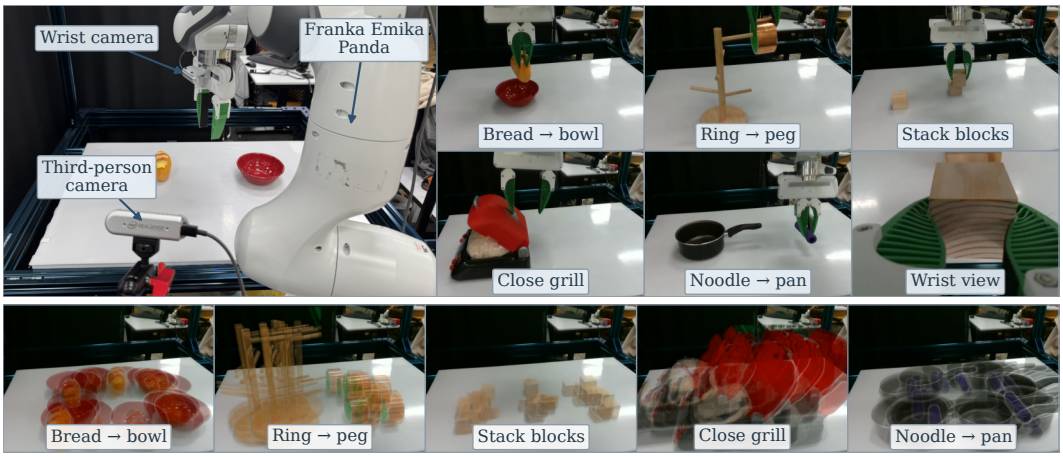

Figure 7: **Real-world setup, rollouts, and randomization. Top:** the Franka Emika Panda workspace with wrist and third-person cameras, in-progress JAM rollouts on each task, and an example wrist-camera view. **Bottom:** overlaid initial frames of all evaluation episodes.

5 Conclusion

JAM makes reciprocity explicit in world-action pretraining. Separate world and action streams exchange information and gradients at every coupled layer, and Motion Tracks provide a shared visual interface across native robot commands and action-free human motion. These architectural choices yield a strong generalist: a single pretrained checkpoint, adapted to each benchmark only through fine-tuning, performs competitively with state-of-the-art VLA policies and world-action models across single-arm, bimanual, mobile, and humanoid embodiments. The same checkpoint also transfers to a physical Franka arm from 50 demonstrations per task, with its largest gains on precise and multi-step tasks. Finally, performance improves consistently with pretraining compute and model capacity and does not yet saturate at our largest scale, suggesting that the architecture may benefit further from additional compute and larger models. Together, these findings support reciprocal learning of actions and their visual effects as a scalable foundation for cross-embodiment control.

Limitations. Foundation pretraining remains computationally demanding. For the real-world experiments, LoRA-based fine-tuning keeps adaptation substantially lighter than foundation pretraining. JAM remains weaker than the strongest VLA policies under severe visual perturbation, as reflected in LIBERO-Plus and randomized scenes in RoboTwin 2.0. Our real-world evaluation covers five tabletop tasks on a single robot, and our scaling study stops at 6B parameters; broader physical evaluation and larger models are left to future work.

AI USAGE STATEMENT

In this work, we used generative AI tools to assist in implementing the methods, including code implementation and experiment execution during the later stages of the project. We did not use generative AI tools to generate synthetic datasets; develop theoretical models or conceptual frameworks; formulate mathematical claims; provide critical ingredients for proving mathematical claims; write proofs; propose or refine hypotheses; design or provide feedback on research methodology or experiments; translate text; clean or reformat datasets; support qualitative or thematic data analysis; or interpret results.

Additionally, we used generative AI tools to improve the wording and readability of the manuscript through discussions with the authors about writing and presentation, correct grammatical errors, and assist in writing and editing code. The experiments were designed by the authors. The authors reviewed and verified the AI-assisted implementations and experimental results, and reviewed and revised all AI-assisted text. The authors take full responsibility for the final content of this work, including its text, code, and experimental results.

REPRODUCIBILITY STATEMENT

The main text describes JAM's architecture and learning objectives. Appendix A details the model interface, supervision masks, training hyperparameters, data mixture, simulation evaluation protocols, ablation and inference settings, and real-world adaptation and evaluation procedures. Appendix B describes the experimental baselines. Our code is available at https://anonymous.4open.science/r/anonymous-JAM-B9BF/ to facilitate reproduction of our results.

REFERENCES

AgiBot-World-Contributors, Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, Shu Jiang, Yuxin Jiang, Cheng Jing, Hongyang Li, Jialu Li, Chiming Liu, Yi Liu, Yuxiang Lu, Jianlan Luo, Ping Luo, Yao Mu, Yuehan Niu, Yixuan Pan, Jiangmiao Pang, Yu Qiao, Guanghui Ren, Cheng Ruan, Jiaqi Shan, Yongjian Shen, Chengshi Shi, Mingkang Shi, Modi Shi, Chonghao Sima, Jianheng Song, Huijie Wang, Wenhao Wang, Dafeng Wei, Chengen Xie, Guo Xu, Junchi Yan, Cunbiao Yang, Lei Yang, Shukai Yang, Maoqing Yao, Jia Zeng, Chi Zhang, Qinglin Zhang, Bin Zhao, Chengyue Zhao, Jiaqi Zhao, and Jianchao Zhu. AgiBot World Colosseo: A Large-Scale Manipulation Platform for Scalable and Intelligent Embodied Systems. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3549–3556. IEEE, 2025. doi: 10.1109/IROS60139.2025.11247088. URL https://doi.org/10.1109/IROS60139.2025.11247088.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-Supervised Video Models Enable Understanding, Prediction and Planning, 2025. URL https://arxiv.org/abs/2506.09985v1.

Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2Act: Predicting Point Tracks from Internet Videos enables Generalizable Robot Manipulation. In *European Conference on Computer Vision*, pp. 306–324. Springer Nature, 2024. doi: 10.1007/978-3-031-73116-7_18. URL https://doi.org/10.1007/978-3-031-73116-7_18.

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/2512.13030v2.

Johan Bjorck, Valts Blukis, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Xiaowei Jiang,

Kaushil Kundalia, Jan Kautz, Zhiqi Li, Kevin Lin, Zongyu Lin, Loic Magne, Yunze Man, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Jing Wang, Qi Wang, Shihao Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Seonghyeon Ye, Zhiding Yu, Yizhou Zhao, Zhe Zhang, Ruijie Zheng, and Yuke Zhu. GR00T N1.5: An Improved Open Foundation Model for Generalist Humanoid Robots. NVIDIA GEAR research release, June 2025. URL https://research.nvidia.com/labs/gear/gr00t-n1_5/.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 2025a. URL https://proceedings.mlr.press/v305/black25a.html.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025b. doi: 10.15607/RSS.2025.XXI.010. URL https://www.roboticsproceedings.org/rss21/p010.html.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Learning to Act Anywhere with Task-centric Latent Actions. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.014. URL https://www.roboticsproceedings.org/rss21/p014.html.

Jisong Cai, Long Ling, Shiwei Chu, Zhongshan Liu, Jiayue Kang, Zhixuan Liang, Wenjie Xu, Yinan Mao, Weinan Zhang, Xiaokang Yang, Ru Ying, Ran Zheng, and Yao Mu. AHA-WAM: Asynchronous Horizon-Adaptive World-Action Modeling with Observation-Guided Context Routing, 2026. URL https://arxiv.org/abs/2606.09811v1.

Xiongyi Cai, Ri-Zhao Qiu, Geng Chen, Lai Wei, Isabella Liu, Tianshu Huang, Xuxin Cheng, and Xiaolong Wang. In-N-On: Scaling Egocentric Manipulation with in-the-wild and on-task Data, 2025. URL https://arxiv.org/abs/2511.15704v1.

Boyuan Chen, Tianyuan Zhang, Haoran Geng, Caiyi Zhang, Peihao Li, Kiwhan Song, William T. Freeman, Jitendra Malik, Pieter Abbeel, Russ Tedrake, Vincent Sitzmann, and Yilun Du. Large Video Planner Enables Generalizable Robot Control, 2026a. URL https://arxiv.org/abs/2512.15840v2.

Delong Chen, Tejaswi Kasarla, Yejin Bang, Mustafa Shukor, Willy Chung, Jade Yu, Allen Bolourchi, Theo Moutakanni, and Pascale Fung. Action100M: A Large-scale Video Action Dataset, 2026b. URL https://arxiv.org/abs/2601.10592v1.

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.

Yuzhi Chen, Ronghan Chen, Dongjie Huo, Yandan Yang, Dekang Qi, Haoyun Liu, Tong Lin, Shuang Zeng, Junjin Xiao, Xinyuan Chang, Feng Xiong, Xing Wei, Zhiheng Ma, and Mu Xu. ABot-PhysWorld: Interactive World Foundation Model for Robotic Manipulation with Physics Alignment, 2026c. URL https://arxiv.org/abs/2603.23376v2.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10–11):1684–1704, 2025. doi: 10.1177/02783649241273668. URL https://doi.org/10.1177/02783649241273668.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling Egocentric Vision: Collection, Pipeline and Challenges for EPIC-KITCHENS-100. *International Journal of Computer Vision*, 130(1):33–55, 2022. doi: 10.1007/s11263-021-01531-2. URL https://doi.org/10.1007/s11263-021-01531-2.

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. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://arxiv.org/abs/2211.03726.

Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement, 2023. URL https://arxiv.org/abs/2306.08637.

Ben Eisner, Harry Zhang, and David Held. FlowBot3D: Learning 3D Articulation Flow to Manipulate Articulated Objects. In *Proceedings of Robotics: Science and Systems*, New York City, NY, USA, June 2022. doi: 10.15607/RSS.2022.XVIII.018. URL https://www.roboticsproceedings.org/rss18/p018.html.

Tim Engelbracht, René Zurbrügg, Matteo Wohlrapp, Martin Büchner, Abhinav Valada, Marc Pollefeys, Hermann Blum, and Zuria Bauer. Hoi! - A Multimodal Dataset for Force-Grounded, Cross-View Articulated Manipulation, 2026. URL https://arxiv.org/abs/2512.04884v3.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis, 2024. URL https://arxiv.org/abs/2403.03206.

Yu Fang, Kanchana Ranasinghe, Le Xue, Honglu Zhou, Juntao Tan, Ran Xu, Shelby Heinecke, Caiming Xiong, Silvio Savarese, Daniel Szafir, Mingyu Ding, Michael S. Ryoo, and Juan Carlos Niebles. Robotic VLA Benefits from Joint Learning with Motion Image Diffusion, 2025. URL https://arxiv.org/abs/2512.18007v1.

Yao Feng, Hengkai Tan, Xinyi Mao, Chendong Xiang, Guodong Liu, Shuhe Huang, Hang Su, and Jun Zhu. Vidar: Embodied Video Diffusion Model for Generalist Manipulation, 2025. URL https://arxiv.org/abs/2507.12898v4.

Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks, 2015. URL https://arxiv.org/abs/1504.06852.

Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, Steven Bohez, Konstantinos Bousmalis, Anthony Brohan, Thomas Buschmann, Arunkumar Byravan, Serkan Cabi, Ken Caluwaerts, Federico Casarini, Oscar Chang, Jose Enrique Chen, Xi Chen, Hao-Tien Lewis Chiang, Krzysztof Choromanski, David D'Ambrosio, Sudeep Dasari, Todor Davchev, Coline Devin, Norman Di Palo, Tianli Ding, Adil Dostmohamed, Danny Driess, Yilun Du, Debidatta Dwibedi, Michael Elabd, Claudio Fantacci, Cody Fong, Erik Frey, Chuyuan Fu, Marissa Giustina, Keerthana Gopalakrishnan, Laura Graesser, Leonard Hasenclever, Nicolas Heess, Brandon Hernaez, Alexander Herzog, R. Alex Hofer, Jan Humplik, Atil Iscen, Mithun George Jacob, Deepali Jain, Ryan Julian, Dmitry Kalashnikov, M. Emre Karagozler, Stefani Karp, Chase Kew, Jerad Kirkland, Sean Kirmani, Yuheng Kuang, Thomas Lampe, Antoine Laurens, Isabel Leal, Alex X. Lee, Tsang-Wei Edward Lee, Jacky Liang, Yixin Lin, Sharath Maddineni, Anirudha Majumdar, Assaf Hurwitz Michaely, Robert Moreno, Michael Neunert, Francesco Nori, Carolina Parada, Emilio Parisotto, Peter Pastor, Acorn Pooley, Kanishka Rao,

Krista Reymann, Dorsa Sadigh, Stefano Saliceti, Pannag Sanketi, Pierre Sermanet, Dhruv Shah, Mohit Sharma, Kathryn Shea, Charles Shu, Vikas Sindhwani, Sumeet Singh, Radu Soricut, Jost Tobias Springenberg, Rachel Sterneck, Razvan Surdulescu, Jie Tan, Jonathan Tompson, Vincent Vanhoucke, Jake Varley, Grace Vesom, Giulia Vezzani, Oriol Vinyals, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Fei Xia, Ted Xiao, Annie Xie, Jinyu Xie, Peng Xu, Sichun Xu, Ying Xu, Zhuo Xu, Yuxiang Yang, Rui Yao, Sergey Yaroshenko, Wenhao Yu, Wentao Yuan, Jingwei Zhang, Tingnan Zhang, Allan Zhou, and Yuxiang Zhou. Gemini Robotics: Bringing AI into the Physical World, 2025. URL https://arxiv.org/abs/2503.20020v1.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Abrham Gebreselasie, Cristina Gonzalez, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jachym Kolar, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbelaez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4D: Around the World in 3,000 Hours of Egocentric Video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18995–19012, June 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Grauman_Ego4D_Around_the_World_in_3000_Hours_of_Egocentric_Video_CVPR_2022_paper.html.

Jiayuan Gu, Sean Kirmani, Paul Wohlhart, Yao Lu, Montserrat Gonzalez Arenas, Kanishka Rao, Wenhao Yu, Chuyuan Fu, Keerthana Gopalakrishnan, Zhuo Xu, Priya Sundaresan, Peng Xu, Hao Su, Karol Hausman, Chelsea Finn, Quan Vuong, and Ted Xiao. Rt-trajectory: Robotic task generalization via hindsight trajectory sketches, 2023. URL https://arxiv.org/abs/2311.01977.

Songen Gu, Yunuo Cai, Tianyu Wang, Simo Wu, and Yanwei Fu. Say, Dream, and Act: Learning Video World Models for Instruction-Driven Robot Manipulation, 2026. URL https://arxiv.org/abs/2602.10717v1.

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, 2026a. URL https://arxiv.org/abs/2604.26694.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-World: A Controllable Generative World Model for Robot Manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 6121–6138, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/0ae94013da7cd459402fd77874e09ee3-Abstract-Conference.html.

Adam W. Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories, 2022. URL https://arxiv.org/abs/2204.04153.

Ryan Hoque, Peide Huang, David J. Yoon, Mouli Sivapurapu, and Jian Zhang. EgoDex: Learning Dexterous Manipulation from Large-Scale Egocentric Video. In *International Conference on Learning Representations*, volume 2026, pp. 4218–4237, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/07fcc6e2b89439d3ee5ab60939aaa6a0-Abstract-Conference.html.

Ze Huang, Jiahui Zhang, Hairuo Liu, Chenxi Zhang, Ran Cheng, and Li Zhang. Learning Transferable Dynamics Priors from Action to World Modeling, 2026. URL https://arxiv.org/abs/2606.29501v1.

Chia-Yu Hung, Qi Sun, Pengfei Hong, Amir Zadeh, Chuan Li, U-Xuan Tan, Navonil Majumder, and Soujanya Poria. NORA: A Small Open-Sourced Generalist Vision Language Action Model for Embodied Tasks, 2025. URL https://arxiv.org/abs/2504.19854v1.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, Loic Magne, Ajay Mandlekar, Avnish Narayan, You Liang Tan, Guanzhi Wang, Jing Wang, Qi Wang, Yinzhen Xu, Xiaohui Zeng, Kaiyuan Zheng, Ruijie Zheng, Ming-Yu Liu, Luke Zettlemoyer, Dieter Fox, Jan Kautz, Scott Reed, Yuke Zhu, and Linxi Fan. DreamGen: Unlocking Generalization in Robot Learning through Video World Models, 2025. URL https://arxiv.org/abs/2505.12705v2.

Yuxin Jiang, Shengcong Chen, Siyuan Huang, Liliang Chen, Pengfei Zhou, Yue Liao, Xindong He, Chiming Liu, Hongsheng Li, Maoqing Yao, and Guanghui Ren. EnerVerse-AC: Envisioning Embodied Environments with Action Condition, 2025. URL https://arxiv.org/abs/2505.09723v1.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL https://arxiv.org/abs/2001.08361.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker: It is Better to Track Together. In *European Conference on Computer Vision*, pp. 18–35. Springer Nature, 2024. doi: 10.1007/978-3-031-73033-7_2. URL https://doi.org/10.1007/978-3-031-73033-7_2.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Donovon Jackson, Charlotte Le, Yunshuang Li, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. DROID: A Large-Scale In-The-Wild Robot Manipulation Dataset. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.120. URL https://www.roboticsproceedings.org/rss20/p120.html.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-Tuning Vision-Language-Action Models: Optimizing Speed and Success. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025a. doi: 10.15607/RSS.2025.XXI.017. URL https://www.roboticsproceedings.org/rss21/p017.html.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. 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. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 2025b. URL https://proceedings.mlr.press/v270/kim25c.html.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos Policy: Fine-Tuning Video Models

for Visuomotor Control and Planning. In *International Conference on Learning Representations*, volume 2026, pp. 71531–71552, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/748becc400a57c0e31cfe6a2e7951467-Abstract-Conference.html.

Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, Winson Han, Wilbert Pumacay, Angelica Wu, Rose Hendrix, Karen Farley, Eli VanderBilt, Ali Farhadi, Dieter Fox, and Ranjay Krishna. MolmoAct: Action Reasoning Models that can Reason in Space, 2025. URL https://arxiv.org/abs/2508.07917v4.

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, 2026. URL https://arxiv.org/abs/2601.21998.

Qixiu Li, Yu Deng, Yaobo Liang, Lin Luo, Lei Zhou, Chengtang Yao, Lingqi Zeng, Zhiyuan Feng, Huizhi Liang, Sicheng Xu, Yizhong Zhang, Xi Chen, Hao Chen, Lily Sun, Dong Chen, Jiaolong Yang, and Baining Guo. Scalable Vision-Language-Action Model Pretraining for Robotic Manipulation with Real-Life Human Activity Videos, 2025a. URL https://arxiv.org/abs/2510.21571v1.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified Video Action Model. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025b. doi: 10.15607/RSS.2025.XXI.074. URL https://www.roboticsproceedings.org/rss21/p074.html.

Yaxuan Li, Yichen Zhu, Junjie Wen, Chaomin Shen, and Yi Xu. WorldEval: World Model as Real-World Robot Policies Evaluator, 2025c. URL https://arxiv.org/abs/2505.19017v1.

Ying Li, Xiaobao Wei, Xiaowei Chi, Yuming Li, Zhongyu Zhao, Hao Wang, Ningning Ma, Ming Lu, and Shanghang Zhang. ManipDreamer: Boosting Robotic Manipulation World Model with Action Tree and Visual Guidance, 2025d. URL https://arxiv.org/abs/2504.16464v1.

Junbang Liang, Pavel Tokmakov, Ruoshi Liu, Sruthi Sudhakar, Paarth Shah, Rares Ambrus, and Carl Vondrick. Video Generators are Robot Policies, 2025a. URL https://arxiv.org/abs/2508.00795v1.

Weixin Liang, Lili Yu, Liang Luo, Srinivasan Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen tau Yih, Luke Zettlemoyer, and Xi Victoria Lin. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models, 2025b. URL https://arxiv.org/abs/2411.04996.

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, pp. 44776–44791. Curran Associates, Inc., 2023. doi: 10.52202/075280-1939. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8c3c666820ea055a77726d66fc7d447f-Abstract-Datasets_and_Benchmarks.html.

Jiaming Liu, Hao Chen, Zhuoyang Liu, Pengju An, Renrui Zhang, Chenyang Gu, Xiaoqi Li, Ziyu Guo, Sixiang Chen, Mengzhen Liu, Chengkai Hou, Mengdi Zhao, Kaichen Zhou, Pheng-Ann Heng, and Shanghang Zhang. HybridVLA: Collaborative Diffusion and Autoregression in a Unified Vision-Language-Action Model. In *International Conference on Learning Representations*, volume 2026, pp. 80668–80694, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/82ad13ec01f9fe44c01cb91814fd7b8c-Abstract-Conference.html.

Hao Luo, Yicheng Feng, Wanpeng Zhang, Sipeng Zheng, Ye Wang, Haoqi Yuan, Jiazheng Liu, Chaoyi Xu, Qin Jin, and Zongqing Lu. Being-H0: Vision-Language-Action Pretraining from Large-Scale Human Videos, 2025. URL https://arxiv.org/abs/2507.15597v1.

Hao Luo, Ye Wang, Wanpeng Zhang, Sipeng Zheng, Ziheng Xi, Chaoyi Xu, Haiweng Xu, Haoqi Yuan, Chi Zhang, Yiqing Wang, Yicheng Feng, and Zongqing Lu. Being-H0.5: Scaling Human-Centric Robot Learning for Cross-Embodiment Generalization, 2026. URL https://arxiv.org/abs/2601.12993v1.

Weishi Mi, Yong Bao, Xiaowei Chi, Xiaozhu Ju, Zhiyuan Qin, Kuangzhi Ge, Kai Tang, Peidong Jia, Shanghang Zhang, and Jian Tang. TC-IDM: Grounding Video Generation for Executable Zero-shot Robot Motion, 2026. URL https://arxiv.org/abs/2601.18323v1.

Soroush Nasiriany, Sepehr Nasiriany, Abhiram Maddukuri, and Yuke Zhu. RoboCasa365: A Large-Scale Simulation Framework for Training and Benchmarking Generalist Robots. In *International Conference on Learning Representations*, volume 2026, pp. 98643–98667, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/a05003fdb1e9562ab0c0a9719ea4de10-Abstract-Conference.html.

NVIDIA. GR00T RoboCasa Tabletop Task Benchmarks. Isaac-GR00T repository, n1.5-release branch, examples/RoboCasa, 2025. URL https://github.com/NVIDIA/Isaac-GR00T/blob/n1.5-release/examples/RoboCasa/README.md. Accessed 2026-09-25.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An Open Foundation Model for Generalist Humanoid Robots, 2025. URL https://arxiv.org/abs/2503.14734v2.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient Action Tokenization for Vision-Language-Action Models. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.012. URL https://www.roboticsproceedings.org/rss21/p012.html.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, and Xuelong Li. SpatialVLA: Exploring Spatial Representations for Visual-Language-Action Model, 2025. URL https://arxiv.org/abs/2501.15830v5.

Julian Quevedo, Ansh Kumar Sharma, Yixiang Sun, Varad Suryavanshi, Percy Liang, and Sherry Yang. WorldGym: World Model as An Environment for Policy Evaluation. In *International Conference on Learning Representations*, volume 2026, pp. 78932–78957, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/7f5e909ac0324db03506b380c695ffaf-Abstract-Conference.html.

Omar Rayyan, John Abanes, Mahmoud Hafez, Anthony Tzes, and Fares J. Abu-Dakka. MV-UMI: A Scalable Multi-View Interface for Cross-Embodiment Learning. *IEEE Access*, 14:104545–104555, 2026. doi: 10.1109/ACCESS.2026.3709982. URL https://doi.org/10.1109/ACCESS.2026.3709982.

StarVLA Community. StarVLA: A Lego-like Codebase for Vision-Language-Action Model Developing, 2026. URL https://arxiv.org/abs/2604.05014v1.

Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume, 2018. URL https://arxiv.org/abs/1709.02371.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and Advanced Large-Scale Video Generative Models. *arXiv preprint arXiv:2503.20314*, 2025. URL https://arxiv.org/abs/2503.20314v2.

Zachary Teed and Jia Deng. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow, 2020. URL https://arxiv.org/abs/2003.12039.

Mel Vecerik, Carl Doersch, Yi Yang, Todor Davchev, Yusuf Aytar, Guangyao Zhou, Raia Hadsell, Lourdes Agapito, and Jon Scholz. RoboTAP: Tracking Arbitrary Points for Few-Shot Visual Imitation, 2023. URL https://arxiv.org/abs/2308.15975.

Homer Rich Walke, Kevin Black, Tony Z. Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, Abraham Lee, Kuan Fang, Chelsea Finn, and Sergey Levine. BridgeData V2: A Dataset for Robot Learning at Scale. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 1723–1736. PMLR, 2023. URL https://proceedings.mlr.press/v229/walke23a.html.

Qianqian Wang, Yen-Yu Chang, Ruojin Cai, Zhengqi Li, Bharath Hariharan, Aleksander Holynski, and Noah Snavely. Tracking Everything Everywhere All at Once. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. URL https://arxiv.org/abs/2306.05422.

Yuqi Wang, Xinghang Li, Wenxuan Wang, Junbo Zhang, Yingyan Li, Yuntao Chen, Xinlong Wang, and Zhaoxiang Zhang. Unified Vision-Language-Action Model. In *International Conference on Learning Representations*, volume 2026, pp. 80929–80944, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/82f625a28d822d2748b9f5c4f9a89bb9-Abstract-Conference.html.

Chuan Wen, Xingyu Lin, John Ian Reyes So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point Trajectory Modeling for Policy Learning. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.092. URL https://www.roboticsproceedings.org/rss20/p092.html.

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. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 3094–3114. PMLR, 27–30 Sep 2025. URL https://proceedings.mlr.press/v305/wen25b.html.

Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, Shichao Fan, Xinhua Wang, Fei Liao, Zhen Zhao, Guangyu Li, Zhao Jin, Lecheng Wang, Jilei Mao, Ning Liu, Pei Ren, Qiang Zhang, Yaoxu Lyu, Mengzhen Liu, Jingyang He, Yulin Luo, Zeyu Gao, Chenxuan Li, Chenyang Gu, Yankai Fu, Di Wu, Xingyu Wang, Sixiang Chen, Zhenyu Wang, Pengju An, Siyuan Qian, Shanghang Zhang, and Jian Tang. RoboMIND: Benchmark on Multi-embodiment Intelligence Normative Data for Robot Manipulation. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.152. URL https://www.roboticsproceedings.org/rss21/p152.html.

Longyan Wu, Checheng Yu, Jieji Ren, Li Chen, Yufei Jiang, Ran Huang, Guoying Gu, and Hongyang Li. FreeTacMan: Robot-free Visuo-Tactile Data Collection System for Contact-rich Manipulation, 2026. URL https://arxiv.org/abs/2506.01941v4.

Zhuoyuan Wu and Jun Gao. OSCAR: Omni-Embodiment Action-Conditioned World Model for Robotics, 2026. URL https://arxiv.org/abs/2606.04463v2.

Yuxi Xiao, Qianqian Wang, Shangzhan Zhang, Nan Xue, Sida Peng, Yujun Shen, and Xiaowei Zhou. SpatialTracker: Tracking Any 2D Pixels in 3D Space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. URL https://arxiv.org/abs/2404.04319.

Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. GMFlow: Learning Optical Flow via Global Matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. URL https://arxiv.org/abs/2111.13680.

Mengda Xu, Zhenjia Xu, Yinghao Xu, Cheng Chi, Gordon Wetzstein, Manuela Veloso, and Shuran Song. Flow as the Cross-Domain Manipulation Interface. In *Conference on Robot Learning (CoRL)*, 2024. URL https://arxiv.org/abs/2407.15208.

Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Dekang Qi, Junjin Xiao, Haoyun Liu, Ronghan Chen, Yuzhi Chen, Dongjie Huo, Feng Xiong, Xing Wei, Zhiheng Ma, and Mu Xu. Abot-m0: Vla foundation model for robotic manipulation with action manifold learning, 2026a. URL https://arxiv.org/abs/2602.11236.

Yi Yang, Xueqi Li, Yiyang Chen, Jin Song, Yihan Wang, Zipeng Xiao, Jiadi Su, You Qiaoben, Pengfei Liu, and Zhijie Deng. Mantis: A Versatile Vision-Language-Action Model with Disentangled Visual Foresight, 2026b. URL https://arxiv.org/abs/2511.16175v2.

Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, Jie Li, Jindi Lv, Jingyu Liu, Min Cao, Peng Li, Qiuping Deng, Wenjun Mei, Xiaofeng Wang, Xinze Chen, Xinyu Zhou, Yang Wang, Yifan Chang, Yifan Li, Yukun Zhou, Yun Ye, Zhichao Liu, and Zheng Zhu. GigaWorld-Policy: An Efficient Action-Centered World–Action Model, 2026a. URL https://arxiv.org/abs/2603.17240v2.

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, 2026b. URL https://arxiv.org/abs/2602.15922v1.

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/2603.16666v2.

Qiyuan Zeng, Chengmeng Li, Jude St. John, Zhongyi Zhou, Junjie Wen, Guorui Feng, Yichen Zhu, and Yi Xu. ActiveUMI: Robotic Manipulation with Active Perception from Robot-Free Human Demonstrations, 2025. URL https://arxiv.org/abs/2510.01607v1.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, XinQiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, Li Yi, Wenjun Zeng, and Xin Jin. DreamVLA: A Vision-Language-Action Model Dreamed with Comprehensive World Knowledge. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-0818. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/22d4f952efa13970f0b1ffb22170d416-Abstract-Conference.html.

Haoyu Zhen, Qiao Sun, Hongxin Zhang, Junyan Li, Siyuan Zhou, Yilun Du, and Chuang Gan. TesserAct: Learning 4D Embodied World Models, 2025. URL https://arxiv.org/abs/2504.20995v1.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Ya-Qin Zhang, Jiangmiao Pang, Jingjing Liu, Tai Wang, and Xianyuan Zhan. X-VLA: Soft-Prompted Transformer as Scalable Cross-Embodiment Vision-Language-Action Model, 2025. URL https://arxiv.org/abs/2510.10274v1.

Pengfei Zhou, Shengcong Chen, Di Chen, Jiaxu Wang, Rongjun Jin, Bingwen Zhu, Yike Pan, Songen Gu, Kuanning Wang, Shufeng Nan, Xingyu Qiu, Chenhao Qiu, Pu Yang, Yunuo Cai, Jianxiong Gao, Yifan Li, Yanwei Fu, Xiangyu Yue, Zhi Chen, and Jianlan Luo. τ_0-WM: A Unified Video-Action World Model for Robotic Manipulation, 2026. URL https://arxiv.org/abs/2606.01027v2.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified World Models: Coupling Video and Action Diffusion for Pretraining on Large Robotic Datasets. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.015. URL https://www.roboticsproceedings.org/rss21/p015.html.

APPENDIX

A IMPLEMENTATION AND TRAINING DETAILS

A.1 MODEL INTERFACE AND PREDICTION TARGETS

Conditioning and backbone. JAM conditions on the current visual observation, a language instruction, and available proprioceptive state. Frozen video and text encoders provide the representations used by the generative model. The world stream inherits the latent embeddings, transformer blocks, and output projection of a pretrained video transformer. The action stream has its own projections and residual computation and exchanges information with the world stream through MM-DiT-style joint attention. Language conditions both streams through cross-attention rather than becoming a third stream in the world–action exchange. The current observation remains a clean visual prefix. State is also a clean condition, represented by a learned null token when unavailable, and is excluded from the generation loss.

Prediction horizons and coordinates. The model predicts 32 action steps in an 80-dimensional typed layout. Motion Tracks span 16 future steps, with each point represented by two displacement coordinates and one observability coordinate. The action stream contains 49 tokens: one state token, 32 action tokens, and 16 Motion Track tokens. Shared slots provide storage for source-specific coordinates, and unused coordinates carry a false validity mask. Table 10 summarizes the roles of the inputs and targets.

Component	Representation	Role in JAM
Current observation	Encoded visual prefix	Clean context for future prediction.
Language instruction	Frozen text features	Conditions both generative streams.
Proprioceptive state	One state token	Clean action context; learned null token when absent.
Future observations	Video latents	Generated world target.
Native actions	32 steps; typed layout	Generated commands retaining source-specific control coordinates.
Motion Tracks	16 steps; displacement and observability	Generated image-space effects of motion.

Table 10: **Conditioning and prediction interface.** State and current observations supply context; future video, native actions, and available Motion Tracks supply generation targets.

Native controls and shared visual effects. Source adapters place available end-effector, gripper, and joint coordinates in the typed action layout while preserving rotation geometry and a common gripper convention. Sharing a layout does not make different embodiments' controls physically interchangeable. Native actions specify how a robot acts, whereas Motion Tracks describe the resulting visible change in image space. Tracks measure image-normalized displacement from an observation-time anchor and include observability, as defined in the main text. Dedicated track projections and noise-dependent adaptive normalization allow these targets to participate in the action stream's reciprocal exchange with the world stream.

Partial supervision and masking. For each generated modality $m \in \{w, a, \mathrm{tr}\}$, JAM independently samples Gaussian noise and a corruption level, using

$$x^m_{\tau_m} = (1 - \tau_m)x^m + \tau_m \epsilon^m, \qquad u^m = \epsilon^m - x^m. \tag{6}$$

Thus, $\tau_m = 0$ denotes the clean target and $\tau_m = 1$ its fully noised endpoint. The velocity sign follows this convention throughout training. Coordinate masks select the valid entries of each target; each modality's loss is then averaged over examples that actually provide that modality, with the noise-level weighting specified by the main objective. A modality with no valid supervision contributes no loss for that batch.

The distinction between validity and observability is essential. An unavailable track is not a zero-displacement label. A defined track with low observability can still convey visibility information, with coordinate masks determining which entries receive supervision. Robot examples supply native actions and available visual targets; action-free human examples contribute video and available Motion Tracks without being converted into pseudo-labeled robot commands. Both cross-stream gradient routes remain attached, so available losses can shape representations in the opposite stream even when some target modalities are absent.

A.2 FOUNDATION OPTIMIZATION AND DATA MIXTURE

Foundation optimization. Table 11 gives the foundation optimization settings. The world, native-action, and Motion Track objectives receive equal scalar weights. Because their valid coordinates and supervised examples differ, equal scalar weights do not imply equal amounts of effective supervision. Saved training states retain the optimizer, scheduler, scaler, and data sampler to support continuation.

Setting	Value
Training hardware	16× NVIDIA H200 GPUs
Global batch size	512
Computation precision	bfloat16
Activation memory	Gradient checkpointing
Distributed training	Hybrid fully sharded data parallelism
World-stream learning rate	5×10^{-5}
Action-stream learning rate	1×10^{-4}
Optimizer	Adam, $\beta_1 = 0.9$, $\beta_2 = 0.95$
Weight decay	0.01
Gradient clipping	1.0
Objective scalar weights	$\lambda_w = \lambda_a = \lambda_{\text{tr}}$

Table 11: **Foundation training settings.** Batch size is global across the listed training GPUs. Objective weights are applied after the modality-specific masked averages.

Data mixture. Pretraining combines real-robot demonstrations, simulated trajectories, and egocentric video. Dataset sources include DROID (Khazatsky et al., 2024), BridgeData V2 (Walke et al., 2023), RoboMIND (Wu et al., 2025), RoboCOIN, InternData, RoboCasa365 (Nasiriany et al., 2026), EgoDex (Hoque et al., 2026), Ego4D (Grauman et al., 2022), and EPIC-KITCHENS-100 (Damen et al., 2022). Table 12 groups them by data domain. Sampling probabilities are assigned at the source-family level; the reported domain percentages are the aggregate sampling allocations. They describe training exposure, not the proportions of raw videos, recording hours, episodes, or examples with action or Motion Track labels.

Domain	Datasets	Sampling (%)
Real robot	DROID; BridgeData V2; RoboMIND; InternData; RoboCOIN	44.86
Simulation	InternData; RoboCasa365 human demonstrations and MimicGen	46.97
Egocentric video	EgoDex (including Extra and Short); Ego4D; EPIC-KITCHENS-100	8.17
Total		**100.00**

Table 12: **Pretraining mixture.** Dataset variants are grouped under their dataset names. Sampling is allocated at the source-family level; the domain totals do not specify individual dataset weights or label coverage.

A.3 SIMULATION BENCHMARKS AND EVALUATION

One foundation checkpoint is adapted to the observations, native action interface, and demonstrations of each downstream benchmark. Adaptation retains reciprocal training. The evaluation setting is defined by each benchmark's task suite and rollout protocol, rather than by a common random split of individual trajectory frames. Table 13 summarizes the seven benchmark families and the metrics used in this paper.

VLABench. The benchmark separates its demonstration data for policy adaptation from configured evaluation episodes. Its evaluation tracks distinguish in-distribution task execution, object-category generalization, common-sense reasoning, and semantic instruction understanding, among other settings. We retain the benchmark's three complementary metrics: success measures complete task achievement, progress gives credit for partial completion, and intention assesses whether behavior is directed toward the instructed goal. Progress and intention should therefore not be interpreted as additional task-success rates.[1]

[1] Official VLABench data and evaluation interface: https://github.com/OpenMOSS/VLABench.

Benchmark	Evaluation setting	Reported metrics
VLABench	Language-conditioned manipulation with benchmark-defined evaluation episodes.	Success, progress, intention
RoboTwin 2.0	Bimanual tasks under clean and randomized conditions.	Clean, Random, Mean
LIBERO	Original Spatial, Object, Goal, and Long suites.	Suite success and average
LIBERO-PRO	Perturbed evaluations organized by the four LIBERO suites.	Suite success and mean
LIBERO-Plus	Robustness evaluation across seven perturbation dimensions.	Mean success
RoboCasa365	Atomic skills and seen/unseen composite tasks in kitchen environments.	Atomic, Seen, Unseen, Mean
RoboCasa-GR1	Humanoid manipulation across 24 tabletop tasks.	Mean task success

Table 13: **Simulation evaluation overview.** Each benchmark supplies its own task definitions and success checks. Standard LIBERO results are reported in Appendix C; the other six families are reported in the main text.

Standard LIBERO. The four reported suites contain ten tasks each and provide demonstrations for policy learning together with task definitions and prescribed initial states for rollout evaluation. Spatial, Object, and Goal vary different aspects of manipulation, while Long contains extended-horizon tasks. Evaluation measures task completion in new rollouts of these tasks rather than prediction accuracy on demonstration frames. The four-suite results are retained in Table 16.[2]

LIBERO-PRO and LIBERO-Plus. These evaluations extend the LIBERO task interface with controlled test variations. LIBERO-PRO defines perturbations of objects, positions, instruction semantics, task goals, and environments, and groups the results by Goal, Spatial, Long, and Object in our main table.[3] LIBERO-Plus varies object layouts, camera viewpoints, robot initial states, language instructions, lighting, background textures, and sensor noise. Its release distinguishes training trajectories from its perturbed evaluation scenarios; the official evaluation runs one rollout per instantiated test variant.[4] The original LIBERO score and the two robustness scores remain separate: high performance on the original tasks does not establish robustness to these changes.

RoboTwin 2.0. The benchmark supplies task demonstrations and separate rollout evaluations for bimanual manipulation. Clean evaluation measures execution under the nominal setting, while randomized evaluation introduces changes such as clutter, lighting, backgrounds, table height, and language. We report both conditions and their mean. The scaling study uses the Clean evaluation setting rather than transfer from clean training to randomized evaluation conditions. [5]

RoboCasa365. The benchmark organizes target-task demonstrations and evaluation into Atomic-Seen, Composite-Seen, and Composite-Unseen groups. These correspond to the Atomic, Seen, and Unseen columns in the main text. The official foundation-model protocol permits target-task adaptation for all three groups and evaluates in target scenes. Consequently, "unseen" identifies a benchmark task group; it does not mean that no downstream demonstrations are available for that group.[6] Results retain the benchmark's distinction between atomic execution and longer compositions (Nasiriany et al., 2026).

RoboCasa-GR1. The GR1 evaluation uses the released humanoid tabletop demonstration data for adaptation and the corresponding simulator tasks for rollout testing. Success is averaged over all 24

[2]Official LIBERO demonstrations, task suites, and initialization interface: https://github.com/Lifelong-Robot-Learning/LIBERO.

[3]Official LIBERO-PRO task definitions and evaluation configurations: https://github.com/Zxy-MLlab/LIBERO-PRO.

[4]Official LIBERO-Plus datasets and evaluation protocol: https://github.com/sylvestf/LIBERO-plus.

[5]Official RoboTwin 2.0 benchmark and randomization settings: https://robotwin-platform.github.io/.

[6]Official RoboCasa365 target splits and adaptation protocol: RoboCasa365 foundation-model learning documentation.

tasks, following the evaluation setting associated with the GR00T-N1.5 reference (NVIDIA, 2025). This setting is reported separately from RoboCasa365 mobile manipulation.

A.4 Interpretation of ablations and inference settings

Coupling and Motion Track ablations. The architecture comparison holds the data, parameter budget, optimization, and adaptation settings fixed while changing the world–action interaction. The cross-attention variant uses a separate other-stream attention sublayer after within-stream self-attention; the shared-residual joint-attention variant places both modalities in one residual sequence. JAM retains stream-specific projections and residual states while allowing both streams to attend to the combined keys and values. Motion Track *coverage* is the proportion of examples carrying track labels, whereas track *form* specifies the spatial support of the tracked points at fixed labeled exposure. Neither quantity is a data-domain sampling percentage. The geometric-control score is

$$G = \frac{S_{\text{Plus-camera}} + S_{\text{Plus-layout}} + S_{\text{RoboTwin-random}}}{3}, \tag{7}$$

where each S is the corresponding success percentage. The scaling study's task performance P instead averages success across the six benchmarks specified in Section 4.4, using RoboTwin 2.0 Clean and excluding RoboCasa365; $100 - P$ is the residual error used for the scaling fits.

Joint and action-only inference. Joint inference samples future observations and actions. Action-only inference omits future-video generation while retaining the current observation and instruction as conditions. During the associated adaptation study, dropout removes the generated-video keys and values read by the action stream with probability p; it does not remove the current visual observation. That study evaluates 20 rollouts for each of the 40 standard LIBERO tasks. The ten-demonstration transfer study is a separate setting: it compares JAM with a matched video-prior initialization using ten target demonstrations per task on ALOHA, mobile, and GR1 tasks.

Experiment	Setting and interpretation
Action-only study	Generated-video K/V dropout during adaptation; comparison of joint and action-only success on standard LIBERO.
Inference-speed study	RTX A5000; JAM uses 10 video and 10 action denoising steps, predicts 32 actions, and executes 16; latency is 1.08 s/request.
Real-world evaluation	Demonstrations recorded at 15 Hz; each model executes 24 actions per predicted chunk.

Table 14: **Distinct inference and execution settings.** The speed comparison measures end-to-end deployment latency under each method's listed configuration, rather than isolated backbone compute. Its executed action horizon differs from the real-world evaluation.

A.5 Real-world adaptation and evaluation

For the real-world experiments in Section 4.7, JAM, $\pi_{0.5}$ (Black et al., 2025a), LingBot-VA (Li et al., 2026), and Fast-WAM (Yuan et al., 2026) are all fine-tuned with low-rank adaptation (LoRA). Each model is trained for 20,000 steps on the same 50 teleoperated demonstrations per task. Data are collected at 15 Hz on a Franka Emika arm with a wrist camera and a third-person camera. All four models execute 24 actions per predicted chunk during evaluation.

Task	Manipulation objective	Train demos	Eval. trials
Bread	Place bread into the bowl.	50	20
Ring	Place the ring onto the peg.	50	20
Stack	Stack three blocks.	50	20
Grill	Close the articulated grill lid.	50	20
Noodle	Place noodles into the pan.	50	20
Total		**250**	**100**

Table 15: **Real-world task protocol.** Demonstration counts are shared across the four compared models; evaluation trial counts are per model. Object placements are randomized during evaluation.

These tasks span pick-and-place, precise alignment, multi-step stacking, and articulated-object manipulation. Per-task results count successful trials; the total aggregates all 100 trials, giving JAM a 71% overall success rate.

B Backbones and experimental baselines

The generalist VLA comparisons cover OpenVLA, an open 7B policy (Kim et al., 2025b); NORA, a compact open generalist (Hung et al., 2025); UniVLA, which learns task-centric latent actions (Bu et al., 2025); and π_0 and $\pi_{0.5}$, which use flow-based action generation and heterogeneous co-training (Black et al., 2025b;a). π_0-FAST adds frequency-space action tokenization (Pertsch et al., 2025). OpenVLA-OFT adds optimized fine-tuning and action chunking (Kim et al., 2025a). X-VLA targets scalable cross-embodiment transfer (Zheng et al., 2025). StarVLA provides a modular VLA implementation (StarVLA Community, 2026). MolmoAct adds spatial action reasoning (Lee et al., 2025), while ABot-M0 learns a low-dimensional action manifold (Yang et al., 2026a) and GR00T-N1.5 targets generalist humanoid control (NVIDIA et al., 2025; Bjorck et al., 2025). Its RoboCasa365 results follow the benchmark report (Nasiriany et al., 2026), and its RoboCasa-GR1 result follows NVIDIA's official N1.5 evaluation (NVIDIA, 2025).

World-action baselines include Mantis with disentangled visual foresight (Yang et al., 2026b), Fast-WAM with action prediction that avoids test-time future imagination (Yuan et al., 2026), DreamVLA with structured world-knowledge prediction (Zhang et al., 2025), and Cosmos Policy with video-model fine-tuning for control (Kim et al., 2026). Motus learns a latent action world model (Bi et al., 2025). AHA-WAM adapts prediction horizons and routes context asynchronously (Cai et al., 2026); X-WAM unifies action decoding with video and depth generation (Guo et al., 2026a); GigaWorld-Policy emphasizes efficient action-centered modeling (Ye et al., 2026a); and UWM couples video and action diffusion (Zhu et al., 2025). Diffusion Policy provides the task-specific action-space denoising baseline (Chi et al., 2025).

C Standard LIBERO benchmark

Table 16 reports the original four-suite LIBERO benchmark. JAM averages 97.3% and achieves 96.4% on Long. The near-saturated original suites provide a complementary view to the LIBERO-PRO and LIBERO-Plus robustness evaluations in the main text.

Standard LIBERO					
Model	**Spatial**	**Object**	**Goal**	**Long**	**Avg.**
OpenVLA	84.7	88.4	79.2	53.7	76.5
π_0	**98.8**	98.8	95.8	85.2	94.7
$\pi_{0.5}$	**98.8**	98.2	**98.0**	92.4	96.9
Fast-WAM	98.2	**100.0**	97.0	95.2	**97.6**
JAM	97.5	98.8	96.5	**96.4**	97.3

Table 16: **Standard LIBERO benchmark.** Success rate (%) on the original Spatial, Object, Goal, and Long suites. Bold marks the highest reported value in each column, including ties.

D Related-work comparison criteria

Table 1 covers January 2025–June 2026. Variables w, a, and tr denote future observations, motor commands, and Motion Tracks. Joint w, a requires coupled world–action prediction; Joint w, a, tr additionally requires supervised track generation, including joint sampling from clean current conditions with fully noised future inputs. Motion loss includes flow-proxy supervision, which alone does not establish joint generation of all three variables.

Embodied pretraining requires a completed stage with documented data, objective, schedule, and downstream checkpoint reuse; inherited priors and head warmup do not qualify. Prior-work Cross-emb. checks additionally require explicit cross-embodiment training and completed benchmarks on at least two robot embodiments from one shared pretrained checkpoint, allowing target-specific fine-tuning. JAM's entry records reuse of one shared foundation checkpoint across the reported embodiment-specific adaptations.

Attention mechanism names the main token interaction; attn. abbreviates attention. MM-DiT uses modality-specific projections and residual streams with joint attention over the combined tokens, while MoT additionally separates normalization and feed-forward parameters (Esser et al., 2024; Liang et al., 2025b). UWM and Cosmos Policy mix video/action tokens via self-attention. DreamZero and LingBot-VA use chunk-causal masking; Motus uses expert-specific projections with shared multi-stream attention. Motion Image Diffusion cross-attends to a vision-language prefix. OSCAR self-attends over video tokens with channel-concatenated skeleton conditioning. A2World denotes the base model: cross-view attention and action-conditioned adaptive normalization; its policy variant is excluded.