arXiv:2611.07229v1 [cs.RO] 4 Nov 2026

Spatial-UMI-Dream: Scaling World Action Models with Metric Geometry for UMI Manipulation

Weiduo Yuan[1,*] Ze Yuan[1,*] Sihan Liu[2,*] Jionghao Wang[3] Liang Heng[1] Zhenyu Zhao[1] Yue Wang[1]
[1]USC Physical Superintelligence Lab [2]CMU [3]Texas A&M University
[*]Equal contribution

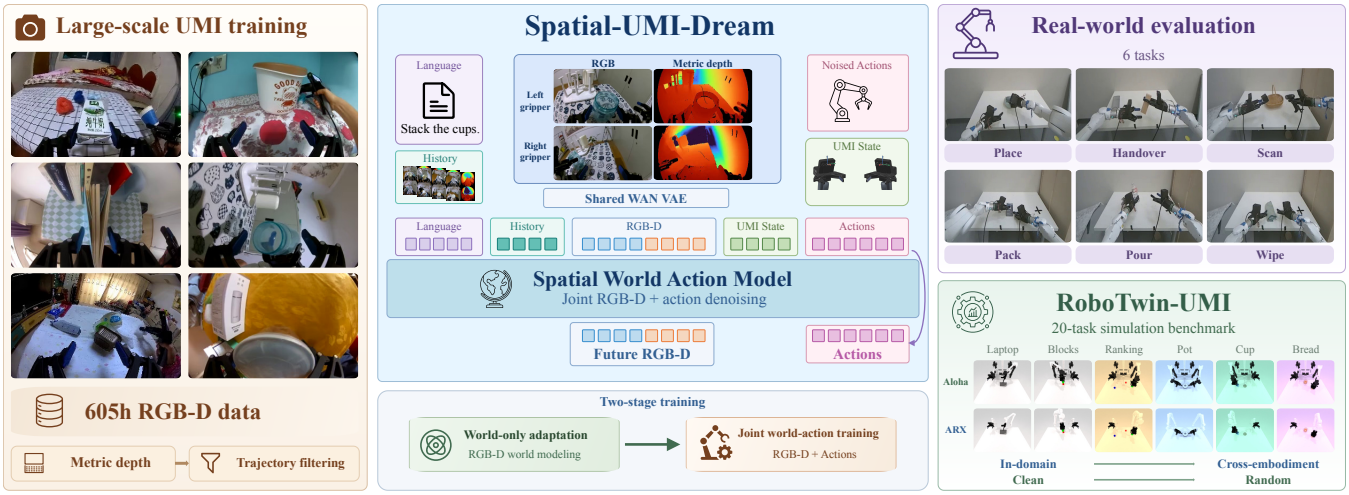


Fig. 1: **Spatial understanding and cross-embodiment transfer for UMI policies.** We introduce *Spatial-UMI-Dream*, a 5B world-action model that learns jointly from RGB, metric depth, and actions through 605 hours of UMI pretraining. Across six real-world bimanual tasks and the twenty-task *RoboTwin-UMI* benchmark, it achieves the best in-domain performance and generalizes best across robot embodiments and environments.

***Abstract*— The Universal Manipulation Interface (UMI) enables scalable demonstration collection independently of a specific robot. However, its moving, close-range gripper cameras entangle camera and object motion without a stable external view, leaving the metric geometry needed for precise manipulation implicit in RGB. We introduce *Spatial-UMI-Dream*, a 5B spatial World Action Model (WAM) pretrained on 605 hours of RGB-D UMI data through two-stage training—world-only adaptation followed by joint world-action learning—to jointly predict future RGB, fisheye-aligned metric depth, and actions. Stereo-derived depth makes scene scale and spatial relationships explicit within a unified dual-gripper RGB-D representation. Furthermore, evaluating UMI policies requires testing whether UMI's robot-independent demonstration collection translates into policy generalization across robot embodiments, alongside robustness to environmental changes. We introduce *RoboTwin-UMI*, a twenty-task simulation benchmark built on RoboTwin with UMI wrist-mounted fisheye observations, enabling controlled, repeatable evaluation of UMI policies under environment and embodiment changes separately and jointly without additional fine-tuning. On *RoboTwin-UMI*, *Spatial-UMI-Dream* achieves 79.10% in-domain success and 72.15% under combined embodiment and environment shifts, exceeding the strongest evaluated baseline in each setting by 4.65 and 10.30 percentage points, respectively. It also exhibits the smallest success-rate drop under embodiment transfer among the evaluated policies. In a matched pretraining ablation, RGB-only pretraining leads to a 5.75-percentage-point decrease in in-domain success compared with RGB-D pretraining. Across six real-world tasks, *Spatial-UMI-Dream* achieves 85.0% average success, exceeding the strongest baseline by 24.2 percentage points. Our project page is available at `https://weiduoyuan.com/spatial-umi-dream/`.**

I. INTRODUCTION

The Universal Manipulation Interface (UMI) [1] enables scalable collection of manipulation demonstrations without requiring a robot at the demonstration site. Its portable handheld grippers record close-range video and embodiment-agnostic end-effector trajectories during in-the-wild human manipulation. These demonstrations provide interaction data for learning policies that can execute manipulation skills across tasks, environments, and robot embodiments.

Vision-Language-Action models (VLAs) and World Action Models (WAMs) offer two approaches to learning manipulation policies. VLAs [2], [3], [4], [5] build on pretrained vision-language models to predict robot actions from visual observations and language instructions. WAMs [6], [7], [8] instead build on pretrained video generation models to jointly predict future visual observations and actions. This joint prediction connects actions with the resulting scene changes. Future video supplies supervision across space and time, while pretrained video diffusion models provide priors over motion and scene dynamics.

Although WAMs have shown promising results with RGB observations, applying them to UMI data presents two challenges. First, RGB-only models leave the metric

geometry needed for manipulation implicit in appearance and image-space motion. UMI has no fixed third-person view, and its gripper-mounted fisheye cameras move with the grippers during close-range manipulation. Changes in the images therefore reflect camera motion, object motion, and scene geometry. In bimanual manipulation, both views move independently, while the model must coordinate two grippers in a shared three-dimensional workspace. The model must therefore infer scene scale and spatial relationships from RGB alone. Second, in-domain task success alone is insufficient to evaluate UMI policies. Lighting and scene appearance can vary substantially in wrist-camera observations, and demonstrations collected in a limited set of environments may not cover these variations. Moreover, because UMI decouples demonstration collection from a specific robot, cross-embodiment transfer is central to evaluating UMI policies. Evaluation must therefore test both robustness to environmental changes and the ability to transfer learned skills across robot embodiments.

To address these challenges, we introduce ***Spatial-UMI-Dream***, a spatial WAM that incorporates metric depth, and ***RoboTwin-UMI***, a benchmark for evaluating UMI policies across environments and robot embodiments.

Spatial-UMI-Dream is a 5B spatial WAM pretrained on 605 hours of RGB-D UMI data to jointly predict future RGB, metric depth, and robot actions. Training proceeds in two stages: world-only adaptation to UMI RGB-D observations, followed by joint world-action learning. We recover metric depth from calibrated stereo observations and align it with the raw fisheye RGB images. Metric depth makes scene scale and the spatial layout around each gripper explicit, reducing the need to infer this information from RGB alone. We arrange both grippers' aligned RGB-D observations in a unified video canvas, incorporating geometry into world-action modeling while retaining the pretrained video tokenizer and WAM backbone.

RoboTwin-UMI is a twenty-task benchmark built on RoboTwin [9] that reproduces UMI's wrist-mounted fisheye observations. Simulation enables controlled, repeatable evaluation of the same tasks across different environments and robot embodiments. This supports testing both environmental robustness and cross-embodiment transfer, which is particularly important because UMI demonstrations are collected independently of a specific robot. The benchmark provides a common closed-loop protocol for comparing UMI policies in both in-domain and generalization settings. Environment and embodiment shifts are tested separately and jointly under a consistent UMI observation interface, with all generalization tests using the trained policies without additional fine-tuning.

Spatial-UMI-Dream achieves 79.10% in-domain success on *RoboTwin-UMI* and 85.0% average success across six real-world tasks, exceeding the strongest evaluated baselines by 4.65 and 24.2 percentage points, respectively. Its advantage extends to generalization: the lead over the strongest baseline widens to 14.00 percentage points after embodiment transfer, and *Spatial-UMI-Dream* remains the strongest policy under environmental and joint shifts.

Our contributions are twofold:

- **Spatial world-action modeling at scale.** We introduce *Spatial-UMI-Dream* with two-stage training on 605 hours of RGB-D UMI data, achieving the highest in-domain and generalization success rates among evaluated policies. Ablations demonstrate the benefit of metric depth and progressive gains from the two training stages.
- **A benchmark for UMI policy generalization.** We develop the twenty-task *RoboTwin-UMI* benchmark with wrist-mounted fisheye observations and a common protocol for evaluating in-domain performance and generalization across environments and robot embodiments, including their combined shifts.

II. RELATED WORK

A. Robot Foundation Models

Robot foundation models seek to reuse broad visual-language knowledge for control [5], [10], enabling policies to interpret diverse scenes and instructions without learning each task in isolation. Representative action-centric Vision-Language-Action models [2], [3], [4], [5] adapt pretrained vision-language representations to predict robot actions from visual observations and language instructions. However, predicting actions alone does not explicitly describe how the scene is expected to evolve, leaving future outcomes unavailable as a direct policy context or planning signal. More recent methods [11], [12], [13] introduce world models into this paradigm without fully coupling world and action generation. By using predicted visual futures as policy context or planning supervision, these approaches improve policy generalization and planning while retaining a separation between world prediction and action generation. World Action Models [6], [7], [8], [14] instead jointly generate future observations and actions, directly connecting predicted scene evolution with control. Traditional RGB-only WAMs leave metric geometry implicit in visual appearance. In contrast, *Spatial-UMI-Dream* jointly predicts RGB, registered metric depth, and actions, making metric geometry an explicit component of the world state.

B. Geometry-Aware Robot Learning

Geometric representations support spatial reasoning in robotic perception and control [15], [16], [17], [18], providing explicit cues about scene layout and spatial relationships. 3D visuomotor policies explicitly encode depth or point clouds into geometric scene features, allowing action prediction to operate on metric structure rather than infer it only from visual appearance. Spatially grounded Vision-Language-Action models [19], [20] further incorporate explicit spatial representations into language-conditioned action prediction. Representation-aligned policies [21], [22] instead distill geometric features from pretrained 3D models into RGB policy backbones, providing spatial supervision without requiring explicit 3D inputs at deployment. In contrast, *Spatial-UMI-Dream* treats metric depth as an observed and predicted component of the world state rather than only as a

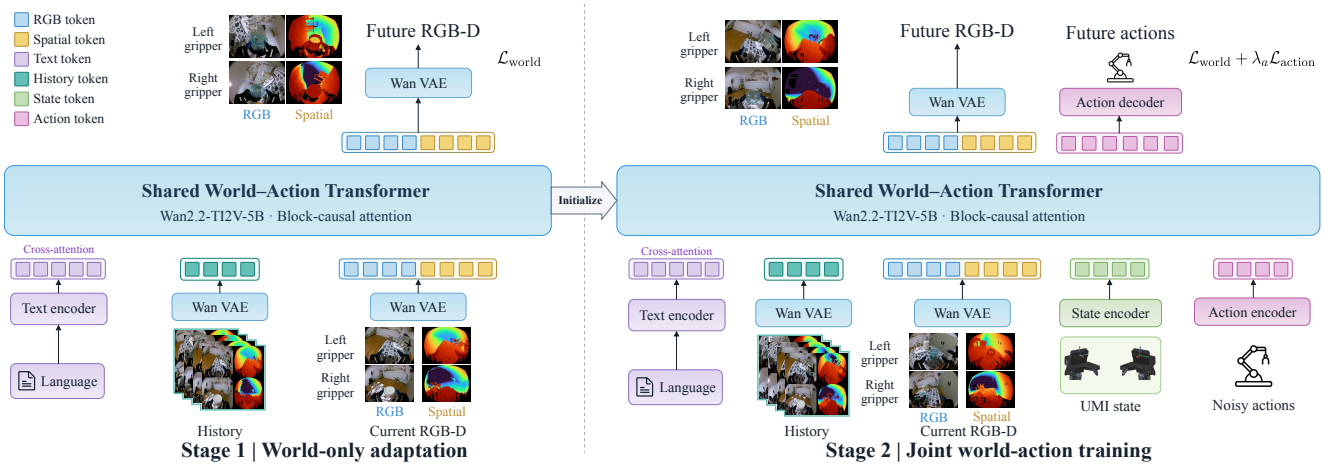


Fig. 2: **Model architecture and two-stage training of *Spatial-UMI-Dream*.** In both stages, dual-gripper RGB and metric depth (Fig. 3) form a unified RGB-D canvas. The Wan VAE encodes observation history and current RGB-D inputs into visual tokens, while a text encoder supplies language tokens through cross-attention. The shared world-action transformer uses **block-causal attention** to condition prediction on preceding visual history. **Stage 1:** world-only adaptation predicts future RGB-D observations, decoded through the Wan VAE and supervised by $\mathcal{L}_{\text{world}}$. **Stage 2:** the transformer backbone is initialized from the Stage 1 world-only adaptation checkpoint. Joint world-action training additionally encodes UMI state and noisy actions, and jointly predicts future RGB-D observations and actions through the Wan VAE and action decoder, optimized with $\mathcal{L}_{\text{world}} + \lambda_a \mathcal{L}_{\text{action}}$.

policy input or representation-level training target. It jointly predicts dual-gripper RGB, metric depth, and actions through a unified video canvas, without a separate 3D encoder or alignment teacher.

C. UMI Data and Evaluation

Diverse robot datasets and standardized evaluation platforms [23], [24] support systematic comparison of learned manipulation policies. UMI [1] enables robot-independent demonstration collection by recording wrist-mounted fisheye observations and end-effector trajectories with a portable handheld gripper. Large-scale robot policies [4], [25], [26] leverage UMI and UMI-style handheld demonstrations for generalist control and zero-shot cross-embodiment transfer. More recent UMI-oriented VLA training [27] studies fisheye-aware visual grounding and embodiment-conditioned physical validation to reconcile handheld demonstrations with pretrained visual representations and target-robot constraints. UMI demonstrations collected across diverse scenes support learning policies beyond a single environment, motivating evaluation under environmental changes. Moreover, because UMI decouples demonstration collection from a specific robot, cross-embodiment evaluation tests whether this flexibility also carries over to the learned policies. We introduce *RoboTwin-UMI*, a benchmark built on RoboTwin [9] that evaluates these two capabilities separately and jointly through a controlled, multi-task simulation protocol under a consistent wrist-mounted fisheye observation interface. This enables repeatable comparisons of different policies on the same tasks.

III. METHOD

We study language-conditioned manipulation from dual-gripper UMI observations. At timestep t, the observation is $o_t = \{I_t^e, D_t^e\}_{e\in\{0,1\}}$, where I_t^e and D_t^e are the RGB image and aligned metric depth in gripper e's native fisheye view. Given visual history, robot state s_t, and language instruction ℓ, our model jointly predicts future RGB-D observations and temporally aligned robot action chunks for coordinated manipulation with both grippers.

Our method constructs a metric RGB-D representation, jointly models visual observations and actions, and learns the policy through two-stage training. Trajectory filtering and deployment command compensation connect handheld demonstrations to robot execution.

A. Metric RGB-D Representation

1) Metric depth reconstruction: We use the GenRobot DAS UMI system, with three calibrated cameras mounted on each gripper. As illustrated in Fig. 3, the middle and right cameras form a stereo pair, with the middle camera providing the reference RGB observation I_t^e. We epipolar-rectify the two images and use Fast FoundationStereo [28] to estimate disparity in the rectified middle-camera image coordinates. We convert disparity to metric Z-depth using the stereo calibration, then map the depth back to the middle camera's original image coordinates, accounting for its lens distortion. The resulting D_t^e is aligned with the native fisheye RGB observation used by the policy.

2) Three-channel depth encoding: Inspired by Marigold [29], we encode depth through the pretrained image VAE used for RGB. We normalize metric depth using fixed bounds $d_{\min}$ and $d_{\max}$ shared across the dataset, then apply a three-channel color mapping ϕ:

$$\Phi(D) = \phi\left(\frac{d_{\max} - \text{clip}(D, d_{\min}, d_{\max})}{d_{\max} - d_{\min}}\right). \quad (1)$$

We choose Turbo as the color mapping ϕ. The fixed bounds maintain a consistent mapping from physical distance to encoded values. Depth preprocessing is identical during offline data construction and online deployment.

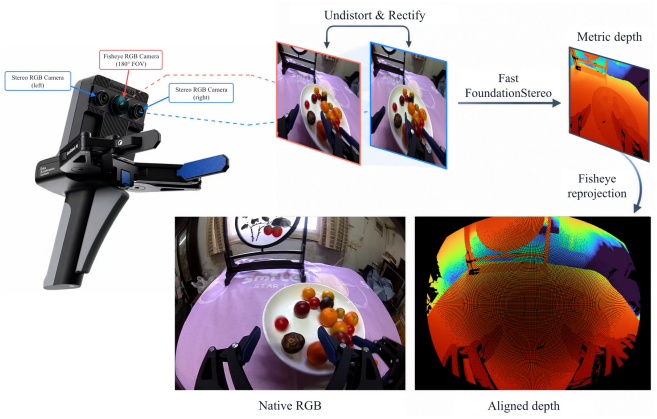


Fig. 3: **Metric depth construction for UMI observations.** Each UMI gripper has three cameras. We undistort and rectify the middle and right camera views, then use Fast FoundationStereo to recover metric depth. We map the depth back to the original middle-camera fisheye view to align it with the RGB observation.

3) Dual-gripper RGB-D canvas: We organize the two RGB views and their encoded depth maps into an eye-paired spatial canvas,

$$C_t = \begin{bmatrix} I_t^0 & \Phi(D_t^0) \\ I_t^1 & \Phi(D_t^1) \end{bmatrix}. \tag{2}$$

The rows correspond to the two grippers and the columns to appearance and geometry, so each RGB observation remains spatially adjacent to the depth from the same eye. Each panel is resized to 160×160, producing a 320×320 frame with a fixed layout across training and deployment. We encode the complete canvas as an ordinary video frame, $z_t = \mathcal{E}_{\text{vid}}(C_t)$, using the pretrained Wan2.2 VAE [30] rather than separate modality-specific encoders. Appearance and encoded geometry consequently follow the same video codec, spatiotemporal patch embedding, and video objective, without a depth-specific branch or explicit 3D backbone.

B. Joint World–Action Modeling

As shown in Fig. 2, action and state encoders map their inputs into the shared transformer token space, while language provides conditioning through cross-attention. The transformer jointly denoises video and action representations, which are decoded into RGB-D canvases and executable actions by the video VAE and action decoder, respectively. With block-causal attention, generation proceeds in temporally aligned RGB-D latent blocks Z_i and action chunks A_i:

$$(Z_i, A_i) \sim p_\theta(Z_i, A_i \mid Z_{<i}, s_i, \ell). \tag{3}$$

Here $Z_{<i}$ denotes preceding visual blocks within a finite context window. Video and action tokens interact bidirectionally within each block and attend to preceding visual blocks, allowing visual history to guide the generation of the current world-action block. During training, we use teacher forcing [6], conditioning each noisy world-action block on clean, ground-truth visual history.

We train the video and action streams with a weighted flow-matching objective [31]. For a clean target x and Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, we construct $x_\tau = (1-\tau)x + \tau\epsilon$ and predict the velocity $\epsilon - x$:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}\Big[w(\tau)\, \|v_\theta(x_\tau, \tau; c) - (\epsilon - x)\|_2^2\Big]. \tag{4}$$

Here $w(\tau)$ is the scheduler's noise-level weight, and c denotes the conditioning inputs of the corresponding stage. Applying this objective to the RGB-D video latents and action sequences gives $\mathcal{L}_{\text{world}}$ and $\mathcal{L}_{\text{action}}$, respectively. In implementation, squared errors are averaged over tensor elements, and invalid action dimensions and samples without action supervision are masked.

C. Two-Stage World–Action Training

We build *Spatial-UMI-Dream* on the pretrained Wan2.2-TI2V-5B [30] model and train it on the 605-hour RGB-D UMI corpus in two stages: (1) world-only adaptation adapts the backbone to the spatial RGB-D canvas; (2) joint world-action training couples this representation with action generation. Both stages use observation history and language conditioning, with metric depth present throughout.

1) World-only adaptation: We optimize $\mathcal{L}_{\text{world}}$ (Eq. (4)) to predict future RGB-D canvases conditioned on visual history and language. The action branch is bypassed, leaving the action encoder and decoder untrained, and robot-state conditioning is disabled in this stage. This stage adapts the pretrained backbone to moving UMI fisheye observations and the fixed dual-gripper RGB-D layout.

2) Joint world-action training: We restore the world-only checkpoint and introduce the action stream and robot-state conditioning. The action encoder and decoder retain their random initial parameters, which were saved but not trained during the first stage, and begin learning jointly with the adapted backbone. The model processes noisy video and action inputs together, conditioned on observation history, language, and robot state, and minimizes

$$\mathcal{L} = \mathcal{L}_{\text{world}} + \lambda_a \mathcal{L}_{\text{action}}. \tag{5}$$

Here λ_a controls the relative weight of action supervision. Both $\mathcal{L}_{\text{world}}$ and $\mathcal{L}_{\text{action}}$ are defined by Eq. (4). Both stages use the same underlying RGB-D UMI corpus.

D. Demonstration Filtering and Deployment

Handheld demonstrations and robot execution differ in kinematic feasibility and gripper response. We address these differences by filtering unsuitable trajectories before training and compensating gripper commands at deployment.

1) Trajectory filtering: Handheld UMI collection is not constrained by the target robot's kinematics or execution limits. We therefore filter demonstrations before training to avoid supervising the policy with motions the robot cannot reliably execute. We solve inverse kinematics for Dexmate Vega1 and replay the motions in ManiSkill [32] to identify and discard unsuitable episodes (Fig. 4). We also discard entire episodes when any velocity, acceleration, or jerk value exceeds its dataset-wide 99th percentile.

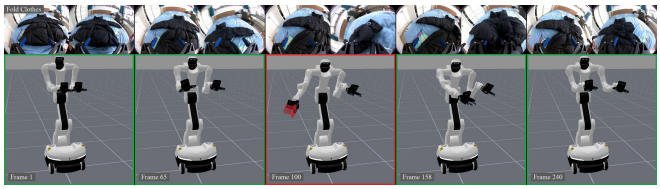


Fig. 4: **UMI trajectory filtering.** Wrist observations (top) and simulation replay (bottom) of a folding demonstration. The third frame violates robot kinematic limits, so the entire episode is discarded.

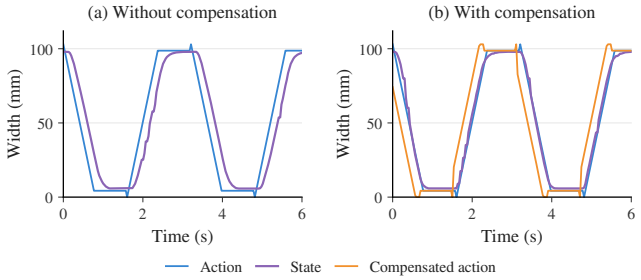


Fig. 5: **Gripper compensation at deployment.** Desired actions (blue), measured states (purple), and compensated actions (orange) of one gripper (a) without and (b) with compensation under the same continuous target trajectory. Both panels show the first 6 seconds without smoothing. Mean absolute tracking error over this interval decreases from 12.49 to 2.85 mm.

2) Deployment command compensation: UMI gripper actions specify desired physical openings, but directly issuing them to the robot controller produces delayed gripper responses (Fig. 5(a)). To align the executed gripper state with the predicted action trajectory, following [33], we calibrate a first-order response model with delay and invert it to compensate the commands:

$$u_k = \mathrm{clip}\left(\frac{r_{k+1+\delta} - \alpha r_{k+\delta}}{1-\alpha},\, 0,\, w_{\max}\right), \tag{6}$$

Here r_k and u_k denote the desired width and compensated command, δ is the delay in control steps, $\alpha = \exp(-h/\tau)$ for control period h and calibrated response time constant τ, and $w_{\max}$ is the maximum opening. Advancing and pre-amplifying the commands aligns the physical response with the desired trajectory under the calibrated model (Fig. 5(b)).

E. RoboTwin-UMI Benchmark

To test whether the flexibility of robot-independent UMI data collection translates into cross-embodiment policy transfer, we equip two robot embodiments with a common UMI gripper and observation interface. To assess environmental robustness, we vary lighting conditions while retaining the same tasks. We introduce *RoboTwin-UMI*, a twenty-task benchmark built on RoboTwin [9] that evaluates these changes separately and jointly.

1) Benchmark Construction: We retain twenty RoboTwin tasks spanning object placement, stacking, and articulated-object interaction, and adapt the robot setups for UMI execution and observation. Specifically, we equip Aloha-AgileX and ARX-X5 with UMI grippers and wrist-mounted fisheye

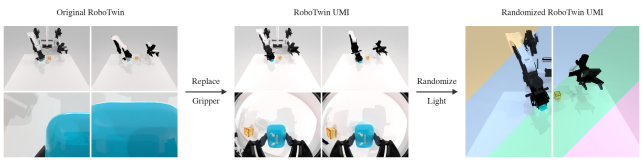


Fig. 6: ***RoboTwin-UMI* benchmark construction.** We retain the same RoboTwin tasks while adapting the grippers and observations for UMI. From left to right: the original setup; UMI grippers and wrist-mounted fisheye cameras on Aloha-AgileX and ARX-X5; and randomized lighting. These settings support evaluation of cross-embodiment and environmental generalization, separately and jointly.

cameras (Fig. 6), matching the real-world UMI camera setup. Both embodiments share the same evaluation tasks (Fig. 8b).

2) Fisheye Observation Rendering: To reproduce UMI's wide-field-of-view fisheye observations in simulation, we synthesize each image from four pinhole views sharing a common optical center: one forward-facing view covers the central region, while three auxiliary views face outward in different directions to cover the periphery. For each output pixel $\mathbf{p}$, the calibrated Kannala–Brandt (KB4) model gives a viewing ray. We then project this ray into a pinhole image to locate the corresponding source pixel $\mathbf{q}_j$:

$$\mathbf{q}_j = \pi_{K_j}\left(R_j^\top \pi_{\mathrm{KB4}}^{-1}(\mathbf{p})\right), \tag{7}$$

where R_j denotes camera j's rotation relative to the fisheye reference frame, $R_j^\top$ maps the ray into camera j's coordinates, and π_{K_j} denotes the pinhole projection with intrinsics K_j, mapping the ray to image coordinates $\mathbf{q}_j$. If multiple cameras cover the ray, we use the one whose optical axis is closest to it, and bilinearly sample its RGB image at $\mathbf{q}_j$.

3) Generalization Evaluation Protocol: To distinguish environmental robustness from cross-embodiment transfer, we vary lighting and robot embodiment separately and jointly while retaining the same twenty tasks. We train all models jointly on these tasks using 50 clean Aloha-AgileX demonstrations per task and a unified action and state representation. Evaluation crosses Aloha-AgileX and ARX-X5 with two lighting conditions: *clean* retains default achromatic illumination, while *randomized lighting* resamples point- and directional-light colors at each episode's start. The resulting four settings serve distinct purposes: clean Aloha establishes in-domain performance; randomized-lighting Aloha tests environmental robustness; clean ARX-X5 tests embodiment transfer; and randomized-lighting ARX-X5 tests both changes together. All generalization settings use the trained policies without additional training data or fine-tuning, so the comparisons measure transfer rather than adaptation to the target setting. For each task in each setting, we run 100 evaluation episodes and report the success rate; aggregate results are averaged over the twenty tasks.

IV. EXPERIMENTS

We evaluate the spatial WAM through closed-loop manipulation rather than standalone perceptual video metrics. Our experiments are designed to answer three questions:

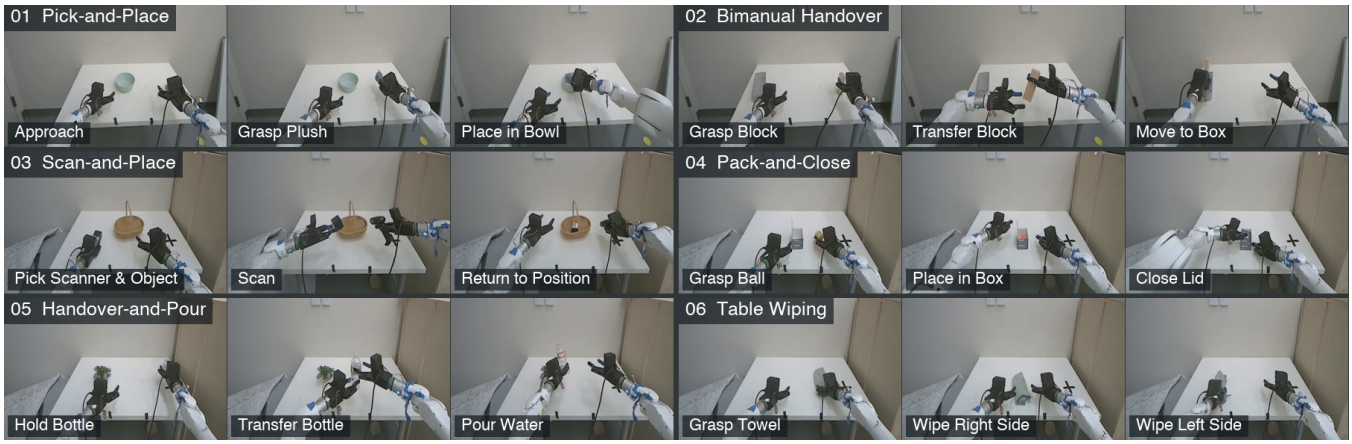


Fig. 7: **Real-world task suite.** We show egocentric keyframes from six real-world tasks, including unimanual manipulation and bimanual coordination. These views are for visualization only and are not used as policy inputs.

TABLE I: **Real-world bimanual manipulation.** We report full-episode success rate (SR) and average progress score (Score), both in percent, over 20 evaluation episodes per task. Entries show SR / Score; Avg. is the unweighted mean over the six tasks.

Method	Pick-and-Place	Handover	Scan-and-Place	Pack-and-Close	Handover-and-Pour	Table Wiping	Avg.
$\pi_{0.5}$	70.0 / 86.7	45.0 / 61.3	50.0 / 82.0	40.0 / 40.0	60.0 / 70.0	**100.0 / 100.0**	60.8 / 73.3
RDT2-FM	70.0 / 76.7	5.0 / 35.0	5.0 / 42.0	30.0 / 47.5	45.0 / 63.0	85.0 / 92.5	40.0 / 59.4
Fast-WAM	50.0 / 81.7	40.0 / 71.3	45.0 / 67.0	30.0 / 42.5	25.0 / 45.0	**100.0 / 100.0**	48.3 / 67.9
LingBot-VA	60.0 / 86.7	5.0 / 35.0	55.0 / **91.0**	40.0 / 55.0	25.0 / 67.0	75.0 / 95.8	43.3 / 71.8
Spatial-UMI-Dream	**95.0 / 98.3**	**80.0 / 93.8**	**75.0** / 87.0	**60.0 / 80.0**	**100.0 / 100.0**	**100.0 / 100.0**	**85.0 / 93.2**

(Q1) How does *Spatial-UMI-Dream* compare with existing VLAs and WAMs on in-domain manipulation on *RoboTwin-UMI* and in the real world? (Section IV-B)

(Q2) Does *Spatial-UMI-Dream* generalize better than baselines beyond the training conditions? (Section IV-C)

(Q3) How do metric depth and two-stage UMI pretraining contribute to downstream manipulation performance? (Section IV-D)

A. Experimental Setup

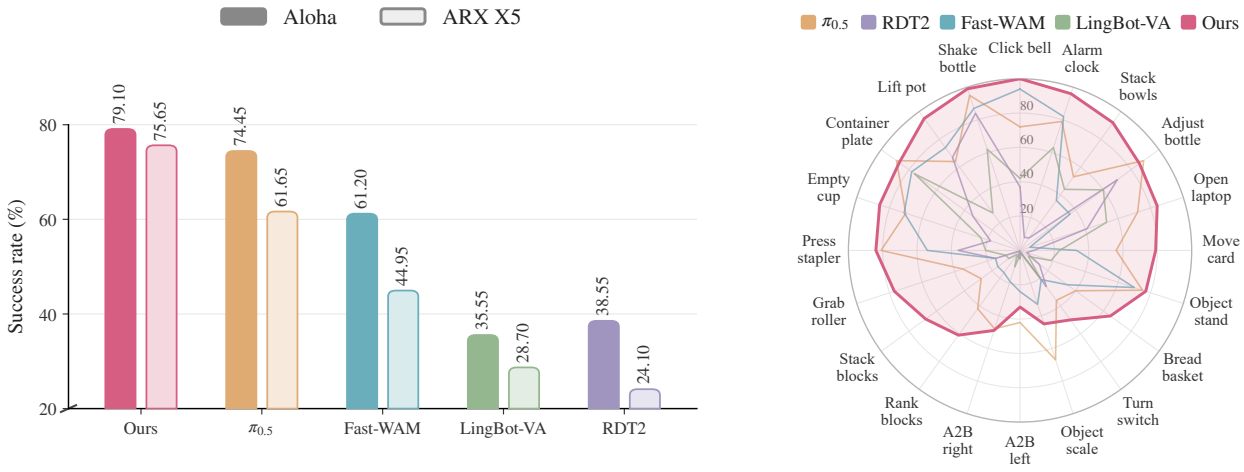
1) Real-world tasks: We construct a challenging real-world bimanual UMI evaluation on the Dexmate Vega1 mobile robot (Vega1), spanning precision-sensitive placement, coordinated handovers, tool use, long-horizon manipulation, liquid transfer, and sustained contact. As illustrated in Fig. 7, its six tasks are **Pick-and-Place**, placing a plush toy into a bowl; **Bimanual Handover**, transferring a wooden block between grippers and into a box; **Scan-and-Place**, scanning and subsequently placing a product; **Pack-and-Close**, packing a baseball and closing its box; **Handover-and-Pour**, transferring a bottle between hands before watering flowers; and **Table Wiping**, wiping the table from right to left. For each task, we evaluate 20 episodes and report success rate (SR), the fraction of fully completed episodes, and progress score (Score), the average fraction of completed task stages. Both metrics are averaged over the six tasks.

2) Baselines: We compare *Spatial-UMI-Dream* with a generalist VLA, $\pi_{0.5}$ [2], which learns manipulation skills from diverse robot data, and a UMI-specific VLA, RDT2-FM [4], which focuses on learning from UMI demonstrations. For WAMs, we choose Fast-WAM [34] and LingBot-VA [35] as general-purpose WAM baselines. Baselines are initialized from their released base checkpoints, while *Spatial-UMI-Dream* uses our spatial world-action pretrained checkpoint. On both *RoboTwin-UMI* and real-world tasks, all methods are fine-tuned on the same demonstrations.

3) Training protocol: Across all evaluation setups, all policies use a 20-dimensional relative action $\mathbf{a} = [\Delta p_L, r_L^{6D}, g_L, \Delta p_R, r_R^{6D}, g_R]$, with chunk size 24, where $\Delta p \in \mathbb{R}^3$ denotes translation, $r^{6D} \in \mathbb{R}^6$ [36] denotes rotation, and $g \in [0, 0.1]$ denotes the absolute gripper opening width in meters. For pretraining, we follow the two-stage procedure in Section III-C, training each stage for 30,000 steps on the same 605-hour RGB-D UMI dataset using 24 NVIDIA RTX PRO 6000 GPUs. For downstream tasks, we initialize *Spatial-UMI-Dream* from its pretrained checkpoint and fine-tune it separately for real-world tasks and *RoboTwin-UMI*, using 8 GPUs for each fine-tuning run and retaining the pretrained action representation.

B. In-Domain Manipulation Performance on RoboTwin-UMI and in Real-World Environments

To answer **Q1**, we compare in-domain manipulation performance in the real world and on *RoboTwin-UMI*. Across six real-world tasks, *Spatial-UMI-Dream* achieves 85.0% average success, compared with 60.8% for the strongest baseline, $\pi_{0.5}$ (Table I). In Bimanual Handover, $\pi_{0.5}$ succeeds in 45.0% of episodes, versus 80.0% for our model. Transferring the wooden block requires **centimeter-level alignment** between the two UMI grippers. Competing policies exhibit gripper misalignment during transfer, highlighting the difficulty of estimating gripper–object distances from



(a) Average success across embodiments.

(b) Cross-embodiment success on ARX X5.

Fig. 8: ***RoboTwin-UMI* evaluation across twenty tasks.** (a) Mean success rates on Aloha and ARX X5. (b) Per-task cross-embodiment success rates on ARX X5. *Spatial-UMI-Dream* achieves the highest mean success rate on both embodiments and the smallest drop.

moving wrist RGB observations. In contrast, explicit metric depth provides our model with distance cues for aligning the receiving gripper with the held block.

The advantage extends to the twenty-task *RoboTwin-UMI* benchmark. As shown in Fig. 8a, *Spatial-UMI-Dream* achieves the highest in-domain average success rate of 79.10%, outperforming the strongest VLA baseline, $\pi_{0.5}$, at 74.45% by 4.65 percentage points. The gap is even larger against competing WAMs: Fast-WAM reaches 61.20% and LingBot-VA reaches 35.55%, placing *Spatial-UMI-Dream* 17.90 percentage points ahead of the strongest WAM baseline. Together, these results establish strong in-domain performance in simulation and the real world.

C. Generalization Across Embodiments and Environments

To answer **Q2**, we analyze the three generalization settings defined in Section III-E. Fig. 8a reveals a clear separation that in-domain scores alone obscure. Although *Spatial-UMI-Dream* and $\pi_{0.5}$ are separated by only 4.65 percentage points on Aloha, their gap widens to 14.00 percentage points after transfer to ARX X5. *Spatial-UMI-Dream* achieves the highest cross-embodiment success rate with the smallest performance drop, showing that its advantage extends beyond strong performance on the training embodiment.

Cross-embodiment generalization requires preserving precise approach and contact behavior, not merely reaching the correct target. In a Click Alarmclock failure case, Fast-WAM approaches the clock but stops short of its top button, failing to establish the contact needed to trigger it. UMI pretraining equips *Spatial-UMI-Dream* with manipulation priors that support fine-grained control beyond the training embodiment. World-only adaptation learns from UMI observations, while joint world-action pretraining connects these observations with end-effector actions, providing a foundation for transferring approach and contact skills to a different robot. The pretraining ablation in Section IV-D further supports its contribution to cross-embodiment performance.

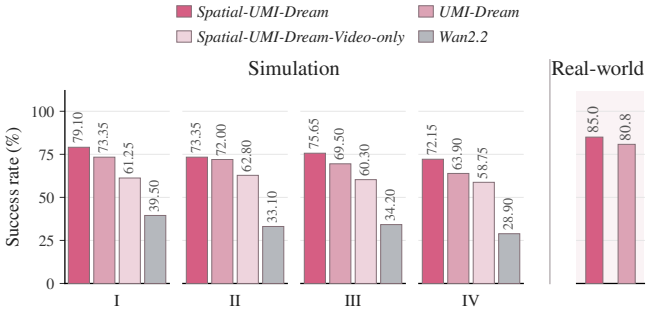


Fig. 9: **Ablation of depth and UMI pretraining.** Simulation: I, Aloha clean; II, Aloha with randomized lighting; III, ARX X5 clean; IV, ARX X5 with randomized lighting. Bars report mean success over all twenty tasks. Real-world: mean success over six tasks for *Spatial-UMI-Dream* (RGB-D) and *UMI-Dream* (RGB).

Under randomized lighting, Place Empty Cup exposes a different failure mode. The policy must grasp a cup from the table and place it on a coaster. In this case, Fast-WAM fails to localize the coaster accurately after illumination changes, leading to unsuccessful placement. This task requires maintaining spatial alignment with the placement target despite

TABLE II: **Environment and joint-shift generalization on *RoboTwin-UMI*.** Mean success rates (%) over twenty tasks: Aloha under clean and randomized lighting, and ARX X5 under randomized lighting.

Method	Aloha Clean ↑	Aloha Rand. Light ↑	ARX X5 Rand. Light ↑
$\pi_{0.5}$	74.45	66.15	61.85
RDT2	38.55	16.55	12.40
Fast-WAM	61.20	18.85	16.90
LingBot-VA	35.55	34.85	28.60
Spatial-UMI-Dream	**79.10**	**73.35**	**72.15**

changes in its appearance. By modeling RGB together with metric depth, ***Spatial-UMI-Dream*** uses explicit scene geometry alongside appearance cues to guide placement.

According to Table II, ***Spatial-UMI-Dream*** achieves 73.35% success on Aloha with randomized lighting and 72.15% under combined embodiment and environmental shifts, exceeding the strongest evaluated baseline by 7.20 and 10.30 percentage points, respectively. Taken together, these results show that ***Spatial-UMI-Dream*** sustains its in-domain advantage across embodiment and environmental shifts, with limited performance degradation.

D. Ablation of Depth and Pretraining

To answer **Q3**, we compare four variants using identical downstream demonstrations within each evaluation setup. *UMI-Dream* and *Spatial-UMI-Dream* share the same UMI demonstrations, two-stage training recipe, training budget, and 2×2 visual layout. They differ only in depth: the RGB-only variant duplicates its RGB views in place of depth panels to match the visual token count, while the full model uses aligned RGB-D observations. *Spatial-UMI-Dream-Video-only* uses the same RGB-D data but only world-only adaptation, without action pretraining. The pretrained *Wan2.2* model is directly fine-tuned on downstream RGB-D demonstrations without UMI pretraining.

1) Depth enables more reliable control: Metric depth improves success across all four simulation settings (Fig. 9), by 5.75 percentage points in-domain and 8.25 under joint shifts. Stack Bowls Two requires moving two bowls to the table center and stacking them. Successful stacking depends on reaching each bowl at the correct distance and aligning it with the bowl underneath. The RGB-only policy misses grasps and misaligns the bowls, indicating difficulty in judging gripper–object distance and relative placement. Depth supplies explicit spatial cues for approaching grasp targets and aligning the bowls during placement. With demonstrations, training budget, and token count matched, these gains support **the value of geometric information rather than additional visual tokens**. On six real-world tasks, RGB-D raises mean success from 80.8% to 85.0% and improves mean progress.

2) Two-stage pretraining improves downstream control: World-only adaptation raises in-domain simulation success from 39.50% (*Wan2.2*) to 61.25%, and joint world-action pretraining further raises it to 79.10% (Fig. 9). World-only adaptation exposes the pretrained backbone to moving UMI views and their metric geometry before downstream learning. Since *Wan2.2* already provides a pretrained visual prior, this gain highlights the value of UMI-specific RGB-D experience even before action pretraining. Joint pretraining then couples these visual representations with end-effector actions, improving control beyond visual adaptation alone. The full model exceeds video-only pretraining by 15.35 and 13.40 percentage points under embodiment and joint shifts, respectively, and **performs best in all four settings**. These gains show that the benefit extends beyond the robot used for downstream training.

V. CONCLUSION

We introduced *Spatial-UMI-Dream*, a 5B spatial WAM that jointly models RGB, fisheye-aligned metric depth, and actions through two-stage pretraining on 605 hours of UMI data. We also introduced *RoboTwin-UMI*, a twenty-task benchmark for controlled evaluation of UMI policies across robot embodiments and environmental changes, separately and jointly. Experiments demonstrate strong in-domain performance and generalization, with 85.0% mean success on six real-world tasks and the highest success rates across all benchmark settings. Matched RGB-only ablations establish the benefit of explicit metric geometry, while pretraining ablations show complementary gains from world-only adaptation and joint world-action learning. These gains extend to precise grasping and placement under moving wrist observations. Together, these results highlight the value of geometry-aware UMI pretraining for reliable manipulation and of controlled benchmarks for assessing policy generalization beyond training conditions.

REFERENCES

[1] C. Chi *et al.*, "Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots," 2024, arXiv:2402.10329.

[2] Physical Intelligence *et al.*, "$\pi_{0.5}$: a vision-language-action model with open-world generalization," 2025, arXiv:2504.16054.

[3] S. Wei *et al.*, "Ψ_0: An open foundation model towards universal humanoid loco-manipulation," *arXiv preprint arXiv:2603.12263*, 2026.

[4] S. Liu *et al.*, "RDT2: Exploring the scaling limit of UMI data towards zero-shot cross-embodiment generalization," 2026, arXiv:2602.03310.

[5] M. Kim *et al.*, "OpenVLA: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

[6] S. Ye *et al.*, "World action models are zero-shot policies," 2026, arXiv:2602.15922.

[7] NVIDIA *et al.*, "Cosmos 3: Omnimodal world models for physical AI," 2026, arXiv:2606.02800.

[8] Dyna Robotics, "Dyna-2: A 1-million-hour scaling law for world-action models," August 2026. [Online]. Available: https://dyna.co/dyna-2

[9] T. Chen *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.

[10] Y. Wu *et al.*, "Large reward models: Generalizable online robot reward generation with vision-language models," 2026. [Online]. Available: https://arxiv.org/abs/2603.16065

[11] Physical Intelligence *et al.*, "$\pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities," 2026, arXiv:2604.15483.

[12] E. Y.-T. Jia, W. Yuan, T. Shi, V. Guizilini, J. Mao, and Y. Wang, "DreamPlan: Efficient reinforcement fine-tuning of vision-language planners via video world models," 2026, arXiv:2603.16860.

[13] Z. Zhang *et al.*, "ForeAct: Steering your VLA with efficient visual foresight planning," 2026, arXiv:2602.12322.

[14] H. Bi *et al.*, "Motus: A unified latent action world model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026, pp. 35 101–35 113.

[15] W. Yuan, J. Li, J. Yue, D. Shah, K. Karydis, and H. Qiu, "BEVCALIB: LiDAR-camera calibration via geometry-guided bird's-eye view representations," in *9th Annual Conference on Robot Learning*, 2025.

[16] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3D diffusion policy: Generalizable visuomotor policy learning via simple 3D representations," in *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[17] T.-W. Ke, N. Gkanatsios, and K. Fragkiadaki, "3D diffuser actor: Policy diffusion with 3D scene representations," 2024. [Online]. Available: https://arxiv.org/abs/2402.10885

[18] T. Nguyen, W. Yuan, S. Wei, H. Li, D. Seita, and Y. Wang, "ICLR: In-context imitation learning with visual reasoning," 2026. [Online]. Available: https://arxiv.org/abs/2603.07530

[19] H. Zhen *et al.*, "3D-VLA: A 3D vision-language-action generative world model," 2024, arXiv:2403.09631.
[20] D. Qu *et al.*, "SpatialVLA: Exploring spatial representations for visual-language-action model," 2025, arXiv:2501.15830.
[21] F. Li *et al.*, "Spatial forcing: Implicit spatial representation alignment for vision-language-action model," 2025. [Online]. Available: https://arxiv.org/abs/2510.12276
[22] G. Sun *et al.*, "ROCKET: Residual-oriented multi-layer alignment for spatially-aware vision-language-action models," *arXiv preprint arXiv:2602.17951*, 2026.
[23] Z. Zhao *et al.*, "Humanoid everyday: A comprehensive robotic dataset for open-world humanoid manipulation," 2026. [Online]. Available: https://arxiv.org/abs/2510.08807
[24] S. Wei *et al.*, "SIMPLE: Simulation-based policy learning and evaluation for humanoid loco-manipulation," *arXiv preprint arXiv:2606.08278*, 2026.
[25] Generalist Team, "GEN-1.5: Embodied foundation models are one-shot learners," *Generalist AI Blog*, 2026, https://generalistai.com/blog/gen-1.5.
[26] H. Zhang *et al.*, "Hy-Embodied-0.5-VLA: From vision-language-action models to a real-world robot learning stack," *arXiv preprint arXiv:2606.14409*, 2026.
[27] S. Yang *et al.*, "VISTA: Vision-grounded and physics-validated adaptation of UMI data for VLA training," *arXiv preprint arXiv:2606.04708*, 2026.
[28] B. Wen, S. Dewan, and S. Birchfield, "Fast-FoundationStereo: Real-time zero-shot stereo matching," *CVPR*, 2026.
[29] B. Ke *et al.*, "Repurposing diffusion-based image generators for monocular depth estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
[30] Team Wan *et al.*, "Wan: Open and advanced large-scale video generative models," 2025, arXiv:2503.20314.
[31] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," 2023, arXiv:2210.02747.
[32] S. Tao *et al.*, "ManiSkill3: GPU parallelized robotics simulation and rendering for generalizable embodied AI," *arXiv preprint arXiv:2410.00425*, 2024.
[33] C. Yang, Y. Hu, Y. Ma, Y. Yang, J. Tan, and H. Fan, "Realtime-VLA V2: Learning to run VLAs fast, smooth, and accurate," 2026, arXiv:2603.26360.
[34] T. Yuan, Z. Dong, Y. Liu, and H. Zhao, "Fast-WAM: Do world action models need test-time future imagination?" *arXiv preprint arXiv:2603.16666*, 2026.
[35] L. Li *et al.*, "Causal world modeling for robot control," *arXiv preprint arXiv:2601.21998*, 2026.
[36] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

APPENDIX

a) Training recipe: Spatial-UMI-Dream is initialized from Wan2.2-TI2V-5B [30]. The 605-hour corpus combines two object-picking collections and a broader high-quality manipulation collection, sampled in proportion to their valid trajectory steps. World-only adaptation uses all source episodes, whereas joint world-action training uses only the action-supervised episodes retained after the filtering in Section III-D. In Eq. (1), $d_{\min} = 0.01\,\mathrm{m}$ and $d_{\max} = 1.5\,\mathrm{m}$, and pixels without valid depth are encoded as black.

We train the full diffusion transformer and keep the text encoder, image encoder and video VAE frozen. All stages use AdamW with learning rate 10^{-5}, $(\beta_1, \beta_2) = (0.95, 0.999)$, $\epsilon = 10^{-8}$ and weight decay 10^{-5}, a cosine schedule with warmup over the first 5% of updates, bfloat16, and DeepSpeed ZeRO-2. Joint world-action training and fine-tuning use $\lambda_a = 1$. Table III lists the stage-specific settings.

TABLE III: Stage-specific training settings.

Stage	Updates	Batch	GPUs
World-only adaptation	30k	192	24
Joint world-action	30k	192	24
Real-world fine-tuning	22.5k	64	8
RoboTwin-UMI fine-tuning	10k	128	8

b) RoboTwin-UMI setup: Fisheye observations follow Eq. (7): the four co-centered source views are rendered at 320×320, with a 92° forward view and 106° auxiliary views, and resampled to a 320×240 KB4 image using the same left and right calibrations on both robots. For adaptive-grasp tasks, *RoboTwin-UMI* computes the target gripper opening from the object and finger-pad collision geometry instead of using a fixed closure. It also replaces RoboTwin's per-arm cuRobo and MPLib planning with a PyRoKi formulation, solved with jaxls, that plans the twelve joints of both arms jointly, checks self- and environment collisions along the interpolated path, and time-parameterizes the result under velocity, acceleration and jerk limits.

For cross-embodiment evaluation, *RoboTwin-UMI* fits the Aloha-AgileX and ARX-X5 models from the official RoboTwin release [9] with the DAS/UMI gripper in place of their native grippers, adds calibrated wrist-fisheye RGB and depth cameras, and uses the planner above for both. The two embodiments therefore differ only in their URDFs, base transforms, home configurations and collision models. A single command $g \in [0, 1]$ (0 closed, 1 open) drives the gripper's six mimic-coupled joints through a driving-joint angle of $0.925(1 - g)$ rad. The gripper is mounted at a 15° pitch, and its TCP lies at $(0.0965, 0, -0.043)$ m in the gripper base frame.

For randomized-lighting evaluation, *RoboTwin-UMI* reuses RoboTwin's light-color randomization: each point- and directional-light RGB color is drawn from $U[0, 1)^3$ at episode start. RoboTwin's background, clutter, table-height and camera randomization stay disabled.

In every setting, the 100 evaluation episodes of a task use the first 100 seeds, counting up from 100000, for which the expert plan succeeds. An episode counts as a success if the task predicate holds at any physics step within the task's step limit (Table IV).

TABLE IV: *RoboTwin-UMI* tasks and step limits (policy actions).

Task	Goal	Limit
Adjust Bottle	Pick up the bottle upright with the appropriate arm	400
Blocks Ranking RGB	Arrange red, green and blue blocks left to right	1200
Click Alarmclock	Press the alarm clock's top button	400
Click Bell	Press the top of the bell	400
Grab Roller	Grasp and lift the roller with both arms	400
Lift Pot	Lift the pot with both arms	400
Move Playingcard Away	Move the playing card away from the table center	400
Open Laptop	Open the laptop lid	1400
Place A2B Left	Place object A to the left of object B	400
Place A2B Right	Place object A to the right of object B	400
Place Bread Basket	Put one or two breads into the basket	700
Place Container Plate	Place the container onto the plate	400
Place Empty Cup	Place the empty cup on the coaster	500
Place Object Scale	Put the object on the scale	400
Place Object Stand	Place the object on the stand	400
Press Stapler	Press down the stapler	400
Shake Bottle	Lift and shake the bottle	700
Stack Blocks Two	Stack the green block on the red block	800
Stack Bowls Two	Stack the two bowls	900
Turn Switch	Toggle the switch	400

TABLE V: Real-world demonstrations and scored stages.

Task	Demos	Frames	Scored stages
Pick-and-Place	20	3,350	reach, grasp, place
Bimanual Handover	20	11,565	grasp, hand over, right-gripper release, place in box
Scan-and-Place	40	18,278	grasp product, grasp scanner, scan, put down scanner, place product
Pack-and-Close	40	13,805	grasp, place, close lid, press
Handover-and-Pour	30	11,103	reach, grasp, hand over, pour, release
Table Wiping	30	10,931	grasp, wipe, release, grasp, wipe, release
Total	180	69,032	

c) Tasks: Table IV lists the twenty *RoboTwin-UMI* tasks with step limits inherited from RoboTwin. They cover object placement (the seven Place tasks, Move Playingcard Away and Adjust Bottle), stacking and arrangement (Stack Blocks Two, Stack Bowls Two and Blocks Ranking RGB), articulated-object and button interaction (Open Laptop, Press Stapler, Turn Switch, Click Alarmclock and Click Bell), and dual-arm or dynamic manipulation (Grab Roller, Lift Pot and Shake Bottle). Each model is trained on 50 clean Aloha-AgileX demonstrations per task. For the six real-world tasks, Table V lists the demonstrations and binary stages. The progress score is the fraction of stages completed, and an episode succeeds only if all stages are completed.