UNITAS: A 3D-NATIVE WORLD ACTION MODEL FOR EMBODIED MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

World action models (WAMs) aim to answer a coupled physical question: given a task instruction, what motion should the robot execute, and how will that motion change the surrounding world? Most existing WAMs build on pretrained video generators and represent world evolution through images or visual latents. Robotic interaction, however, takes place in metric three-dimensional space, while images are view-dependent projections whose pixel distances do not directly encode physical distances. We introduce UNITAS, to our knowledge the first 3D-native world action model that unifies observations, actions, and scene dynamics in a shared metric 3D frame within each interaction, using a common representation across robot embodiments and human hands. Action flow represents human hands and robot grippers as 3D point trajectories, while scene flow describes scene-point displacements conditioned on these trajectories. World-aligned 3D positional embeddings ground visual tokens with or without depth input, and a physical-time trajectory tokenizer encodes each point trajectory as one token anchored at its current 3D position. This interface supports both direct action execution and action-conditioned scene prediction. With 1.7B parameters, UNITAS achieves the best action-conditioned scene prediction on RoboTwin among the compared methods, with up to 49% lower displacement errors than PointWorld, and state-of-the-art manipulation success, including 99.8% on LIBERO and an average of 85% across real-world tasks. Code will be released upon acceptance.

1 INTRODUCTION

World Action Models (WAMs) aim to answer a coupled physical question: *given a task instruction, what motion should the robot execute, and how will that motion change the surrounding world?* Most existing WAMs couple action learning with prediction of future images or visual latents, increasingly building on pretrained video models (Yang et al., 2023; Li et al., 2025b; Zhu et al., 2025; Cen et al., 2025; Bi et al., 2025; Ye et al., 2026; Yuan et al., 2026b; Zhou et al., 2026b). Robotic interaction, however, takes place in metric three-dimensional space, and an image is only a view-dependent projection of that space.

The same robot motion, object displacement, or contact event can produce different pixel changes across views, and equal pixel displacements need not correspond to equal physical distances (Ze et al., 2024; Qu et al., 2025). Differences in camera intrinsics and camera-to-robot poses further make this mapping vary across embodiments. Evaluations of video-based WAMs report sensitivity to viewpoint and robot-state changes (Zhang et al., 2026b), and a photorealistic prediction may still misrepresent object displacement, deform rigid geometry, or allow static regions to drift (Huang et al., 2026). Such visual representations leave metric motion and contact relationships implicit for spatial reasoning, prediction, and control.

Recent methods incorporate depth, optical flow, scene flow, or reconstructed 4D geometry into prediction (Zhen et al., 2025; Guo et al., 2026; Zhao et al., 2026; Wang et al., 2026b) or ground action generation in explicit 3D structure (Shridhar et al., 2023; Ze et al., 2024; Zhen et al., 2024; Qu et al., 2025). Yet geometry often serves as an input cue or auxiliary prediction target, rather than a shared interface for action generation and scene dynamics. PointWorld (Huang et al., 2026) directly predicts metric point flows conditioned on externally supplied robot motion, rather than generating that motion from a task instruction. A 3D-native WAM should place observations, actions, and re-

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

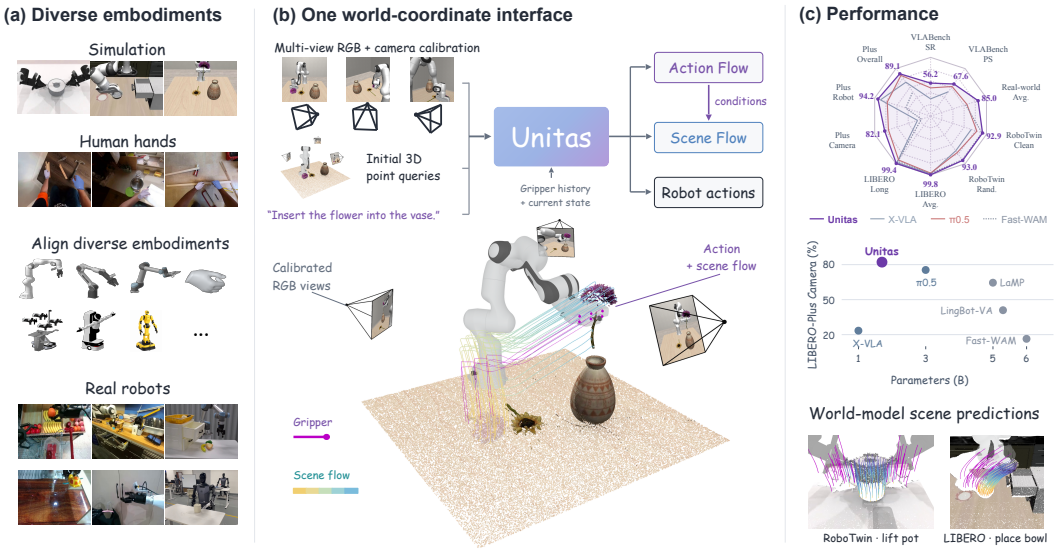

Figure 1: **UNITAS is a 3D-native world action model.** Observations, action flow, and scene flow share a metric frame within each prediction window, with a common representation across embodiments. With 1.7B parameters, UNITAS achieves leading manipulation results among the compared methods and predicts action-conditioned scene trajectories.

sulting scene changes in a shared metric frame, enabling task-conditioned behavior, consequence prediction, and executable control.

We introduce UNITAS, **UNI**fied **T**rajectories for **A**ction and **S**cene flow. To our knowledge, it is the first 3D-native world action model that aligns observations, actions, and scene dynamics in a shared metric frame within each prediction window, using a common representation across robot embodiments and human hands, as illustrated in Fig. 1.

Realizing this 3D-native world action model presents three challenges. First, visual observations must be grounded in metric space even when depth is unavailable. We propose *world-aligned 3D position embeddings* derived from observed depth or learned weights over candidate positions along calibrated camera rays. Second, human hands and robot grippers have different morphologies and control interfaces. We propose *action flow*, which represents their motion as trajectories of corresponding 3D points in the shared frame. Third, trajectories differ in frame rate and duration across data sources. Representing each timestep as a token allocates backbone capacity across unequal physical-time intervals. We propose a *physical-time trajectory tokenizer*, which aligns each trajectory to elapsed time and encodes the whole trajectory as one token before the backbone. *Scene flow* describes scene-point displacements decoded from trajectory tokens conditioned on action flow.

In the backbone, observation tokens are grounded in world coordinates, while action-flow tokens across embodiments and scene-flow tokens are anchored at each point's current 3D position, allowing the network to model motion through features associated with fixed spatial anchors. Given the instruction and current observations, a layer-coupled Mixture-of-Transformers jointly models point trajectories and executable commands. This supports two modes: *policy mode* directly generates executable actions, while *simulator mode* predicts scene changes conditioned on externally supplied action flow. A causal attention mask enables key–value caching of shared context across denoising steps, reducing repeated computation and accelerating inference.

We evaluate UNITAS on action-conditioned scene prediction and robotic manipulation. In simulator mode, it leads the compared methods across all four RoboTwin scene-prediction metrics, reducing Moving ADE, Moving FDE, and Static ADE by 35%, 21%, and 49%, respectively, relative to PointWorld (Huang et al., 2026). In policy mode, it achieves 99.8% success on LIBERO and outperforms larger video-based WAMs on RoboTwin. It leads the compared methods in VLABench success rate and progress score, and reaches 89.1% success on LIBERO-Plus with strong robustness to camera and robot-state changes. Real-robot success reaches 90% on *Organize Books* and 65% on *Pack the Ball*, exceeding both baselines. Ablations show that world-aligned visual grounding, action- and scene-flow supervision, and motion history improve policy performance and robustness.

Our contributions are:

- To our knowledge, UNITAS is the first 3D-native world action model that aligns observations, actions, and scene dynamics in a shared metric frame, using a common representation across robot embodiments and human hands.
- Action flow represents human hands and robot grippers through point trajectories, while scene flow represents the environmental response conditioned on that motion. Both policy and simulator modes use this interface.
- World-aligned positional embeddings ground visual observations with or without depth, and a physical-time trajectory tokenizer encodes each point trajectory as one token spatially anchored at its current 3D position.
- With 1.7B parameters, UNITAS achieves leading action-conditioned scene prediction and strong manipulation performance in simulation and on real hardware.

2 RELATED WORK

Video world–action modeling. Video-based policies use future prediction for control (Hu et al., 2024), with early WAMs coupling prediction and action learning through shared representations and generative objectives (Li et al., 2025b; Zhu et al., 2025; Cen et al., 2025; Xu et al., 2025a). Video-generation priors subsequently support joint video–action learning and future-conditioned policies (Liao et al., 2025; Shen et al., 2025; Bi et al., 2025; Kim et al., 2026; Li et al., 2026; Wang et al., 2026a), while heterogeneous robot and video-only data support cross-task and cross-embodiment transfer (Ye et al., 2026; Zhang et al., 2026a). Efficient variants omit future generation at deployment or replace it with latent subgoals (Yuan et al., 2026b; Chen et al., 2026a). These models primarily represent world evolution in images or visual latents. Reported sensitivity to viewpoint and robot initial-state changes (Zhang et al., 2026b; Hong et al., 2026) highlights the importance of grounding perception, world evolution, and executable actions in a shared physical space.

Policy learning in 3D space. Explicit 3D representations improve spatial understanding for manipulation (Shridhar et al., 2023; Ze et al., 2024; Zhen et al., 2024; Qu et al., 2025). Geometry-aware world models augment visual prediction with image-aligned geometric signals (Zhen et al., 2025; Guo et al., 2026; Zhao et al., 2026). In policy learning, LaMP (Wang et al., 2026b) and GeomVLA (Xiong et al., 2026) use intermediate motion features from trajectory denoising to condition action generation. PointWorld (Huang et al., 2026) follows the complementary direction, predicting scene flow conditioned on robot motion for planning. UNITAS jointly models agent motion and its conditional scene response through point flows, while supporting direct command execution without generating scene flows.

Unified action representations for manipulation. Cross-embodiment learning must reconcile different morphologies and control interfaces (Chi et al., 2024). Generalist policies use embodiment-specific prompts, canonical state–action formats, and aligned end-effector conventions (Liu et al., 2024; Zheng et al., 2025; Yuan et al., 2026a; Xiaomi Robotics Team et al., 2026). Point flows provide transferable motion targets across embodiments (Yuan et al., 2024; Bharadhwaj et al., 2024; Xu et al., 2025b), connecting object-motion prediction to control (Zhi et al., 2025; Dharmarajan et al., 2025) and representing robot and object trajectories (Huang et al., 2026; Lin et al., 2026). UNITAS unifies human-hand, robot-gripper, and scene trajectories with a shared physical-time tokenizer.

3 METHODOLOGY

We first revisit video-based world action modeling, then formulate UNITAS in a shared metric 3D frame, as illustrated in Fig. 2.

3.1 REVISITING VIDEO WORLD ACTION MODELS

Let o denote current image observations, ℓ the instruction, $a_{1:H}$ an action chunk, and $v_{1:T}$ future images. Video-based WAMs represent future interactions through images or visual latents. For models that condition actions on predicted futures, the action distribution can be written as

$$p(a_{1:H} \mid o, \ell) = \int p(v_{1:T} \mid o, \ell)\, p(a_{1:H} \mid o, \ell, v_{1:T})\, dv_{1:T}. \tag{1}$$

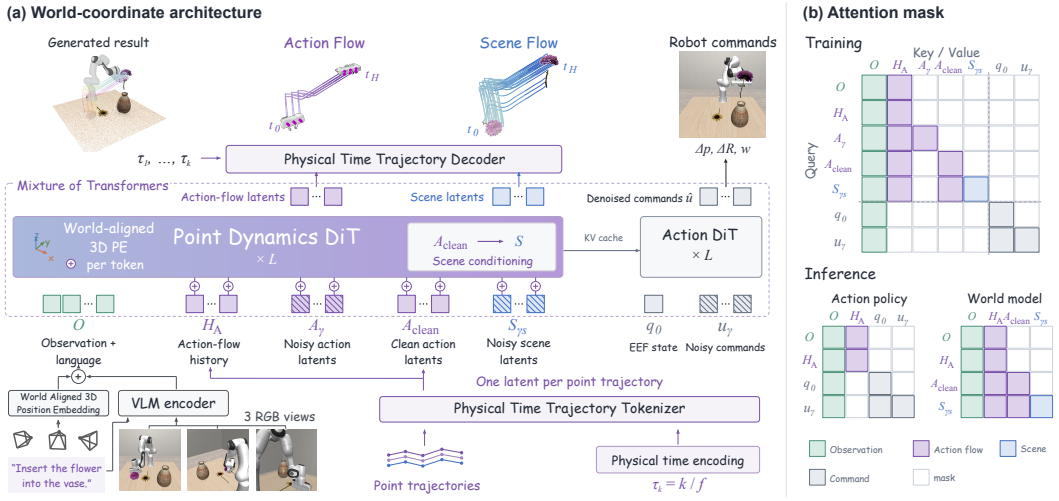

Figure 2: **UNITAS architecture and attention masks.** Left: The point dynamics expert models both action and scene flows, while the action expert generates the end-effector action chunk. Scene flow generation is conditioned on clean demonstrated action flow during training and generated or supplied action flow at inference. Right: Colored/white cells indicate allowed/blocked attention. Policy-only inference omits both future action-flow and scene-flow tokens.

Future videos and actions may be generated jointly or sequentially. Fast-WAM (Yuan et al., 2026b) instead retains video co-training but omits future-video generation at inference:

$$p_\theta(a_{1:H} \mid o, \ell) = p_\theta(a_{1:H} \mid z(o, \ell)), \qquad \mathcal{L} = \mathcal{L}_{\text{act}} + \lambda \mathcal{L}_{\text{vid}}. \tag{2}$$

Its video backbone encodes the current observation and instruction into $z(o, \ell)$ in one forward pass, while the action expert still performs iterative denoising. This removes future-video sampling, but the shared representation remains image-based, without an explicit metric 3D frame.

3.2 3D-NATIVE WORLD ACTION FORMULATION

At time t, context $\mathcal{C}$ comprises the instruction ℓ, multi-view observations $\mathcal{O}_t = \{\mathcal{O}_{t,i}\}_{i=1}^{N_v}$, camera calibration $\{\mathbf{K}_i, \mathbf{T}_{t,i}^W\}_{i=1}^{N_v}$, and historical action flow $\mathcal{H}_\text{A}$. Here $\mathbf{K}_i$ denotes camera intrinsics, and $\mathbf{T}_{t,i}^W \in \text{SE}(3)$ maps camera coordinates into the metric frame $\mathcal{F}^W$, fixed within each prediction window. Action prediction additionally uses the current end-effector state $\mathbf{q}_0$.

Given query points and future timestamps, we predict action flow $\mathbf{A}$ for the interacting body and scene flow $\mathbf{S}$ for the environment, both as displacement trajectories in $\mathcal{F}^W$. A separate action branch predicts an executable end-effector action chunk $\mathbf{u}$. Our model factorizes their joint distribution as

$$p_\theta(\mathbf{A}, \mathbf{u}, \mathbf{S} \mid \mathcal{C}, \mathbf{q}_0) = p_\theta(\mathbf{A} \mid \mathcal{C})\, p_\theta(\mathbf{u} \mid \mathcal{C}, \mathbf{q}_0)\, p_\theta(\mathbf{S} \mid \mathbf{A}, \mathcal{C}). \tag{3}$$

Action flow and the action chunk can be sampled in parallel; scene prediction then conditions on generated action flow $\widehat{\mathbf{A}}$ or an externally supplied trajectory $\mathbf{A}_\text{ext}$ in simulator mode. Policy-only execution evaluates only $p_\theta(\mathbf{u} \mid \mathcal{C}, \mathbf{q}_0)$, omitting both point flows. The following sections describe spatial grounding, the shared trajectory representation, and the coupled experts that implement these factors.

3.3 WORLD-ALIGNED OBSERVATIONS

Frame alignment. Within each prediction window, observations, robot or human motion, and scene trajectories share one fixed metric frame. Robot demonstrations use a robot-base or shared embodiment frame. Egocentric human demonstrations use the camera frame of the window anchor, held fixed throughout the window. Appendix A.1 lists the dataset conventions.

World-aligned 3D position embedding. Each visual token receives a position embedding in the world frame. For a view with valid depth, $\mathcal{O}_{t,i} = (\mathbf{I}_{t,i}, \mathbf{D}_{t,i})$ contains the RGB image and the depth along the optical axis. The vision encoder maps the image to a token grid $\mathbf{F}_{t,i} = \text{Enc}_\text{vis}(\mathbf{I}_{t,i})$. We back-project each pixel and transform it into world coordinates,

$$\mathbf{P}_{\text{camera},i}(u, v) = \mathbf{D}_{t,i}(u, v)\, \mathbf{K}_i^{-1} [u, v, 1]^\top, \qquad \begin{bmatrix} \mathbf{P}_{\text{world},i}(u, v) \\ 1 \end{bmatrix} = \mathbf{T}_{t,i}^W \begin{bmatrix} \mathbf{P}_{\text{camera},i}(u, v) \\ 1 \end{bmatrix}. \tag{4}$$

Here, (u, v) denotes pixel-center coordinates, and the time index is omitted from the coordinate maps. A fixed Fourier map $\boldsymbol{\psi}$ encodes the world position. Inside each patch, $\mathrm{Enc}_{3\mathrm{D}}$ applies $\boldsymbol{\psi}$ and projects the result to the visual-token dimension:

$$\mathbf{E}_{\mathrm{pos},t,i} = \mathrm{Enc}_{3\mathrm{D}}(\mathbf{P}_{\mathrm{world},i}), \qquad \widetilde{\mathbf{F}}_{t,i} = \mathbf{F}_{t,i} + \mathbf{E}_{\mathrm{pos},t,i}. \tag{5}$$

Spatial grounding without observed depth. Without depth, the position embedding is computed from a weighted combination of positional features along the camera ray. For a patch centered at (u, v), we draw $M = 64$ depths uniformly from 0.1 to $2.0\,\mathrm{m}$, back-project them by Eq. 4, and score them with the patch feature $\mathbf{f}$:

$$\begin{aligned} \alpha_m &= \mathrm{softmax}_m\left(\frac{(\mathbf{W}_q\mathbf{f})^\top \mathbf{W}_k \boldsymbol{\psi}(\mathbf{p}_{\mathrm{camera},m})}{\sqrt{d}}\right), \\ \mathbf{E}_{\mathrm{ground}} &= \mathbf{W}_o \sum_{m=1}^{M} \alpha_m \mathbf{W}_v \boldsymbol{\psi}(\mathbf{p}_{\mathrm{world},m}). \end{aligned} \tag{6}$$

Here $\mathbf{f} \in \mathbb{R}^{d_f}$ is the patch feature and $\boldsymbol{\psi}(\cdot) \in \mathbb{R}^{d_\psi}$, with $\mathbf{W}_q \in \mathbb{R}^{d\times d_f}$, $\mathbf{W}_k \in \mathbb{R}^{d\times d_\psi}$, and $\mathbf{W}_v \in \mathbb{R}^{d\times d_\psi}$. Keys encode camera-frame positions, and values encode the same candidates in the world frame. $\mathbf{E}_{\mathrm{ground}}$ is added to the visual tokens in place of $\mathbf{E}_{\mathrm{pos},t,i}$ in Eq. 5. Where depth is reliable, we use depth-derived embeddings and supervise the routing weights by matching Fourier features:

$$\mathcal{L}_{\mathrm{PE}} = \sum_{i=1}^{N_v} \mathbb{E}_{\mathcal{R}\in\mathcal{V}_i} \left\| \sum_{m=1}^{M} \alpha_m \boldsymbol{\psi}(\mathbf{p}_{\mathrm{world},m}) - \mathrm{sg}\Big(\mathrm{Avg}_{(u,v)\in\mathcal{R}}\, \boldsymbol{\psi}(\mathbf{P}_{\mathrm{world},i}(u,v))\Big) \right\|_2^2. \tag{7}$$

Here $\mathcal{V}_i$ contains regions with reliable depth in view i, and sg denotes stop-gradient. Only valid-depth pixels contribute to the target average. With this module, UNITAS pretrains jointly on RGB and RGB-D data, and depth may be absent at inference. Appendix B evaluates the router when depth is removed at test time.

3.4 PHYSICAL-TIME POINT-FLOW REPRESENTATION

Action flow and scene flow. In the same world frame, action and scene flows describe displacement from each query point's current position. For query n at $\mathbf{x}_{t,n}$, the target is $\{\Delta\mathbf{x}_{n,k}\}_{k=1}^{H}$, with $\Delta\mathbf{x}_{n,k} = \mathbf{x}_{t+\tau_k,n} - \mathbf{x}_{t,n}$. The elapsed time is τ_k, and point identity is preserved throughout the trajectory.

For robots, we sample $N_g = 20$ points from the URDF-defined gripper geometry of each arm and retain their link-local coordinates. Forward kinematics maps these points into the world frame at each recorded joint configuration, preserving correspondence through arm and finger motion. Human demonstrations use 21 MANO-derived keypoints (Romero et al., 2017) per observed hand. Following PointWorld (Huang et al., 2026), we mask robot pixels using the URDF and current joint state, back-project the remaining RGB-D pixels, and sample N_s scene query points. Simulator states supply ground-truth correspondences where available and scene-flow loss is applied only to valid trajectories. At inference, scene prediction requires current 3D query coordinates from depth or supplied geometry.

Physical-time trajectory tokenizer. To better align different trajectories within diverse datasets, we encode time in seconds rather than frame indices, using $\tau_k = k/f$ at sampling rate f. A fixed basis $\boldsymbol{\phi}_{\mathrm{time}}$ of $\tau/T_{\max}$, with $T_{\max} = 3\,\mathrm{s}$ shared across datasets, gives equal elapsed times the same features; history uses negative offsets. An input MLP maps each displacement to $\mathbf{h}_{n,k}$, and temporal MLPs map $\boldsymbol{\phi}_{\mathrm{time}}(\tau_k)$ to FiLM parameters $(\mathbf{a}_k, \mathbf{b}_k)$ for the encoder and the decoder. The encoder pools the trajectory into one latent, and the decoder reads that latent back at a requested time:

$$\begin{aligned} \mathbf{z}_n &= \frac{1}{\sqrt{H}} \sum_{k=1}^{H} \left[(\mathbf{1} + \mathbf{a}_k^{\mathrm{enc}}) \odot \mathbf{h}_{n,k} + \mathbf{b}_k^{\mathrm{enc}}\right], \\ \widehat{\Delta\mathbf{x}}_{n,k} &= \mathrm{MLP}_{\mathrm{out}}\left[(\mathbf{1} + \mathbf{a}_k^{\mathrm{dec}}) \odot \mathbf{z}_n + \mathbf{b}_k^{\mathrm{dec}}\right]. \end{aligned} \tag{8}$$

Padding is excluded from the sum. Action flow, scene flow, and history share this tokenizer, so each point is one token regardless of trajectory length or sampling density. Flow tokens are anchored only at current world-frame points: positional embeddings as in Eq. 5 are added throughout denoising. Decoding gives $\widehat{\mathbf{x}}_{t+\tau_k,n} = \mathbf{x}_{t,n} + \widehat{\Delta\mathbf{x}}_{n,k}$. Appendix A.2 gives the basis and the reconstruction stage.

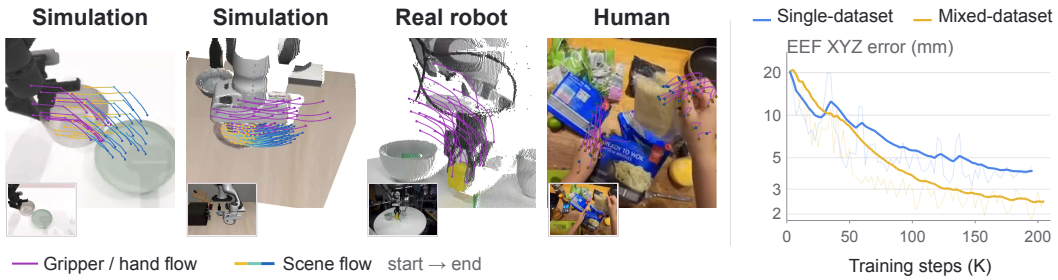

Figure 3: **Shared point-flow representations across data sources.** Left: gripper/hand trajectories (purple) and scene trajectories (yellow–blue), with RGB insets. Right: RoboTwin end-effector XYZ error (mm). Thick curves show exponential moving averages (EMA) of the thin curves.

3.5 Model Architecture

Coupled experts. UNITAS uses a Mixture-of-Transformers (Liang et al., 2024) with two diffusion transformers (Peebles & Xie, 2023): a point dynamics expert for action and scene flows, and an action expert for the end-effector action chunk $\mathbf{u}$. The point dynamics expert processes spatially grounded primary-view VLM tokens, auxiliary-view features from the shared vision tower, and historical and future flow latents. At each layer, the smaller action expert reads the dynamics expert's observation/language keys and values (KV), together with its own current-state and noisy command tokens. It predicts relative end-effector translation, rotation, and gripper openness at each future timestep.

Attention structure. Let O and H_{A} be the observation/language and history tokens, and let A_{γ}, A_{clean}, and $S_{\gamma_{\mathrm{s}}}$ be the noisy action-flow, clean action-flow, and noisy scene latents. The context $[O, H_{\mathrm{A}}]$ is bidirectional. Each action-flow group attends to this context and to itself. Scene tokens attend to the context, to A_{clean}, and to one another, but not to A_{γ}, so scene prediction conditions on action flow without entering action-flow denoising. Action chunk tokens attend to the observation KV, the current end-effector state, and one another. They do not attend to future action-flow or scene-flow tokens, so policy-only inference can skip both future-flow branches, as shown in Fig. 2.

Training. We first train the trajectory tokenizer to reconstruct historical and future action flow and available scene trajectories, then freeze it. Both experts learn through rectified flow (Liu et al., 2022). For a clean target $\mathbf{y}$, either a normalized trajectory latent or an action chunk, and noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, we form $\mathbf{y}_{\gamma} = (1-\gamma)\boldsymbol{\epsilon} + \gamma\mathbf{y}$ and regress $\mathbf{y} - \boldsymbol{\epsilon}$. Action flow and action chunk share γ but use independent noise, while scene flow draws γ_{s} independently. During training, scene prediction conditions on clean action-flow latents from demonstrations. The objective is

$$\mathcal{L} = \lambda_{\mathrm{A}}\mathcal{L}_{\mathrm{A}} + \lambda_{\mathrm{s}}\mathcal{L}_{\mathrm{scene}} + \lambda_{\mathrm{ctrl}}\mathcal{L}_{\mathrm{ctrl}} + \lambda_{\mathrm{PE}}\mathcal{L}_{\mathrm{PE}}. \tag{9}$$

The first three terms use mean squared error (MSE) losses to supervise action flow, scene flow, and the action chunk, with $\lambda_{\mathrm{A}} = \lambda_{\mathrm{s}} = \lambda_{\mathrm{ctrl}} = 1$. $\mathcal{L}_{\mathrm{PE}}$ is Eq. 7, with $\lambda_{\mathrm{PE}} = 0.1$. Each loss uses only available annotations, with missing targets and absent arm slots masked.

Inference. Action flow and action chunk are denoised in parallel on one Euler schedule. Generated action flow, or an externally supplied trajectory in simulator mode, then conditions scene denoising. The context and clean action-flow keys and values are cached and reused across scene-denoising steps. Policy-only execution keeps the observation and history context and denoises only $\mathbf{u}$, skipping both point dynamics branches. Appendix A.3 gives the schedule, widths, and optimization.

4 Policy Evaluation

Our policy evaluation tests whether representing interaction in a shared metric space supports effective robot control. We examine task completion across simulation benchmarks, robustness when observations and initial states change, and execution on real-world robots.

4.1 Experimental Setup

Training settings. We initialize the vision-language backbone from Florence-2 (Xiao et al., 2023) and randomly initialize all remaining components. We consider two training settings: (1) *with pretraining*, UNITAS is pretrained on a mixture of real-robot, simulation, and human hand demonstrations, then fine-tuned on each downstream benchmark or dataset; (2) *without pretraining*, UNITAS

(w/o Pretrain) is fine-tuned directly on each target benchmark or dataset without the mixed-dataset pretraining stage. More detailed training configurations are provided in the appendix.

Baselines. We compare against VLA policies (Zheng et al., 2025; Sun et al., 2026; Black et al., 2024; Physical Intelligence et al., 2025; Yang et al., 2026), video-based WAMs (Kim et al., 2026; Liao et al., 2025; Bi et al., 2025; Li et al., 2026; Yuan et al., 2026b), and 3D-aware policies (Qu et al., 2025; Wang et al., 2026b; Guo et al., 2026; Zhou et al., 2026a).

4.2 Simulation Benchmarks

We evaluate UNITAS on single-arm and bimanual manipulation benchmarks. LIBERO (Liu et al., 2023) includes Spatial, Object, Goal, and long-horizon LIBERO-10 suites; RoboTwin 2.0 (Chen et al., 2025) covers 50 bimanual tasks under Clean and Randomized conditions. VLABench (Zhang et al., 2024) tests object and instruction generalization across five tracks, measured by success rate (SR), progress score (PS), and intention score (IS). LIBERO-Plus (Fei et al., 2025) evaluates robustness to seven types of observation, instruction, and initial-state perturbations. Detailed protocols and per-task results are provided in the appendix.

Manipulation performance. Tab. 1 reports policy performance across the evaluated simulation benchmarks. On LIBERO, UNITAS achieves a state-of-the-art average success rate of 99.8%, with strong results on spatial and long-horizon tasks. Without mixed-dataset pretraining, it achieves 99.5%, also exceeding all baselines. On RoboTwin 2.0, UNITAS achieves 92.94% and 93.04% under Clean and Randomized conditions, respectively. Its 1.7B model outperforms the compared methods while using fewer parameters than Fast-WAM (6B) and Motus (8B). On VLABench, it achieves 56.2% SR compared with 48.1% for $\pi_{0.5}$, alongside a higher progress score.

Robustness to distribution shifts. As shown in Tab. 1, UNITAS achieves the highest overall success rate of 89.1% on LIBERO-Plus. Appendix C reports the perturbation-wise results. Under camera perturbations, it achieves 82.1%, compared with 75.8% for Cosmos Policy and 64.5% for LaMP. Both training settings show strong viewpoint robustness, consistent with world-coordinate grounding of visual observations.

Mixed-dataset pretraining improves overall success by 2.6 percentage points, with the largest gains under robot initial-state changes from 84.8% to 94.2%, and language perturbations from 77.6% to 85.0%. These improvements suggest better generalization to changes in both physical configuration and instruction phrasing. As illustrated in Fig. 3, adding other data sources while keeping the number of RoboTwin samples per batch unchanged yields lower RoboTwin test XYZ error. Together, these results support positive cross-dataset transfer under our unified modeling formulation.

Table 1: Policy performance (%, ↑). Our 1.7B model is evaluated with and without mixed-dataset pretraining. Bold and underlined scores denote the best and second-best results, respectively. LIBERO-Plus ranks use one-decimal precision. Dashes denote unavailable results.

Method	Params (B)	LIBERO					LIBERO-Plus	RoboTwin 2.0		VLABench		
		Spatial	Object	Goal	10	Avg.	Overall	Clean	Rand.	SR	PS	IS
VLA Policies												
X-VLA	1.0	98.2	98.6	97.8	97.6	98.1	71.4	72.88	72.84	29.4	51.2	<u>70.2</u>
VLA-JEPA	2.0	96.2	99.6	97.2	95.8	97.2	77.9	–	–	–	–	–
π_0	3.0	96.8	98.8	95.8	85.2	94.15	53.6	65.92	58.40	29.4	44.1	55.0
$\pi_{0.5}$	3.0	98.8	98.2	98.0	92.4	96.9	85.7	82.74	76.76	48.1	62.3	64.9
ABot-M0	4.0	98.8	<u>99.8</u>	99.0	96.6	98.6	80.5	80.42	81.16	–	–	–
Video-Based WAMs												
Cosmos Policy	2.0	98.1	**100.0**	98.2	97.6	98.5	82.2	–	–	–	–	–
GE-Act	2.2	98.2	97.6	95.8	94.4	96.5	80.3	–	–	–	–	–
LingBot-VA	5.3	98.5	99.6	97.2	98.5	98.5	69.5	<u>92.93</u>	91.55	–	–	–
Fast-WAM	6.0	98.2	**100.0**	97.0	95.2	97.6	51.5	91.88	<u>91.78</u>	–	–	–
Motus	8.0	96.8	<u>99.8</u>	96.6	97.6	97.7	–	88.66	87.02	–	–	–
3D-Aware Policies												
SpatialVLA	3.5	88.2	89.9	78.6	55.5	78.1	–	–	–	–	–	–
LaMP	5.0	<u>99.4</u>	<u>99.8</u>	97.4	96.7	98.3	79.3	–	–	–	–	–
X-WAM	6.7	–	–	–	–	–	–	89.8	90.7	–	–	–
Dex-BEV	–	98.2	98.0	97.8	97.0	97.8	–	76.0	42.0	–	–	–
UNITAS (w/o Pretrain)	1.7	**99.8**	<u>99.8</u>	<u>99.4</u>	<u>99.0</u>	<u>99.5</u>	<u>86.5</u>	91.08	90.74	<u>50.3</u>	<u>64.3</u>	**71.2**
UNITAS	1.7	**99.8**	**100.0**	**100.0**	**99.4**	**99.8**	**89.1**	**92.94**	**93.04**	**56.2**	**67.6**	69.3

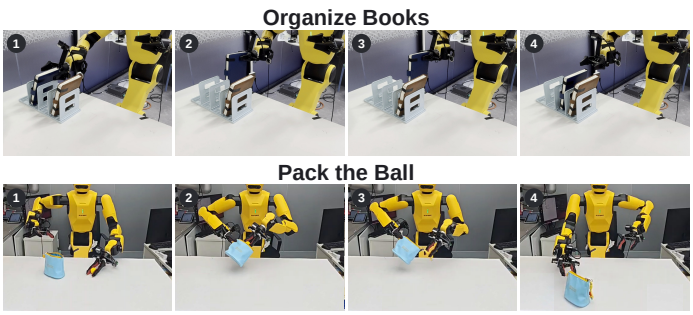

Figure 4: Qualitative real-robot rollouts on *Organize Books* and *Pack the Ball*. Numbered keyframes show task progression through precise placement and sequential manipulation.

Table 2: Quantitative comparison results of real-robot experiments (20 trials per task).

Organize Books	
$\pi_{0.5}$	15/20 (75%)
Fast-WAM	13/20 (65%)
UNITAS	**18/20 (90%)**
Pack the Ball	
$\pi_{0.5}$	11/20 (55%)
Fast-WAM	12/20 (60%)
UNITAS	**16/20 (80%)**

4.3 REAL-WORLD EVALUATION

We evaluate two real-world tasks: *Organize Books* requires aligning books with shelf compartments, while *Pack the Ball* requires placing a ball inside a bag and closing its zipper. We compare against $\pi_{0.5}$ and Fast-WAM using 20 trials per method per task. As shown in Tab. 2, UNITAS achieves 90% and 80% success on the two tasks, respectively, outperforming both baselines. Fig. 4 shows representative executions.

4.4 ABLATION STUDY

We ablate visual 3D positional encoding, action-flow supervision, action-flow history tokens, and scene-flow supervision on LIBERO-Plus without mixed-dataset pretraining. As shown in Tab. 3, the first three ablations use a reference model without scene-flow supervision, which achieves 85.74% overall success. Removing 3D positional encoding reduces success to 81.35%, with the largest drop under camera perturbations, from 79.11% to 66.73%. This highlights the role of world-coordinate grounding in relating visual observations to robot motion and improving generalization across camera viewpoints.

Removing action-flow supervision reduces overall success to 78.89%, supporting the value of geometric motion supervision for action learning. Removing action-flow history tokens reduces overall success to 75.92%, with a drop of 19.87 percentage points under robot initial-state perturbations, suggesting that recent motion provides useful context beyond the current state. Removing all three components yields a VLA-style baseline with 67.58% success. Adding scene-flow supervision to the reference model improves overall success from 85.74% to 86.47%, suggesting that future scene prediction helps learn representations beneficial to action prediction, consistent with findings in other WAMs (Yuan et al., 2026b).

5 ACTION-CONDITIONED WORLD MODELING

Evaluation setup. We evaluate action-conditioned scene prediction on held-out RoboTwin rollouts using matched initial observations and robot-motion conditions, with simulator states providing ground-truth point trajectories. We compare against PointWorld (Huang et al., 2026), which directly predicts 3D trajectories, and video-based Ctrl-World (Guo et al., 2025) and X-WAM (Guo et al., 2026). Video predictions are lifted to 3D using SpatialTrackerV2 (Xiao et al., 2025), with Depth Anything 3 (Lin et al., 2025) for Ctrl-World and native depth predictions for X-WAM. Both are cal-

Table 3: LIBERO-Plus ablations without mixed-dataset pretraining. Checkmarks indicate enabled components. Success rates (%) are reported for each perturbation; Overall pools all 10,030 evaluation episodes.

Components				Perturbation Type							Overall
Visual 3D PE	Action-flow Loss	History Tokens	Scene-flow Loss	Camera	Robot	Language	Light	Background	Noise	Layout	
				50.47	59.48	67.14	81.00	74.44	74.70	71.80	67.58
	✓	✓		66.73	79.29	75.21	94.57	92.19	84.13	84.46	81.35
✓		✓		75.05	76.13	66.17	90.72	87.45	86.45	75.74	78.89
✓	✓			69.04	64.71	72.41	88.88	80.48	81.01	79.80	75.92
✓	✓	✓		79.11	84.58	77.10	96.32	**94.42**	89.07	85.05	85.74
✓	✓	✓	✓	**81.86**	**84.84**	**77.62**	**96.67**	**94.42**	**89.82**	**85.11**	**86.47**

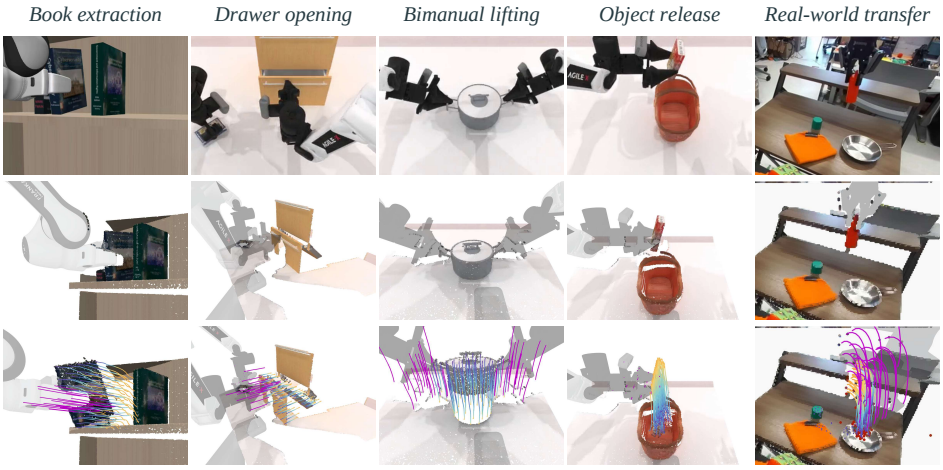

Figure 5: **Action-conditioned scene-flow predictions.** Top: observed RGB images. Middle: initial point clouds. Bottom: predicted endpoint point clouds and scene trajectories (color gradient), conditioned on recorded gripper trajectories (purple). Translucent robot meshes indicate the corresponding poses.

ibrated using only initial-frame ground-truth depth. We report Moving ADE and FDE for trajectory and endpoint accuracy, Static ADE for background drift, and Moving Acc@1cm for the fraction of moving-point predictions within 1 cm of ground truth. Detailed evaluation protocols are provided in Appendix F.1.

Table 4: Action-conditioned scene prediction on RoboTwin. Errors are in cm (↓); Acc@1cm is in % (↑). DA3 and STv2 denote Depth Anything 3 and SpatialTrackerV2, respectively.

Method	Moving ADE ↓	Static ADE ↓	Moving FDE ↓	Moving Acc@1cm ↑
PointWorld	<u>0.656</u>	<u>0.084</u>	<u>1.050</u>	<u>88.22</u>
Ctrl-World + DA3 + STv2	3.310	0.369	5.503	41.54
X-WAM + STv2	2.286	0.122	3.886	49.69
UNITAS	**0.427**	**0.043**	**0.825**	**92.31**

Action-conditioned scene prediction. UNITAS achieves the best results across all four metrics, as shown in Tab. 4, reducing Moving ADE from 0.656 to 0.427 cm relative to PointWorld while improving moving-point accuracy and reducing static-region drift. Fig. 5 illustrates object motion and resulting spatial configurations conditioned on gripper trajectories. The simulation examples use training clips and DROID example shows real-world scene prediction without real-world scene-flow supervision.

Action-scene consistency. We test whether predicted scene changes agree with the consequences of the model's own actions on three RoboTwin tasks. UNITAS achieves the best consistency between predicted scene evolution and executed actions among the compared WAMs. Appendix F.2 provides the evaluation details and full results.

6 CONCLUSION

UNITAS advances a physical-space formulation of world action modeling: what the robot observes, how it moves, and how the scene responds are expressed within a common metric reference. Making interaction geometry an explicit prediction target connects world modeling with action learning and provides a shared representation across differing embodiments and observation views. The strong policy performance observed in simulation supports the value of this formulation for manipulation. More broadly, this work positions physical interaction as a unifying basis for embodied learning, with robot behavior and environmental change represented as complementary parts of the same process. Extending this foundation to reliable prediction over longer interactions and broader real-world deployment is a promising direction for general-purpose robot intelligence.

AI USE STATEMENT

Generative AI tools were used to assist with literature review to identify related work and with language polishing.

REFERENCES

AgiBot-World-Contributors, Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. AgiBot World Colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.

Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2Act: Predicting point tracks from internet videos enables generalizable robot manipulation. *arXiv preprint arXiv:2405.01527*, 2024. doi: 10.48550/arXiv.2405.01527. URL http://arxiv.org/abs/2405.01527.

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. *arXiv preprint arXiv:2512.13030*, 2025. doi: 10.48550/arXiv.2512.13030. URL http://arxiv.org/abs/2512.13030.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael 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, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. doi: 10.48550/arXiv.2410.24164. URL http://arxiv.org/abs/2410.24164.

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

Jialei Chen, Kai Wang, Kang Chen, Shuaihang Chen, Feng Gao, Wenhao Tang, Zhiyuan Li, Weilin Liu, Zhuyu Yao, Boxun Li, Yuanbo Xu, and Chao Yu. LaWAM: Latent world action models for efficient dynamics-aware robot policies. *arXiv preprint arXiv:2606.15768*, 2026a. doi: 10.48550/arXiv.2606.15768. URL http://arxiv.org/abs/2606.15768.

Ronghan Chen, Yandan Yang, Zuojin Tang, Dongjie Huo, Tong Lin, Haoning Wu, Haoyun Liu, Yuzhi Chen, Lulu Zheng, Botai Yuan, Tianlun Li, Mingxin Wang, Dekang Qi, Bin Hu, Wei Mei, Yuze Xuan, Haolong Yang, Yanqing Zhu, Mu Xu, Zhiheng Ma, and Xinyuan Chang. ABot-M0.5: Unified Mobility-and-Manipulation World Action Model. *arXiv preprint arXiv:2607.00678*, 2026b. URL http://arxiv.org/abs/2607.00678.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. *arXiv preprint arXiv:2402.10329*, 2024. doi: 10.48550/arXiv.2402.10329. URL http://arxiv.org/abs/2402.10329.

Karthik Dharmarajan, Wenlong Huang, Jiajun Wu, Li Fei-Fei, and Ruohan Zhang. Dream2Flow: Bridging video generation and open-world manipulation with 3D object flow. *arXiv preprint arXiv:2512.24766*, 2025. doi: 10.48550/arXiv.2512.24766. URL http://arxiv.org/abs/2512.24766.

Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Chenxi Wang, Junbo Wang, Haoyi Zhu, and Cewu Lu. RH20T: A comprehensive robotic dataset for learning diverse skills in one-shot. *arXiv preprint arXiv:2307.00595*, 2023. URL https://arxiv.org/abs/2307.00595.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: In-depth Robustness Analysis of Vision-Language-Action Models. *arXiv preprint arXiv:2510.13626*, 2025. URL http://arxiv.org/abs/2510.13626.

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

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, et al. Ctrl-World: A Controllable Generative World Model for Robot Manipulation, 2025. URL https://arxiv.org/abs/2510.10125v1.

Jihoon Hong, Julian Skifstad, Qiyue Dai, Alice Chan, and Glen Chou. Steering robustness into world action models via mechanistic interpretability and optimal control. *arXiv preprint arXiv:2607.14943*, 2026. URL http://arxiv.org/abs/2607.14943.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. *arXiv preprint arXiv:2412.14803*, 2024. URL http://arxiv.org/abs/2412.14803.

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

Alexander Khazatsky, Karl Pertsch, Suraj Nair, et al. DROID: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024. URL https://arxiv.org/abs/2403.12945.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. doi: 10.48550/arXiv.2406.09246. URL http://arxiv.org/abs/2406.09246.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos Policy: Fine-Tuning Video Models for Visuomotor Control and Planning, 2026. URL https://arxiv.org/abs/2601.16163v1.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026. URL http://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. *arXiv preprint arXiv:2510.21571*, 2025a.

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

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. *arXiv preprint arXiv:2411.04996*, 2024. URL https://arxiv.org/abs/2411.04996.

Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Jingbin Cai, Si Liu, Jianlan Luo, Liliang Chen, Shuicheng Yan, Maoqing Yao, and Guanghui Ren. Genie Envisioner: A Unified World Foundation Platform for Robotic Manipulation, 2025. URL https://arxiv.org/abs/2508.05635v2.

Bokai Lin, Yifu Xu, Xinyu Zhan, Hongjie Fang, Jialin Tian, Fu-Cheng Zhang, Yong-Lu Li, Cewu Lu, and Lixin Yang. ChronoFlow-Policy: Unifying past-current-future interaction flow in visuomotor policy learning. *arXiv preprint arXiv:2606.31493*, 2026. doi: 10.48550/arXiv.2606.31493. URL http://arxiv.org/abs/2606.31493.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth Anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025. URL http://arxiv.org/abs/2511.10647.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023. URL https://arxiv.org/abs/2306.03310.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1B: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024. URL http://arxiv.org/abs/2410.07864.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022. URL http://arxiv.org/abs/2209.03003.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023. URL http://openaccess.thecvf.com/content/ICCV2023/html/Peebles_Scalable_Diffusion_Models_with_Transformers_ICCV_2023_paper.html.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025. URL http://arxiv.org/abs/2501.09747.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025. doi: 10.48550/arXiv.2504.16054. URL http://arxiv.org/abs/2504.16054.

Delin Qu, Haoming Song, Qizhi Chen, Dong Wang, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, Jiayuan Gu, Bin Zhao, and Xuelong Li. SpatialVLA: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025. doi: 10.48550/arXiv.2501.15830. URL http://arxiv.org/abs/2501.15830.

Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 36(6):245:1–245:17, 2017. URL http://mano.is.tue.mpg.de/.

Yichao Shen, Fangyun Wei, Zhiying Du, Yaobo Liang, Yan Lu, Jiaolong Yang, Nanning Zheng, and Baining Guo. VideoVLA: Video generators can be generalizable robot manipulators. *arXiv preprint arXiv:2512.06963*, 2025. URL http://arxiv.org/abs/2512.06963.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Proceedings of the 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 785–799. PMLR, 2023. URL http://proceedings.mlr.press/v205/shridhar23a.html.

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. VLA-JEPA: Enhancing vision-language-action model with latent world model. *arXiv preprint arXiv:2602.10098*, 2026. doi: 10.48550/arXiv.2602.10098. URL http://arxiv.org/abs/2602.10098.

Ruixiang Wang, Qingming Liu, Yueci Deng, Guiliang Liu, Zhen Liu, and Kui Jia. EVA: Aligning video world models with executable robot actions via inverse dynamics rewards. *arXiv preprint arXiv:2603.17808*, 2026a. URL https://arxiv.org/abs/2603.17808.

Xinkai Wang, Chenyi Wang, Yifu Xu, Mingzhe Ye, Fucheng Zhang, Jialin Tian, Xinyu Zhan, Lifeng Zhu, Cewu Lu, and Lixin Yang. LaMP: Learning vision-language-action policy with 3D scene flow as latent motion prior. *arXiv preprint arXiv:2603.25399*, 2026b. doi: 10.48550/arXiv.2603.25399. URL http://arxiv.org/abs/2603.25399.

Kun Wu, Chengkai Hou, Jiaming Liu, et al. RoboMIND: Benchmark on multi-embodiment intelligence normative data for robot manipulation. *arXiv preprint arXiv:2412.13877*, 2024. URL https://arxiv.org/abs/2412.13877.

Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. Florence-2: Advancing a unified representation for a variety of vision tasks. *arXiv preprint arXiv:2311.06242*, 2023. doi: 10.48550/arXiv.2311.06242. URL http://arxiv.org/abs/2311.06242.

Yuxi Xiao, Jianyuan Wang, Nan Xue, Nikita Karaev, Yuri Makarov, Bingyi Kang, Xing Zhu, Hujun Bao, Yujun Shen, and Xiaowei Zhou. SpatialTrackerV2: 3d point tracking made easy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025. URL http://arxiv.org/abs/2507.12462.

Xiaomi Robotics Team, Jun Guo, Piaopiao Jin, Jason Li, Peiyan Li, Yingyan Li, Futeng Liu, Wanli Peng, Optimus Qin, Yifei Su, Nan Sun, Qiao Sun, Runze Suo, Heyun Wang, Yunhong Wang, Rujie Wu, Caoyu Xia, Lina Zhang, Jack Zhao, Guoliang Chen, Wenlong Chen, Xinze He, Bin Li, Qing Li, Zhuorong Li, Heng Qu, Wenxuan Song, Diyun Xiang, Yifan Xie, Peiran Xu, Hangjun Ye, Wen Ye, Han Zhao, and Quanyun Zhou. Xiaomi-Robotics-1: Scaling vision-language-action models with over 100K hours of real-world trajectories. *arXiv preprint arXiv:2607.15330*, 2026. doi: 10.48550/arXiv.2607.15330. URL http://arxiv.org/abs/2607.15330.

Ziyin Xiong, Nikolaos Gkanatsios, Moritz Reuss, and Katerina Fragkiadaki. GeomVLA: Unifying scene, motion, and action in 3D. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2026. URL http://arxiv.org/abs/2609.13812.

Huilin Xu, Jian Ding, Jiakun Xu, Ruixiang Wang, Jun Chen, Jinjie Mai, Yanwei Fu, Bernard Ghanem, Feng Xu, and Mohamed Elhoseiny. Diffusion-based imaginative coordination for bimanual manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11469–11479, 2025a. doi: 10.1109/iccv51701.2025.01067. URL http://openaccess.thecvf.com/content/ICCV2025/html/Xu_Diffusion-Based_Imaginative_Coordination_for_Bimanual_Manipulation_ICCV_2025_paper.html.

Mengda Xu, Zhenjia Xu, Yinghao Xu, Cheng Chi, Gordon Wetzstein, Manuela Veloso, and Shuran Song. Flow as the cross-domain manipulation interface. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2475–2499. PMLR, 2025b. URL http://proceedings.mlr.press/v270/xu25a.html.

Sherry Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023. URL http://arxiv.org/abs/2310.06114.

Yandan Yang, Shuang Zeng, Tong Lin, et al. ABot-M0: VLA Foundation Model for Robotic Manipulation with Action Manifold Learning, 2026. URL https://arxiv.org/abs/2602.11236v1.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi "Jim" Fan, and Joel Jang. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026. doi: 10.48550/arXiv.2602.15922. URL http://arxiv.org/abs/2602.15922.

Chengbo Yuan, Chuan Wen, Tong Zhang, and Yang Gao. General flow as foundation affordance for scalable robot learning. *arXiv preprint arXiv:2401.11439*, 2024. doi: 10.48550/arXiv.2401.11439. URL http://arxiv.org/abs/2401.11439.

Haoqi Yuan, Zhixuan Liang, Anzhe Chen, Ye Wang, Haoyang Li, Pei Lin, Yiyang Huang, Zixing Lei, Tong Zhang, Jiazhao Zhang, Jie Zhang, Jingyang Fan, Gengze Zhou, Qihang Peng, Chenxu Lv, Xiaoyue Chen, An Yang, Fei Huang, Junyang Lin, Dayiheng Liu, Jingren Zhou, Chenfei Wu, and Xiong-Hui Chen. Qwen-RobotManip technical report: Alignment unlocks scale for robotic manipulation foundation models. *arXiv preprint arXiv:2606.17846*, 2026a. doi: 10.48550/arXiv.2606.17846. URL http://arxiv.org/abs/2606.17846.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-WAM: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026b. doi: 10.48550/arXiv.2603.16666. URL http://arxiv.org/abs/2603.16666.

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3D diffusion policy: Generalizable visuomotor policy learning via simple 3D representations. In *Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.067. URL http://www.roboticsproceedings.org/rss20/p067.html.

Qihang Zhang, Lin Li, Luyao Zhang, Shuai Yang, Yiming Luo, Shuaiting Li, Ruilin Wang, Junke Wang, Jiahao Shao, Gangwei Xu, Jiaming Zhou, Yishu Shen, Yudong Jin, Fangyi Xu, Shuailei Ma, Jiaqi Liao, Guanxing Lu, Zifan Shi, Yongkun Wen, Yujie Zhao, Weixuan Tang, Xinyang Wang, Chaojian Li, Jiapeng Zhu, Ka Leong Cheng, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Native Video-Action Pretraining for Generalizable Robot Control, 2026a. URL https://arxiv.org/abs/2607.08639v1.

Shiduo Zhang, Zhe Xu, Peiju Liu, Xiaopeng Yu, Yuan Li, Qinghui Gao, Zhaoye Fei, Zhangyue Yin, Zuxuan Wu, Yu-Gang Jiang, and Xipeng Qiu. VLABench: A large-scale benchmark for language-conditioned robotics manipulation with long-horizon reasoning tasks. *arXiv preprint arXiv:2412.18194*, 2024. URL https://arxiv.org/abs/2412.18194.

Zhanguang Zhang, Zhiyuan Li, Behnam Rahmati, Rui Heng Yang, Yintao Ma, Amir Rasouli, Sajjad Pakdamansavoji, Yangzheng Wu, Lingfeng Zhang, Tongtong Cao, Feng Wen, Xinyu Wang, Xingyue Quan, and Yingxue Zhang. Do world action models generalize better than VLAs? a robustness study. *arXiv preprint arXiv:2603.22078*, 2026b. doi: 10.48550/arXiv.2603.22078. URL http://arxiv.org/abs/2603.22078.

Haoyu Zhao, Xingyue Zhao, Siteng Huang, Xin Li, Deli Zhao, and Zhongyu Li. RynnWorld-4D: 4D embodied world models for robotic manipulation. *arXiv preprint arXiv:2607.06559*, 2026. URL http://arxiv.org/abs/2607.06559.

Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3D-VLA: A 3D vision-language-action generative world model. *arXiv preprint arXiv:2403.09631*, 2024. URL http://arxiv.org/abs/2403.09631.

Haoyu Zhen, Qiao Sun, Hongxin Zhang, Junyan Li, Siyuan Zhou, Yilun Du, and Chuang Gan. TesserAct: Learning 4D embodied world models. *arXiv preprint arXiv:2504.20995*, 2025. URL http://arxiv.org/abs/2504.20995.

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. *arXiv preprint arXiv:2510.10274*, 2025. doi: 10.48550/arXiv.2510.10274. URL http://arxiv.org/abs/2510.10274.

Hongyan Zhi, Peihao Chen, Siyuan Zhou, Yubo Dong, Quanxi Wu, Lei Han, and Mingkui Tan. 3DFlowAction: Learning cross-embodiment manipulation from 3D flow world model. *arXiv preprint arXiv:2506.06199*, 2025. doi: 10.48550/arXiv.2506.06199. URL http://arxiv.org/abs/2506.06199.

Huayi Zhou, Wei Gao, Dekun Lu, Ruiji Liu, Zhanqi Zhang, Ziyang Zhang, Jian Chen, Wenlve Zhou, Sheng Xu, Shumin Li, Kangyi Guo, Shichen Xu, Zixin Huang, Yongyi Su, and Kui Jia. Dexterity-BEV: Aligning 3D world and actions for generalizable robot policies learning. *arXiv preprint arXiv:2606.02274*, 2026a. URL https://arxiv.org/abs/2606.02274v2.

Pengfei Zhou, Shengcong Chen, Di Chen, Jiaxu Wang, Rongjun Jin, Bingwen Zhu, Yike Pan, Songen Gu, Kuanning Wang, Shufeng Nan, et al. τ_0-WM: A unified video-action world model for robotic manipulation. *arXiv preprint arXiv:2606.01027*, 2026b. URL http://arxiv.org/abs/2606.01027.

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. *arXiv preprint arXiv:2504.02792*, 2025. URL http://arxiv.org/abs/2504.02792.

A Implementation and Training Details

A.1 Data Mixture and Temporal Sampling

Heterogeneous data mixture. We pretrain on a mixture of nine data sources: RoboTwin (Chen et al., 2025), LIBERO (Liu et al., 2023), VLABench (Zhang et al., 2024), DROID (Khazatsky et al., 2024), AgiBot (AgiBot-World-Contributors et al., 2025), RoboMIND (Wu et al., 2024), RH20T (Fang et al., 2023), VITRA (Li et al., 2025a), and our self-collected W1 data. Fig. 6 shows representative observations and aligned geometry.

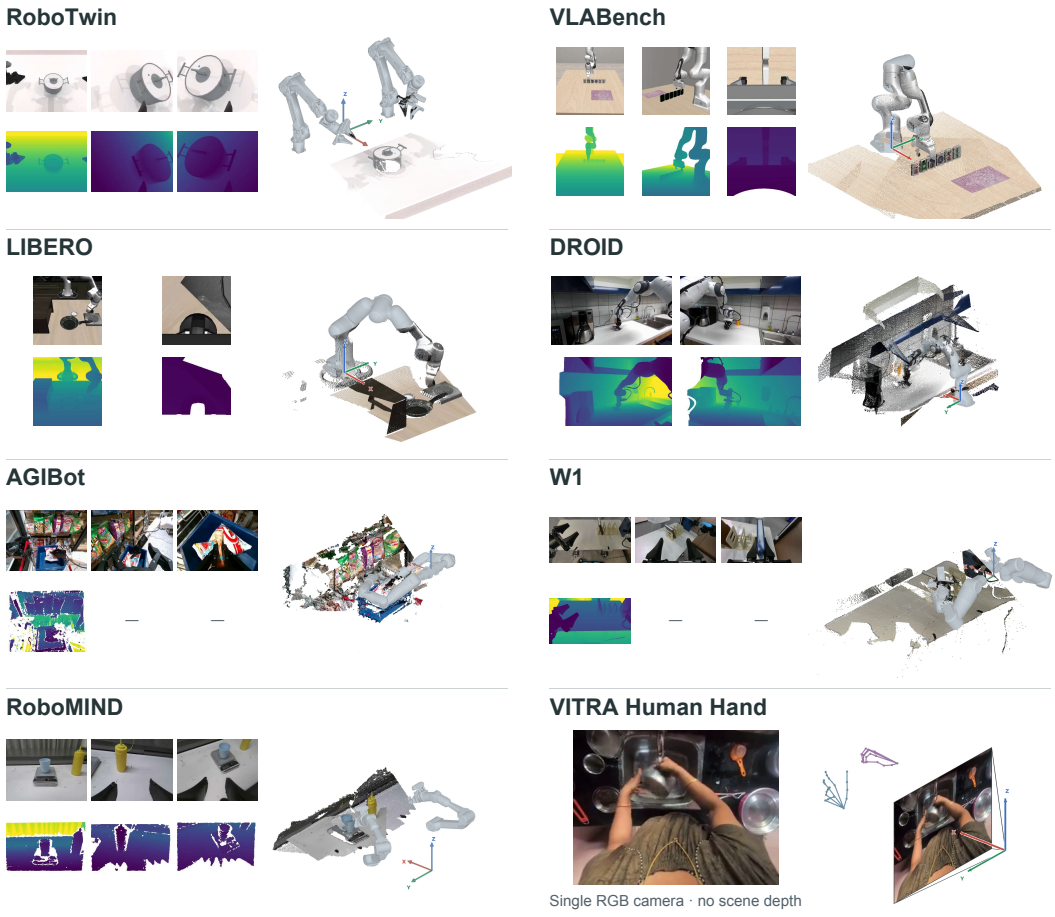

Figure 6: **Representative observations and aligned geometry from pretraining data.** Robot panels show RGB, depth where available, and scene points with robot overlays. VITRA shows egocentric RGB and hand landmarks.

Dataset-specific coordinate conventions. Robot data use a base or shared embodiment frame. Where specified by the bimanual adapter, the origin is the arm-base midpoint, with orientation inherited from the shared root or torso. W1 training coordinates are centered at the median end-effector trajectory midpoint. For VITRA, source camera poses transform hand positions into the current anchor's camera frame, held fixed throughout the window, with $(x_M, y_M, z_M) = (z_C, -x_C, -y_C)$.

Temporal sampling. Tab. 5 reports the trajectory sampling settings of pretraining. The effective sampling rate f accounts for temporal subsampling: H future intervals span H/f seconds and yield $H + 1$ coordinates including the current anchor. We observe slower robot motion in AgiBot and RoboMIND and therefore subsample their 45-Hz trajectories with stride 3 (15 Hz). This extends the physical time covered by each action chunk, bringing its spatial motion extent closer to that of

chunks from the other datasets. VITRA uses 6 Hz for Ego4D, 8 Hz for EgoExo4D and Something-Something V2, and 10 Hz for EPIC-KITCHENS. H_{hist} specifies the data history window.

Table 5: Pretraining trajectory sampling configuration. H and H_{hist} count future and historical intervals; duration is H/f in seconds. $H+1$ includes the current anchor.

Dataset	Source	f (Hz)	H	H_{hist}	Duration	$H+1$
RoboTwin	Simulation	15	20	20	1.33	21
LIBERO	Simulation	20	20	20	1.00	21
VLABench	Simulation	10	15	15	1.50	16
DROID	Real robot	15	20	20	1.33	21
AgiBot	Real robot	15	20	20	1.33	21
RoboMIND	Real robot	15	20	20	1.33	21
RH20T	Real robot	20	20	20	1.00	21
W1-data	Real robot	15	20	20	1.33	21
VITRA	Human hand	6/8/10	10	10	$10/f$	11

A.2 Physical-Time Trajectory Tokenizer

The encoder and decoder are defined in Eq. equation 8. The temporal basis takes timestamps in seconds on the shared scale $T_{\max} = 3\,\text{s}$:

$$[\boldsymbol{\phi}_{\text{time}}(\tau)]_j = \operatorname{sgn}(\tau)\,\sin\left[\frac{\pi}{2}\left|\frac{\tau}{T_{\max}}\right|^{j/d_z}\right], \qquad j = 0, \ldots, d_z - 1, \tag{10}$$

with latent dimension $d_z = 256$. Future trajectories use positive offsets; history uses negative offsets and displacements from past positions toward the current anchor. An input MLP produces $\mathbf{h}_{n,k}$ from $\Delta\mathbf{x}_{n,k}$, and separate temporal MLPs produce the encoder and decoder FiLM parameters.

Reconstruction training. We minimize squared Euclidean error over valid point–time pairs, averaging the future and historical action losses and adding available scene reconstruction loss. The trained tokenizer is frozen, and its latents are standardized with fixed channelwise training statistics before world-action model training.

Spatial anchoring and query sets. The spatial anchor is separate from the displacement latent. Pointwise Fourier features of $\mathbf{x}_{t,n}$ are projected to the point dynamics expert's hidden dimension and added to the trajectory token throughout denoising. Shared token processing and order-independent attention masks make prediction permutation equivariant within each query type: reordering query points and their noise tokens together reorders the outputs. Variable query counts support sparse object-level prediction or denser scene coverage.

Held-out trajectory reconstruction. We evaluate the frozen tokenizer on 600 held-out trajectory windows per source and trajectory type, none used for tokenizer training. Each point trajectory is encoded into one token and decoded at its original physical timestamps. Tab. 6 reports ADE, the mean Euclidean error across future timesteps, and FDE, the error at the final timestep, averaged over point trajectories. The millimeter-scale errors indicate that the tokenizer preserves motion information across robot action flow, human-hand action flow, and scene flow in these samples.

Table 6: Held-out trajectory reconstruction errors of the frozen tokenizer (mm). Dashes indicate unavailable scene-flow ground truth.

Data source	Action flow		Scene flow	
	ADE ↓	FDE ↓	ADE ↓	FDE ↓
RoboTwin	1.96	4.62	1.01	2.32
LIBERO	2.54	6.05	0.78	1.56
VLABench	3.38	6.25	0.94	1.89
DROID	3.17	7.85	–	–
AgiBot	2.26	4.62	–	–
RoboMIND	3.48	7.60	–	–
RH20T	3.16	4.91	–	–
W1	1.70	3.33	–	–
VITRA (human hand)	4.68	8.18	–	–

A.3 Architecture and Optimization

Tab. 7 summarizes the model configuration.

Table 7: Model architecture. Parameter counts include the frozen tokenizer.

Component	Configuration	Params. (M)
VLM	Florence-2-large	570.7
Point dynamics expert	28 layers; width 1,280; 20 heads	826.3
Action expert	28 layers; width 640; 10 heads	256.0
Trajectory tokenizer	Time-conditioned MLPs; latent dim. 256	1.9
Embeddings and projections	Spatial and modality interfaces	39.7
Total		1,694.5

Token structure and attention. The mask is specified in Sec. 3.5. The action expert has the same depth as the dynamics expert and a smaller hidden dimension, with one noisy command token per future timestep. Physical execution time τ_k enters command tokens through FiLM, separately from the denoising time.

Flow matching and inference. The interpolation and the loss weights are given in Sec. 3.5. At inference, action flow and commands use the same Euler schedule for 10 steps. Scene denoising starts from independent noise.

Training configuration. We pretrain for 200K steps with a total batch size of 1,024 using AdamW. The learning rates are 10^{-4} for the point dynamics and action experts and 10^{-5} for the VLM, with weight decay 10^{-8}. We use constant learning rates after 1,000 steps of linear warmup. Training uses 64 BW1000_H DCUs (64 GiB each) with FSDP.

Queries and objectives. Robot grippers use 20 query points sampled from URDF-defined gripper geometry per arm; human hands use 21 MANO-derived keypoints per observed hand. Scene points are voxelized and capped at 12,000 before random subsampling. The mixed-dataset run retains 1% (approximately 120 scene queries at the cap), with 5-mm voxels for the simulation datasets. RGB inputs are resized to 224×224. The flow-matching weights are set in Sec. 3.5; unavailable supervision is masked. This run uses a frustum-routing substitution probability of 0.1. Action position coordinates are scaled by 10; rotation and gripper coordinates retain unit scale. Inference uses 10 Euler steps for each active expert.

A.4 Inference Efficiency

We measure inference latency on a single NVIDIA GeForce RTX 5090 D. As shown in Tab. 8, UNITAS directly produces executable actions without generating action or scene flows in 55.77 ms per chunk, providing a $2.20\times$ speedup over jointly generating actions and action flow. With only

1.7B parameters, this mode is $2.81\times$ faster than Fast-WAM without video generation and substantially faster than the joint video–action baselines Fast-WAM (Yuan et al., 2026b) and Motus (Bi et al., 2025), supporting low-latency robot control. In world-model mode, the number of scene queries can be adjusted at inference to match the desired spatial sampling density and compute budget. Reducing scene queries from 3,000 to 1,024 lowers joint prediction latency from 561.60 ms to 305.49 ms, allowing a flexible trade-off between scene coverage and inference cost.

Table 8: **Inference efficiency on an RTX 5090 D.** Latency is measured per prediction chunk; N_s denotes the number of scene queries.

Method / output mode	Params (B)	N_s	Latency (ms)
Fast-WAM (w/o video)	6.0	–	156.81
Fast-WAM (joint)	6.0	–	579.79
Motus	8.0	–	2,414.90
UNITAS: actions only	1.7	–	55.77
UNITAS: actions + action flow	1.7	–	122.51
UNITAS: actions + both flows	1.7	1,024	305.49
UNITAS: actions + both flows	1.7	3,000	561.60

B SPATIAL GROUNDING WITHOUT DEPTH INPUT

Candidate depths, the router, and $\mathcal{L}_{\mathrm{PE}}$ are defined in Sec. 3.3, Eqs. equation 6 and equation 7. During training, routed features occasionally replace depth-derived features on valid regions, with the substitution rate reported in Appendix A.3. This section evaluates the router when depth is removed at test time.

B.1 EVALUATION WITHOUT DEPTH INPUT

We evaluate policy execution on LIBERO-Plus at 100K training steps, without mixed-dataset pretraining or scene-flow supervision. Tab. 9 compares routed PE without depth input against ground-truth depth PE using the same checkpoint, and includes the no-3D-PE baseline from Tab. 3. Camera calibration and robot-state inputs remain available in all settings.

Table 9: **Spatial grounding without depth input on LIBERO-Plus.** Success rates (%, ↑) over 10,030 episodes. Routed PE and ground-truth depth PE use the same checkpoint.

Visual spatial encoding	Camera	Robot	Language	Light	Background	Noise	Layout	Overall
No visual 3D PE	66.73	79.29	75.21	94.57	92.19	84.13	84.46	81.35
Routed PE (no depth)	83.74	79.61	74.30	91.68	84.20	93.38	85.57	84.43
Ground-truth depth PE	84.37	79.68	74.24	91.94	85.13	93.88	85.18	84.68

Results. Without depth input, routed PE achieves 84.43% overall success, closely matching the 84.68% obtained with ground-truth depth PE. The gap of only 0.25 percentage points supports the effectiveness of distilling depth-derived spatial information into RGB-conditioned positional features, preserving policy performance when depth is unavailable at inference. Compared with removing visual 3D PE, routed PE improves overall success by 3.08 points, with the largest gains under camera-viewpoint perturbations and sensor noise.

C DETAILED LIBERO-PLUS RESULTS

For both training settings, we train UNITAS on standard LIBERO demonstrations and use the same checkpoints as in the LIBERO evaluation to evaluate directly on LIBERO-Plus. No additional training or adaptation is performed on LIBERO-Plus; this is an out-of-distribution evaluation.

Tab. 10 breaks down success rates across all seven perturbation types in LIBERO-Plus (Fei et al., 2025); Tab. 1 reports overall success. The comparison additionally includes OpenVLA (Kim et al., 2024) and ABot-M0.5 (Chen et al., 2026b).

Table 10: LIBERO-Plus success rates (%, ↑) across seven perturbation types. Robot denotes initial-state perturbations.

Method	Params (B)	Camera	Robot	Language	Light	Background	Noise	Layout	Overall
X-VLA	1	23.4	89.7	75.7	88.2	96.0	62.7	71.8	71.4
π_0	3	13.8	6.0	58.8	85.0	81.4	79.0	68.9	53.6
$\pi_{0.5}$	3	75.4	77.5	85.6	96.9	94.6	89.7	85.7	85.7
OpenVLA	7	0.8	3.5	23.0	8.1	34.8	15.2	28.5	15.6
Cosmos Policy	2	75.8	63.3	81.7	96.5	88.9	**92.7**	82.2	82.2
LingBot-VA	5.3	40.9	83.0	86.4	82.3	53.1	64.4	76.2	69.5
Fast-WAM	6	16.4	44.5	68.9	78.2	53.7	37.7	60.7	51.5
ABot-M0.5	6	70.5	87.4	**88.6**	94.0	89.7	75.5	85.2	83.4
LaMP	5	64.5	69.6	88.2	95.3	**97.4**	76.9	73.8	79.3
UNITAS (w/o Pretrain)	1.7	81.9	84.8	77.6	96.7	94.4	89.8	85.1	86.5
UNITAS	1.7	**82.1**	**94.2**	85.0	**98.1**	96.3	84.5	**87.7**	**89.1**

D DETAILED VLABENCH RESULTS

Tab. 11 reports five evaluation tracks: in-distribution, cross-category, commonsense, semantic instruction, and unseen texture (Tracks 1, 2, 3, 4, and 6). The comparison additionally includes π_0-FAST (Pertsch et al., 2025).

Table 11: Detailed VLABench results (%). SR, PS, and IS denote success rate, progress score, and intention score, respectively (all ↑).

Method	In-distribution			Cross-category			Commonsense		
	SR	PS	IS	SR	PS	IS	SR	PS	IS
π_0	47.0	62.7	67.8	21.2	33.6	44.0	29.1	43.0	54.9
π_0-FAST	56.2	66.8	72.4	31.0	38.2	47.8	38.0	48.6	56.8
X-VLA	40.0	66.5	**88.0**	11.0	24.0	42.0	33.0	51.7	66.0
$\pi_{0.5}$	65.4	77.8	80.4	**38.2**	**49.7**	**52.0**	43.9	57.3	60.0
UNITAS (w/o Pretrain)	62.0	76.5	80.2	25.4	37.2	45.7	49.0	63.4	**72.8**
UNITAS	**75.1**	**85.6**	82.3	28.9	39.1	44.4	**53.8**	**65.9**	68.5

Method	Semantic instruction			Unseen texture			Avg.		
	SR	PS	IS	SR	PS	IS	SR	PS	IS
π_0	17.3	38.7	58.0	32.2	42.5	50.6	29.4	44.1	55.0
π_0-FAST	35.0	45.0	59.4	39.0	49.0	56.8	39.8	49.5	58.6
X-VLA	39.0	62.5	**82.0**	24.0	51.2	73.0	29.4	51.2	70.2
$\pi_{0.5}$	48.2	64.2	67.0	44.9	62.3	65.0	48.1	62.3	64.9
UNITAS (w/o Pretrain)	62.4	75.8	79.8	52.6	**68.5**	**77.6**	50.3	64.3	**71.2**
UNITAS	**67.7**	**79.9**	78.7	**55.4**	67.4	72.7	**56.2**	**67.6**	69.3

E DETAILED ROBOTWIN RESULTS

Tab. 12 gives per-task results for both training settings in Tab. 1. Each setting contains 50 tasks with 100 evaluation episodes per task; averages weight all tasks equally.

Table 12: Detailed RoboTwin 2.0 success rates (%, ↑). Clean and Rand. denote `demo_clean` and `demo_randomized`. All methods are evaluated over 100 episodes per task in each setting.

Task	$\pi_{0.5}$		Motus		Fast-WAM		UNITAS (w/o Pretrain)		UNITAS	
	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.
Adjust Bottle	**100**	99	89	93	**100**	**100**	**100**	98	**100**	**100**
Beat Block Hammer	96	93	95	88	99	97	97	92	**100**	**98**
Blocks Ranking RGB	92	85	99	97	**100**	**100**	**100**	99	97	94
Blocks Ranking Size	49	26	75	63	**94**	**98**	91	83	78	82
Click Alarmclock	98	89	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
Click Bell	99	66	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
Dump Bin Bigbin	92	97	95	91	97	96	97	**98**	**98**	97
Grab Roller	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
Handover Block	66	57	86	73	95	81	88	74	**96**	**84**
Handover Mic	98	97	78	63	99	**100**	**100**	99	**100**	97
Hanging Mug	18	17	38	38	58	**62**	48	57	**61**	41
Lift Pot	96	85	96	99	**100**	**100**	98	99	99	99
Move Can Pot	51	55	34	74	90	88	**97**	**100**	94	96
Move Pillbottle Pad	84	61	93	96	**100**	**99**	99	**99**	93	98
Move Playingcard Away	96	84	**100**	96	**100**	**100**	98	**100**	**100**	**100**
Move Stapler Pad	56	42	**83**	**85**	77	64	54	50	79	78
Open Laptop	90	96	95	91	98	**100**	98	99	**100**	**100**
Open Microwave	34	77	95	**91**	62	45	**96**	86	74	80
Pick Diverse Bottles	81	71	**90**	**91**	80	85	84	81	86	88
Pick Dual Bottles	93	63	96	90	**100**	96	96	**98**	**100**	**98**
Place A2B Left	87	82	88	79	95	93	90	95	**100**	**98**
Place A2B Right	87	84	91	87	**93**	**99**	92	95	**93**	98
Place Bread Basket	77	64	91	94	91	93	**95**	**98**	**95**	91
Place Bread Skillet	85	66	86	83	**90**	93	87	**95**	89	86
Place Burger Fries	94	87	98	98	96	**99**	**100**	**99**	96	98
Place Can Basket	62	62	**81**	76	71	69	72	74	77	**92**
Place Cans Plasticbox	94	84	98	94	**99**	96	**99**	97	**99**	**99**
Place Container Plate	99	95	98	99	96	**100**	99	95	**100**	99
Place Dual Shoes	75	75	93	87	94	88	92	91	**99**	**96**
Place Empty Cup	**100**	99	99	98	**100**	**100**	**100**	**100**	**100**	**100**
Place Fan	87	85	91	87	**96**	**96**	89	92	90	94
Place Mouse Pad	60	39	66	68	83	89	89	**92**	**91**	87
Place Object Basket	80	76	81	87	89	88	79	**94**	**94**	91
Place Object Scale	86	80	88	85	90	**97**	**91**	94	**91**	91
Place Object Stand	91	85	**98**	**97**	90	94	92	88	97	**97**
Place Phone Stand	81	81	87	86	**97**	**99**	85	89	89	95
Place Shoe	92	93	99	97	96	**99**	97	97	**100**	98
Press Stapler	87	83	93	**98**	90	97	**98**	**98**	**98**	96
Put Bottles Dustbin	84	79	81	79	**95**	90	87	90	85	**98**
Put Object Cabinet	80	79	88	71	**94**	89	88	84	77	**92**
Rotate QR Code	89	87	89	73	**93**	89	91	91	92	**92**
Scan Object	72	65	67	66	89	92	87	92	**97**	**96**
Shake Bottle	99	97	**100**	97	**100**	**100**	**100**	97	**100**	99
Shake Bottle Horizontally	99	99	**100**	98	**100**	**100**	**100**	94	**100**	**100**
Stack Blocks Three	91	76	91	95	95	**97**	93	84	**98**	94
Stack Blocks Two	97	**100**	**100**	98	**100**	**100**	99	98	**100**	96
Stack Bowls Three	77	71	79	**87**	80	81	**84**	**87**	**84**	85
Stack Bowls Two	95	96	**98**	98	92	98	91	99	95	**100**
Stamp Seal	79	55	**93**	92	90	**94**	79	63	83	92
Turn Switch	62	54	**84**	**78**	61	59	68	63	83	72
Average	82.74	76.76	88.66	87.02	91.88	91.78	91.08	90.74	**92.94**	**93.04**

F WORLD MODEL EVALUATION DETAILS

F.1 ACTION-CONDITIONED SCENE PREDICTION

Evaluation protocol. Tab. 4 evaluates 550 test clips from 50 RoboTwin tasks, with one Clean and ten Randomized episodes per task. Each episode contributes a fixed window near its midpoint, containing the current frame and 20 future recorded frames. Initial scene points are sampled with a 1.5 cm voxel grid. All methods share initial queries, point identities, ground-truth trajectories, and robot-motion conditions. For video methods, queries are projected into the initial image us-

ing camera calibration, accounting for resizing and cropping; only in-frame projections are tracked by SpatialTrackerV2. Ctrl-World uses Depth Anything 3 depth, whereas X-WAM uses its native depth predictions. Depth scale for Ctrl-World and inverse-depth affine calibration for X-WAM are estimated from initial observed depth and fixed throughout the video. Tracked trajectories are transformed into world coordinates, and their predicted displacements are anchored at the shared initial queries. No future ground-truth depth or trajectories are used for alignment.

Scoring. In our evaluation, we label a point as moving when its maximum ground-truth displacement from its initial position over the 20 future frames exceeds 1 cm, and as static otherwise. These labels are fixed per clip and shared across methods. We score future frames $4, 8, 12, 16, 20$: ADE averages Euclidean errors over valid point–frame pairs in each group, Moving FDE measures moving-point error at frame 20, and Moving Acc@1cm is the fraction of valid moving-point–frame pairs with prediction error at most 1 cm. Metrics are first computed per clip, averaged within each task–condition group, and then averaged equally across groups. Clips without valid moving points are excluded only from the corresponding moving metrics.

F.2 Closed-Loop Action–Scene Consistency

We evaluate three RoboTwin tasks, Adjust Bottle, Beat Block Hammer, and Blocks Ranking RGB, under Clean and Randomized conditions, yielding six settings. Each method is evaluated over 10 trials per setting, for 60 trials in total. We compare with X-WAM (Guo et al., 2026), Fast-WAM (Yuan et al., 2026b), LingBot-VA (Li et al., 2026), and Motus (Bi et al., 2025). Each generated action chunk is executed in the simulator, and the resulting observation conditions the next prediction. Models retain their native prediction horizons and are scored at completed action boundaries until termination. Fast-WAM executes all 32 actions rather than replanning after 24.

For UNITAS, the point dynamics expert generates an action flow, while the action expert separately generates an executable command chunk from the same observation and instruction. The dynamics expert then predicts scene-flow trajectories conditioned on its generated action flow. We compare these predictions with the ground-truth scene trajectories resulting from executing the action expert's command chunk in the simulator.

Trajectory extraction and scoring. Ground-truth query points follow their simulator objects or links. SpatialTrackerV2 lifts video predictions using X-WAM's native depth or Depth Anything 3 estimates for RGB-only models, with initial-frame depth calibration and query anchoring. We apply the 1 cm displacement threshold to label moving points within each scored chunk. Metrics are averaged within rounds, then within episodes, across trials within each setting, and finally equally across the six settings. Rounds without valid moving points are excluded only from moving-point metrics; all episodes are retained regardless of task outcome.

Table 13: **Closed-loop action–scene consistency on RoboTwin.** Predictions are compared with the scene evolution produced by executing the same model's generated actions. Errors are in cm; Moving Acc@1cm is in %.

Method	Moving ADE ↓	Static ADE ↓	Moving FDE ↓	Moving Acc@1cm ↑
X-WAM + STv2	<u>2.579</u>	<u>0.166</u>	<u>4.056</u>	35.55
Fast-WAM + STv2	3.554	0.586	5.133	30.37
LingBot-VA + STv2	2.603	0.810	4.716	33.64
Motus + STv2	3.216	0.318	6.778	<u>35.66</u>
UNITAS	**1.211**	**0.054**	**2.020**	**64.26**

These scores measure action–scene consistency along each model's own rollout and include video-to-3D reconstruction error for video-based methods. For UNITAS, agreement between predicted and observed scene changes also provides indirect evidence that its generated action flow and executable commands are consistent in their scene effects.

G Additional Action-Conditioned Scene Predictions

G.1 Scene Prediction in Simulation

Figs. 7–10 extend Fig. 5 with 19 training clips, showing RGB, initial points, predicted flows, and ground-truth flows. Purple denotes conditioning gripper motion. These examples illustrate learned geometry rather than held-out accuracy. For denser visualization, we relax part-boundary and depth-jump filters while retaining valid-depth and non-robot checks. Every displayed scene point is directly predicted; initial RGB colors are retained and unobserved surfaces are not synthesized.

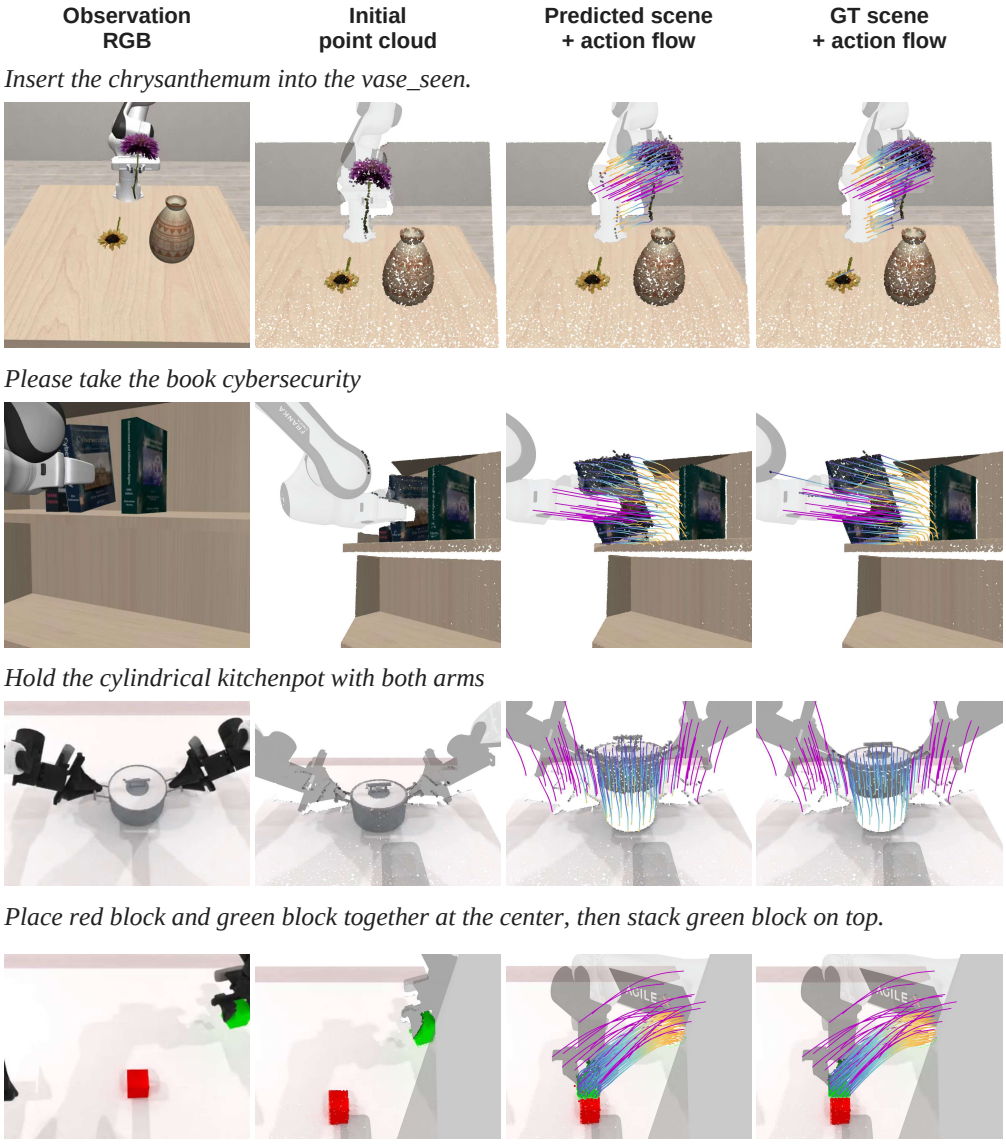

Figure 7: Additional scene-flow visualizations (1/4): insertion, extraction, bimanual lifting, and block stacking.

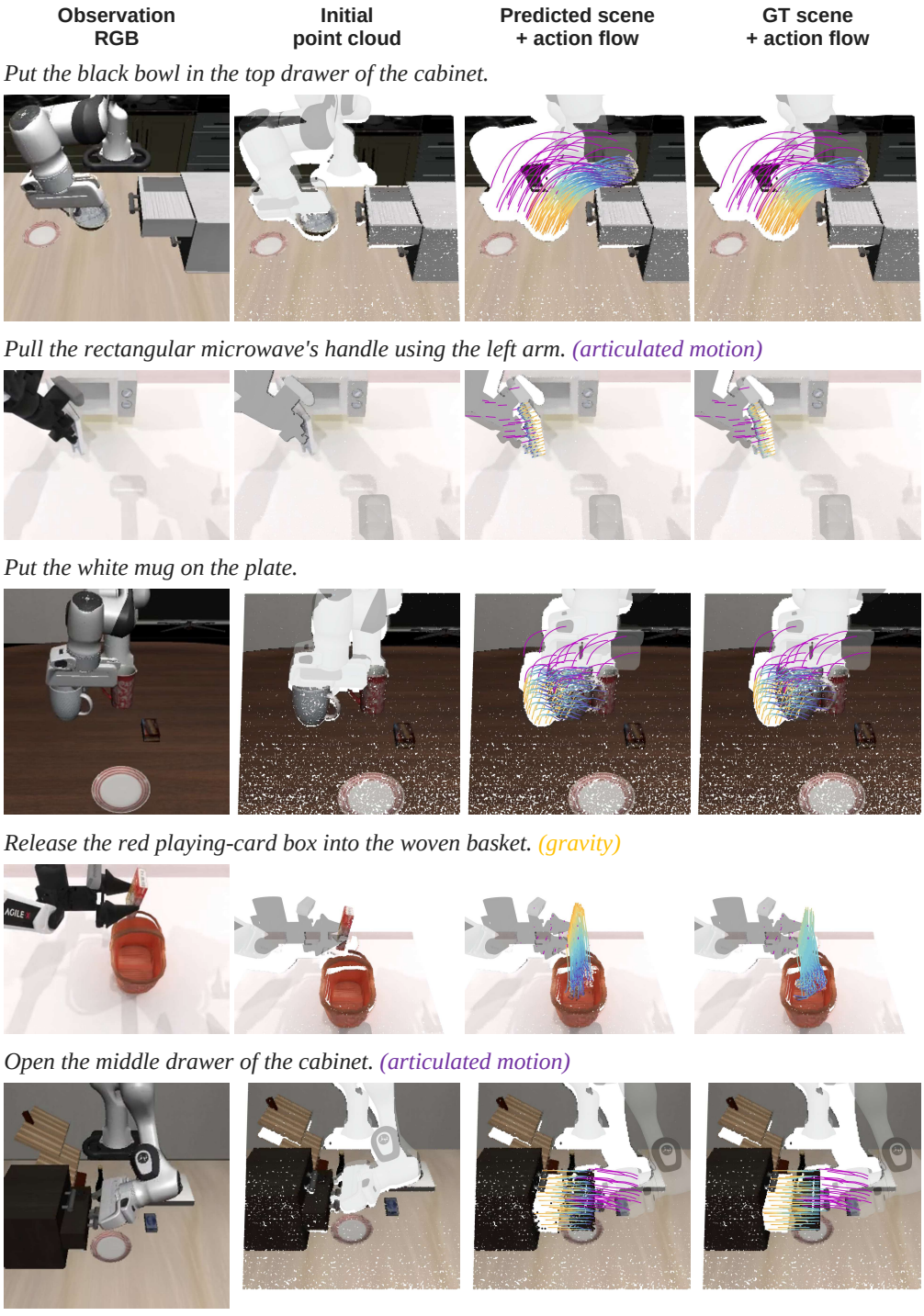

Figure 8: Additional scene-flow visualizations (2/4): bowl and mug placement, articulated motion, and gravity-driven motion after object release.



Figure 9: Additional scene-flow visualizations (3/4): book, bottle, block, and bowl manipulation across LIBERO and RoboTwin.



Figure 10: Additional scene-flow visualizations (4/4): object placement, drawer opening and closing, and microwave opening.

G.2 TRANSFER TO REAL-WORLD OBSERVATIONS

Fig. 11 shows predictions on DROID and RH20T conditioned on ground-truth gripper trajectories. Despite receiving scene-flow supervision only from simulation, UNITAS captures action-conditioned scene changes in real environments outside this supervision distribution. These qualitative results suggest that the learned interaction dynamics can transfer from simulation to real scenes.



Figure 11: **Zero-shot scene-flow predictions on real-world observations.** The first four rows are from DROID and the last is from RH20T. Columns show observed RGB, initial points, and predicted midpoint and endpoint point clouds with scene trajectories (orange to purple) and ground-truth gripper trajectories used for conditioning (magenta). No scene-trajectory ground truth is shown.