X-WORLD: TRANSFERRING VIDEO PRIORS TO VARIABLE-VIEW ROBOT CONTROL

Chun Gu[1,2,3], **Xiaofei Wei**[1,2,3], **Yiming Chen**[2,3], **Juntong Zhang**[2,3], **Li Zhang**[1,2,3†]

[1]School of Data Science, Fudan University [2]Shanghai Innovation Institute [3]Logos Intelligence Lab

logosintelligencelab.github.io/X-World/

ABSTRACT

Video diffusion models provide a source of interaction representations for robot control, but transferring these representations requires reconciling variable camera inputs with heterogeneous action interfaces and closed-loop execution. We present *X-World*, a framework that transfers a video prior through a fixed-length token sequence to a shared action expert. A residual cross-view attention branch extends a pretrained video backbone to one, two, or three views while retaining its original self-attention path. A layer-aligned Q-Former aggregates backbone features, and dataset-specific modules map between native state and action interfaces and the shared action expert. Training separates video adaptation from action-only pretraining by freezing the video backbone in the latter stage, followed by joint target-domain post-training. The policy extracts video features with a single backbone pass and uses asynchronous action chunking for execution. On LIBERO, *X-World* reaches 98.5% mean success; its LIBERO-trained checkpoint obtains 76.5% on LIBERO-Plus without further adaptation. The seven-shift breakdown reveals gains in robot, language, lighting, and layout perturbations, alongside weaknesses under camera, background, and noise shifts relative to OpenVLA-OFT. On six dual-arm real-world tasks, it achieves 81.7% mean success over 20 trials per task. Removing video adaptation or cross-view fusion produces the largest observed success drops among the tested removals.

1 INTRODUCTION

Video data and robot action data offer different forms of supervision. Videos capture object motion, contact, and task progression across many scenes, whereas action-labeled trajectories specify how a particular robot produces those transitions. Vision-language-action policies (Brohan et al., 2022, 2023; Kim et al., 2024; Octo Model Team et al., 2024; Bjorck et al., 2025; Black et al., 2024; Physical Intelligence et al., 2025) learn the latter mapping effectively, but collecting action supervision across robots and camera setups remains expensive. Video diffusion models offer a complementary source of representations (Ye et al., 2024; Du et al., 2023b; Hu et al., 2024). The question we study is how to transfer these representations to a policy whose observations and executable actions vary across datasets.

This transfer has two interfaces. At the visual interface, datasets provide different numbers of cameras and different viewpoints. A single-view video backbone must gain access to information across views without discarding its pretrained computation. At the control interface, video hidden states contain many spatiotemporal tokens, while the policy must generate a short action chunk in the robot's native coordinates. Exposing a changing visual context directly to an action expert couples these two forms of heterogeneity. We instead use a fixed-length token sequence between the video backbone and the action expert, so camera processing and action decoding can be adapted separately.

X-World instantiates this design with a pretrained LTX-Video backbone (HaCohen et al., 2024), a layer-aligned Q-Former, and a flow-matching action expert. The video backbone retains its pretrained self-attention and adds cross-view attention through a zero-initialized residual branch. View embeddings, randomized view order, and masked replication support training on one-, two-, and

†Corresponding author (lizhangfd@fudan.edu.cn).

three-view datasets. The Q-Former aggregates features across backbone layers into the token sequence consumed by the action expert. Following the use of dataset-specific interfaces in cross-embodiment learning (Zheng et al., 2026), lightweight projections map native state and action representations to and from the shared action backbone.

Training first adapts the video backbone on robot and human videos, then freezes it while learning the Q-Former and action expert from heterogeneous action-labeled trajectories. Target-domain post-training updates the model with video and action supervision. At deployment, one backbone pass extracts features from observed image latents and noisy future slots; the policy does not generate and decode a future video before acting. Asynchronous real-time chunking (Black et al., 2025a,b) overlaps policy queries with execution. On a single H100, the reported cached model response is 46 ms; the deployed refresh, including communication and processing, is 91 ms, while queued actions execute at 30 Hz.

Our evaluation asks three questions: whether the transferred representation supports manipulation, which parts of the transfer account for the observed performance, and where robustness breaks down. The policy reaches 98.5% mean success on LIBERO (Liu et al., 2023) and 76.5% on zero-shot LIBERO-Plus (Fei et al., 2025). Because LIBERO is nearly saturated, we examine the full distribution-shift profile and the training-stage interaction rather than relying on its average ranking. On six dual-arm UR5e tasks, *X-World* achieves 81.7% mean success, compared with 58.3% for X-VLA, the strongest baseline in this real-world comparison. Removing cross-view attention reduces LIBERO success to 83.1% and Fold Towel success from 12/20 to 0/20, providing evidence beyond the saturated benchmark average.

This paper makes three main contributions: **(i)** We introduce ***X-World***, a framework for variable-view robot control that adapts a pretrained video backbone through masked residual cross-view fusion while retaining its original self-attention path; **(ii)** We build a video-to-action interface that combines a layer-aligned Q-Former with dataset-specific state and action projections, using fixed-length tokens and separating video adaptation from action-only pretraining; and **(iii)** We analyze the benefits and limits of this transfer through training-stage ablations, seven distribution shifts, and dual-arm experiments, with separate measurements of model and system latency.

2 Related work

From video models to robot actions. Video-action models differ in how they expose visual dynamics to control. One approach generates future observations and maps predicted transitions to actions through inverse dynamics (Du et al., 2023b,a; Wen et al., 2024). Another uses internal video representations as features for action generation (Hu et al., 2024; Yang et al., 2025). Joint video-action architectures couple the two modalities within a transformer (Bi et al., 2025; Liao et al., 2025; Ye et al., 2026; Li et al., 2026; Pai et al., 2025), while Cosmos Policy (Kim et al., 2026) encodes actions within a video generation formulation. These works establish video-to-control transfer as an existing direction. *X-World* studies a particular interface within that direction: layer-aligned aggregation into a fixed-length token sequence, combined with a video backbone adapted to variable-view inputs. At control time, it uses hidden states from one backbone pass rather than a decoded future rollout.

Multi-view adaptation. Genie-Envisioner (Liao et al., 2025) builds multi-view video modeling on LTX-Video and provides a close reference for transferring video features to actions. Our architectural distinction is to retain per-view pretrained self-attention and append cross-view fusion as a residual branch. The added branch contributes zero at initialization while the original per-view computation is retained. We evaluate residual fusion and view identity separately, and distinguish missing-view evaluation from robustness to camera-pose perturbations.

Heterogeneous action learning and continuous execution. Open-X-Embodiment (O'Neill et al., 2024) and other multi-platform datasets (Bu et al., 2025a; Wu et al., 2025; Jiang et al., 2025; Wu et al., 2024) enable training across robot interfaces. Alignment layers, modular projections, and embodiment representations address differences in state and action conventions (Doshi et al., 2024; Wang et al., 2024; Zheng et al., 2025, 2026). Other methods address observation differences through feature alignment (Bian et al., 2025), novel-view synthesis, or embodiment editing (Chen et al.,

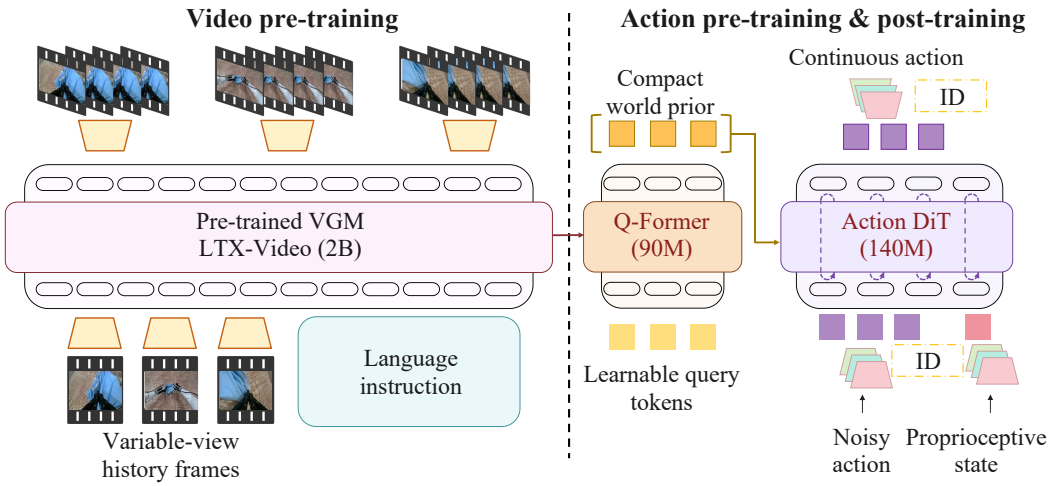


Figure 1: Overview of *X-World*. The video backbone is first adapted on robot and human videos with one to three camera views (left). A layer-aligned Q-Former then compresses its features into fixed-length tokens that condition the action DiT, while dataset-specific projections map native states and actions to and from the shared expert (right). Action pretraining learns this interface with the video backbone frozen; target-domain post-training updates all modules using video and action supervision.

2024b,a; Ji et al., 2025). We use dataset-specific input and output modules around a shared action expert. For deployment, we adopt asynchronous chunking inspired by RTC (Black et al., 2025a,b); our evaluation reports the latency budget of the resulting system separately from task success.

The design focus is therefore the residual variable-view video representation and its layer-aligned token interface. Dataset-specific projections and asynchronous execution build on established approaches; our comparisons evaluate their role within this complete policy.

3 Method

3.1 Problem and design

At control step t, the policy receives RGB observations $\mathcal{I}_t = \{I_t^{(v)}\}_{v=1}^{V_t}$, an instruction l, and proprioceptive state s_t. It predicts an action chunk $\mathbf{a}_{t:t+H-1} \in \mathbb{R}^{H \times d_a}$, with $H = 16$ and a dataset-dependent action dimension d_a. Our design separates a variable-view visual encoder from native action decoding through a fixed-length token sequence (Fig. 1). The video prediction horizon used for adaptation is $K = 24$ future frames at 256×256 resolution; it is distinct from the executable action horizon H.

3.2 Variable-view video representations

We initialize the visual backbone from LTX-Video (HaCohen et al., 2024). Its video VAE compresses inputs by $16\times$ spatially and $8\times$ temporally. A single observed frame per available camera, together with the instruction, provides the context for learning future-video representations.

Residual cross-view fusion. We retain the pretrained per-view self-attention path and append cross-view attention through an added residual branch (Fig. 2). Writing u_ℓ for the features on that path, the cross-view update has the form

$$\widetilde{u}_\ell = u_\ell + C_\ell(u_\ell; M), \qquad C_\ell(\cdot; M) = 0 \text{ at initialization}, \tag{1}$$

where M masks disallowed interactions between view slots. The zero initialization makes the added cross-view update vanish initially. View embeddings encode view identity, and training randomizes the view order. These operations support heterogeneous camera inputs without replacing the original self-attention with a joint attention operation over every view.

View slots and time coordinates. Training uses three view slots. For a sample with one or two cameras, an existing view is replicated into the remaining slots and independent diffusion noise is

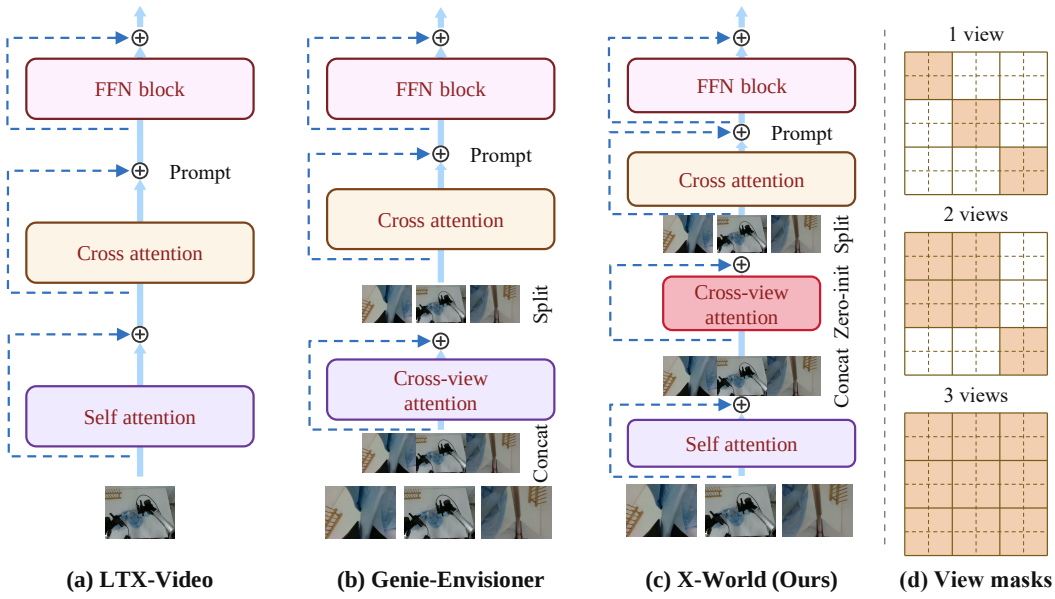


Figure 2: Cross-view adaptation of the video transformer. (a) LTX-Video (HaCohen et al., 2024) applies self-attention to a single view. (b) Genie-Envisioner (Liao et al., 2025) fuses concatenated views through cross-view attention. (c) *X-World* retains pretrained per-view self-attention and learns cross-view fusion through a separate residual branch. Zero initialization preserves the pretrained computation at the start of adaptation while allowing cross-camera interactions to be learned. (d) Training-time masks for one, two, and three available views. Shaded entries allow attention; replicated slots are isolated from available views, supporting joint training across camera counts.

applied to the replicas. The cross-view mask blocks interactions between original and replicated slots in both directions. The replicas therefore supply additional noisy video inputs without introducing cross-view messages into the original slots. Per-sample frame rates scale temporal RoPE coordinates (Su et al., 2024); the exact interpolation rule is given in Sec. B.2. The evaluated setting supports one to three views, rather than an unbounded number of cameras.

Layer-aligned token aggregation. Let $\{h_\ell\}_{\ell=1}^{L}$ be the exported hidden states of the video transformer's L blocks. A Q-Former refines learnable queries through a corresponding stack of cross-attention blocks:

$$z_\ell = Q_\ell(z_{\ell-1}, h_\ell), \qquad z = z_L. \tag{2}$$

Here Q_ℓ denotes the query-update block attending to video features h_ℓ. The final token sequence z has fixed length, independent of the number of video tokens or available views. This fixes the context length seen by the action expert; the cost of computing the video features and Q-Former can still depend on the input.

3.3 Shared Action Expert and Training

The action expert is a DiT-style transformer (Peebles & Xie, 2023). It applies self-attention to state and action tokens and cross-attention to z. Following the use of dataset-specific interfaces in X-VLA (Zheng et al., 2026), each dataset has a state projector, an input projection for noisy actions, and an output head. The shared transformer operates between these modules, while each output head decodes actions in its dataset's native parameterization.

Stage 1: Video adaptation. We adapt the video backbone on robot and human videos. For future-video latents $\mathbf{y}$, noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$, and scheduler-sampled noise level σ, we set $\mathbf{y}_\sigma = (1-\sigma)\mathbf{y} + \sigma\boldsymbol{\epsilon}$ and optimize

$$\mathcal{L}_{\text{vid}} = \mathbb{E}\left[\left\|v_\phi(\mathbf{y}_\sigma, \sigma, \mathcal{I}_t, l) - (\boldsymbol{\epsilon} - \mathbf{y})\right\|_2^2\right]. \tag{3}$$

Table 1: Three-stage training of *X-World*. Stage 1 adapts the video backbone and view modules through video prediction on robot and human data. Stage 2 freezes the backbone and trains the Q-Former, action expert, and native interfaces on action-labeled trajectories. Stage 3 jointly optimizes video and action objectives on LIBERO or real-world demonstrations, updating all modules.

Stage	Data	Loss	Updated modules
1: Video adaptation	Robot/human videos	$\mathcal{L}_{\text{vid}}$	Video backbone and view modules
2: Action pretraining	Robot trajectories	$\mathcal{L}_{\text{act}}$	Q-Former, action expert, native projections
3: Post-training	Target trajectories	$\mathcal{L}_{\text{vid}} + \mathcal{L}_{\text{act}}$	All modules

Caption dropout replaces some instructions with an empty-prompt embedding for classifier-free guidance. The training mixture includes the robot and human video sources listed in Table 6. The recorded training budget is seven days on 64 H100 GPUs.

Stage 2: Action-only pretraining. We freeze the video backbone and update the Q-Former, action expert, and dataset-specific modules with action supervision. Dataset d is sampled with probability proportional to $N_d^{1/2}$, where N_d is its number of sampleable transitions; all base mixture weights are one. For a ground-truth action chunk $\mathbf{a}$, noise $\boldsymbol{\epsilon}_a$, and noise level σ_a, the flow-matching objective is

$$\begin{aligned} \mathbf{a}' &= (1-\sigma_a)\mathbf{a} + \sigma_a \boldsymbol{\epsilon}_a, \\ \mathcal{L}_{\text{act}} &= \mathbb{E}\left[\left\| v_\theta(\mathbf{a}', \sigma_a, z, s_t) - (\boldsymbol{\epsilon}_a - \mathbf{a}) \right\|_2^2\right]. \end{aligned} \tag{4}$$

Freezing prevents action gradients from updating the video backbone during this stage. The data sources are listed in Table 7; the recorded budget is three days on 64 H100 GPUs.

Stage 3: Target-domain post-training. We post-train on LIBERO demonstrations for simulation or on collected teleoperation trajectories for the real-world platform. All model modules are updated with the joint objective $\mathcal{L}_{\text{post}} = \mathcal{L}_{\text{vid}} + \mathcal{L}_{\text{act}}$. This stage adapts both the visual representation and action generation to the target domain.

3.4 Feature extraction and asynchronous execution

At control time, we VAE-encode the current views, fix the observed-frame latents, and initialize the unobserved future slots with noise. One LTX forward pass at the highest-noise endpoint ($\sigma = 1$, diffusion timestep 1000) exports features, with observed-frame tokens assigned timestep zero. The Q-Former and a four-step flow action expert then predict the action chunk. No iterative video denoising or video decoding lies on this control path.

Asynchronous real-time chunking (Black et al., 2025a,b) overlaps a policy query with execution of committed actions. A FIFO queue $\mathcal{Q}$ stores these actions. When its remaining length $R = |\mathcal{Q}|$ reaches the threshold P, the policy reuses the remaining actions as a committed prefix and predicts a refreshed chunk. After dropping the overlapping prefix, it appends up to E newly predicted suffix actions. Here E denotes the appended suffix length, not the prefix length. For control frequency f_c and refresh latency T_{refresh}, the nominal deadline condition is

$$T_{\text{refresh}} < P/f_c. \tag{5}$$

With $P = 6$ and $f_c = 30\,\text{Hz}$, the prefix covers 200 ms. This budget must cover the actual latency of each query for uninterrupted execution.

Training exposes the action expert to committed prefixes by sampling $\delta \sim \text{Uniform}(\{0, \dots, 8\})$. The first δ action positions remain noise-free and the suffix is noised:

$$\mathbf{a}'_i = (1-\sigma_{a,i})\mathbf{a}_i + \sigma_{a,i}\boldsymbol{\epsilon}_{a,i}, \qquad \sigma_{a,i} = \begin{cases} 0, & i \le \delta, \\ \sigma_a, & i > \delta. \end{cases} \tag{6}$$

Thus the action expert receives a known prefix during both training and deployment.

Table 2: LIBERO success rates (%) on the Spatial, Object, Goal, and Long suites. Average is the mean of the four suite scores. Best results are bold with pink backgrounds; second-best results are underlined with lavender backgrounds.

Method	Spatial↑	Object↑	Goal↑	Long↑	Average
Diffusion Policy (Chi et al., 2025)	78.5	87.5	73.5	64.8	76.1
OpenVLA (Kim et al., 2024)	84.7	88.4	79.2	53.7	76.5
SpatialVLA (Qu et al., 2025)	88.2	89.9	78.6	55.5	78.1
CoT-VLA (Zhao et al., 2025a)	87.5	91.6	87.6	69.0	83.9
π_0-Fast (Pertsch et al., 2025)	96.4	96.8	88.6	60.2	85.5
GR00T-N1 (Bjorck et al., 2025)	94.4	97.6	93.0	90.6	93.9
π_0 (Black et al., 2024)	98.0	96.8	94.4	88.4	94.4
F1 (Lv et al., 2025)	98.2	97.8	95.4	91.3	95.7
InternVLA-M1 (InternVLA-M1 Contributors, 2025)	98.0	<u>99.0</u>	93.8	92.6	95.9
Discrete Diffusion VLA (Liang et al., 2025)	97.2	98.6	97.4	92.0	96.3
Genie-Envisioner (Liao et al., 2025)	98.2	97.6	95.8	94.4	96.5
$\pi_{0.5}$ (Physical Intelligence et al., 2025)	<u>98.8</u>	98.2	<u>98.0</u>	92.4	96.9
GR00T-N1.6 (GEAR Team et al., 2025)	97.7	98.5	97.5	94.4	97.0
OpenVLA-OFT (Kim et al., 2025)	97.6	98.4	97.9	94.5	97.1
X-VLA (Zheng et al., 2026)	98.2	98.6	97.8	**97.6**	<u>98.1</u>
X-World	**99.6**	**99.4**	**99.0**	<u>95.8</u>	**98.5**

4 Experiments

We evaluate task performance on LIBERO and LIBERO-Plus, then present the real-world platform, tasks, and rollouts before analyzing the training and deployment choices. Future-video prediction examples are provided in Sec. C.

4.1 Simulation results

LIBERO. We post-train one unified policy on demonstrations from all four LIBERO suites (Liu et al., 2023). Inputs are the language instruction, synchronized third-person and wrist RGB images, and proprioceptive state. Evaluation uses 50 episodes for each of 10 tasks per suite, giving 500 rollouts per suite. Table 2 reports the complete reference comparison (Chi et al., 2025; Kim et al., 2024; Qu et al., 2025; Zhao et al., 2025a; Pertsch et al., 2025; Bjorck et al., 2025; Black et al., 2024; Lv et al., 2025; InternVLA-M1 Contributors, 2025; Liang et al., 2025; Liao et al., 2025; Physical Intelligence et al., 2025; Kim et al., 2025; Zheng et al., 2026). These reference results use different pretraining data and compute budgets.

X-World reaches 98.5% mean success. Its largest numerical advantage over X-VLA is on Spatial (99.6% versus 98.2%), whereas X-VLA performs better on Long (97.6% versus 95.8%). The 0.4-point average gap is small at this saturation level, motivating the distribution-shift evaluation and within-framework ablations below.

Zero-shot LIBERO-Plus. We evaluate the LIBERO-post-trained checkpoint on LIBERO-Plus (Fei et al., 2025) without further adaptation. Table 3 gives the full seven-shift comparison (Kim et al., 2024; Black et al., 2024; Kim et al., 2025; Hung et al., 2025; Cen et al., 2025; Bu et al., 2025b; Pertsch et al., 2025; Tan et al., 2025). The reported total is 76.5%, compared with 69.6% for OpenVLA-OFT. The largest category difference is under robot perturbations (72.5% versus 31.9%); language and layout also improve by 14.8 and 10.5 points. Camera, background, and noise success are lower by 3.8, 7.5, and 16.1 points, respectively. The aggregate gain therefore reflects selective robustness: accommodating different camera counts does not by itself confer invariance to changed camera poses or corrupted images.

Table 3: Zero-shot success rates (%) across seven LIBERO-Plus distribution shifts (Fei et al., 2025). Cam., Lang., and Bg. denote camera, language, and background changes, respectively. The w and m variants use no wrist input and mixed-data supervised fine-tuning, respectively. OpenVLA-OFT+ is excluded because it is trained on LIBERO-Plus. Best results are bold with pink backgrounds; second-best results are underlined with lavender backgrounds.

Method	Cam.	Robot	Lang.	Light	Bg.	Noise	Layout	Total
OpenVLA (Kim et al., 2024)	0.8	3.5	23.0	8.1	34.8	15.2	28.5	15.6
π_0 (Black et al., 2024)	13.8	6.0	58.8	85.0	81.4	**79.0**	68.9	53.6
OpenVLA-OFT (Kim et al., 2025)	56.4	31.9	79.5	88.7	93.3	75.8	74.2	69.6
OpenVLA-OFT$_{\text{w}}$ (Kim et al., 2025)	10.4	38.7	70.5	76.8	**93.6**	49.9	69.9	55.8
NORA (Hung et al., 2025)	2.2	37.0	65.1	45.7	58.6	12.8	62.1	39.0
WorldVLA (Cen et al., 2025)	0.1	27.9	41.6	43.7	17.1	10.9	38.0	25.0
UniVLA (Bu et al., 2025b)	1.8	46.2	69.6	69.0	81.0	21.2	31.9	43.9
π_0-Fast (Pertsch et al., 2025)	**65.1**	21.6	61.0	73.2	73.2	74.4	68.8	61.6
RIPT-VLA (Tan et al., 2025)	55.2	31.2	77.6	88.4	91.6	73.5	74.2	68.4
OpenVLA-OFT$_{\text{m}}$ (Kim et al., 2025)	55.6	21.7	81.0	92.7	91.0	78.6	68.7	67.9
X-World	52.6	**72.5**	**94.3**	**95.4**	85.8	59.7	**84.7**	**76.5**

4.2 Real-world setup and tasks

Platform and scene. We evaluate *X-World* on the dual-arm platform in Fig. 3. Two wrist cameras and a shared third-person camera provide synchronized RGB observations of the tabletop workspace.

Data and evaluation protocol. Each task provides 30 minutes of VR teleoperation demonstrations with synchronized three-view observations. Data recording and action execution run at 30 Hz. Actions are relative joint position targets, and absolute joint positions form the proprioceptive input. Each policy is evaluated on 20 trials per task using the same 20 predefined initial placements, camera views, control frequency, and task definitions.

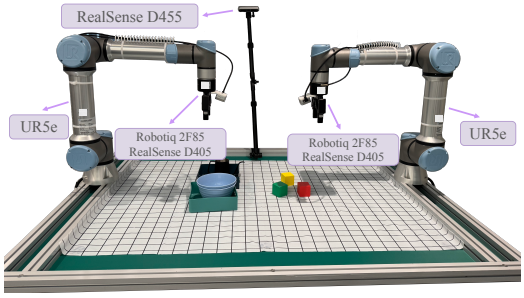


Figure 3: Dual-arm evaluation platform. Two UR5e arms use Robotiq 2F85 grippers and wrist-mounted RealSense D405 cameras; a shared D455 camera supplies the third-person view.

The six evaluated tasks require sequential or simultaneous coordination:

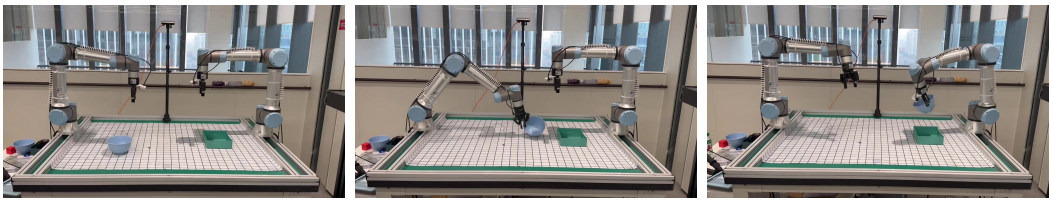
- **Stack bowl and block.** Stack a side bowl onto a center bowl, then place a block into it.
- **Put bottle into trash can.** Pick up scattered bottles and place them in the trash can.
- **Pour water from teapot.** Pour water into a cup, then place the teapot and cup on a tray.
- **Move bowl into a tray.** Move a bowl to the center with one arm, then transfer it into a tray with the other.
- **Stack two blocks.** Move blocks from the table to a tray and stack them.
- **Clear the table.** Place all objects into a drawer.

Qualitative execution. Figure 4 presents representative real-world rollouts, with three ordered snapshots per row. The instruction above each row identifies that example. These selected rollouts illustrate the workspace and manipulation sequence; the success rates below are computed from the evaluation trials. The supplementary package provides six task videos (Sec. D).

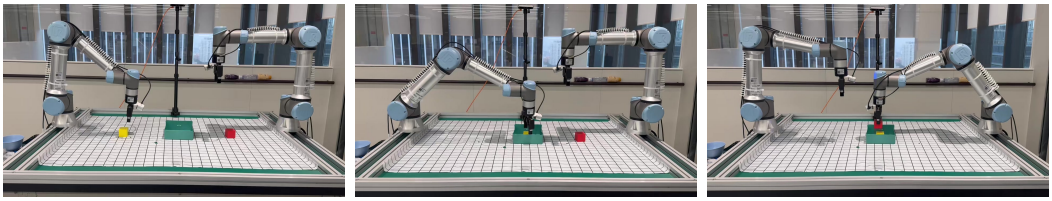
4.3 Real-world results

Figure 5 reports the per-task success rates for *X-World* and the three baseline policies (Liao et al., 2025; Physical Intelligence et al., 2025; Zheng et al., 2026). *X-World* achieves the highest point estimate on each task, with success rates between 75% and 95%. Across six tasks it succeeds in 98 of 120 trials, giving an 81.7% task mean. X-VLA achieves 70/120 (58.3%), Genie-Envisioner 62/120

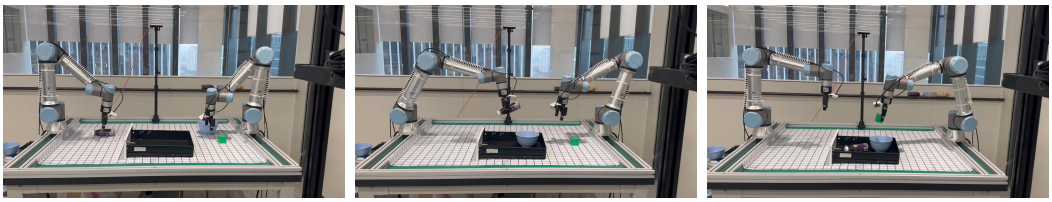
"Move the bowl to the center, then into the drawer."



"Place the yellow block in the center, then stack the red block on top."



"Put all the items on the table into the drawer."



"Place the yellow block in the center, then stack the red block on top."

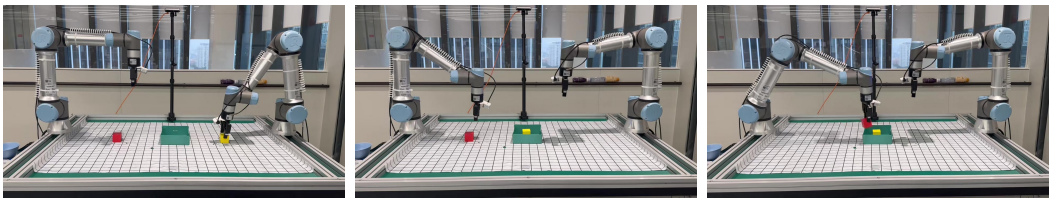


Figure 4: Real-world manipulation examples. Each row corresponds to a separate rollout of the language instruction above it, with three third-person snapshots arranged in execution order from left to right.

(51.7%), and $\pi_{0.5}$ 54/120 (45.0%). The comparison is based on 20 trials per task; the available aggregate counts do not determine training-seed variability or paired trial outcomes.

4.4 Ablation studies

Training stages. We compare the four combinations of video adaptation and action pretraining, with target-domain post-training and the LIBERO evaluation protocol held fixed. With video adaptation present, action pretraining increases mean success from 97.4% to 98.5%, and Long success from 94.2% to 95.8%. Without video adaptation, action pretraining instead lowers mean success from 92.1% to 89.5%, and Long success from 81.2% to 72.4%. The two stages therefore do not behave as independent, additive improvements in these results. The comparison is consistent with an interaction between visual adaptation and action pretraining, but does not identify the cause of the reversal.

Q-Former and native action projections. Replacing the Q-Former with direct access to last-layer video features gives 97.9% mean success, compared with 98.5% for the full model. This comparison changes both feature aggregation and compression. The Q-Former takes 5 ms in the full-model profile; a matched latency measurement for the direct-feature variant is unavailable. Shared state and action projections yield 97.4% mean success, 1.1 points below the dataset-specific design.

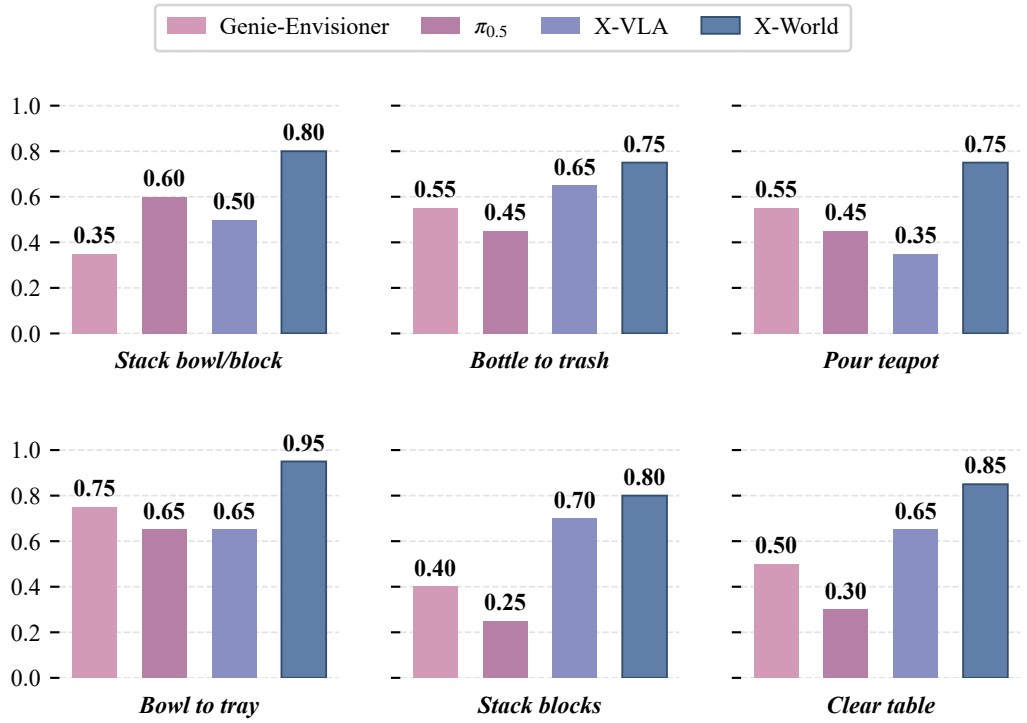


Figure 5: Real-world task success. Bars show success fractions over 20 trials per policy and task on six dual-arm UR5e tasks under a shared evaluation protocol and identical initial placements. *X-World* achieves the highest observed success on every task.

Table 4: LIBERO success rates (%) for ablations of training stages, model components, and video loss. Stage 1 is video adaptation and Stage 2 is action pretraining; Average is the mean across the four suites. Best results are bold with pink backgrounds; second-best results are underlined with lavender backgrounds.

Setting	Spatial↑	Object↑	Goal↑	Long↑	Average
w/o Stage 1	95.4	97.6	92.4	72.4	89.5
w/o Stage 2	99.2	98.8	97.2	94.2	97.4
w/o Stage 1 and Stage 2	97.4	96.2	93.4	81.2	92.1
Shared projections	98.6	98.4	98.2	94.4	97.4
w/o Q-Former	**99.8**	98.6	<u>98.6</u>	94.6	97.9
w/o video loss	**99.8**	**100.0**	97.2	**96.2**	<u>98.3</u>
Full model	<u>99.6</u>	<u>99.4</u>	**99.0**	<u>95.8</u>	**98.5**

Video loss during post-training. Removing the video loss gives 98.3% mean success, including a higher Long score (96.2% versus 95.8%). This small average difference provides limited evidence for a control benefit from the joint video objective.

Cross-view components. Removing residual cross-view attention lowers LIBERO success from 98.5% to 83.1%; removing view embeddings lowers it to 93.3%. We additionally evaluate *Fold Towel*, which requires flattening a cluttered towel and then folding it. The full model succeeds in 12/20 trials, compared with 0/20 without residual cross-view attention and 2/20 without view embeddings. These component removals are distinct from the test-time view-availability analysis below.

4.5 View availability and real-time execution

Available views. We use the same model checkpoint for all view-availability settings. One-view evaluation retains the third-person camera; two-view evaluation removes one wrist camera; the shuffled setting permutes the three-camera input order. The reported success rates are 40%, 55%, 80%,

Table 5: Single-H100 inference latency (ms). Component timings exclude 14 ms of model-side overhead included in the 46 ms cached response; T5 encoding adds 13 ms on the first call. The 91 ms deployed refresh additionally includes network transfer, data processing, and control-side overhead.

Model timing		Deployment timing	
T5 encoder (first call)	13	Cached model response	46
VAE	7	Network transfer	32
Video diffusion	12	Data processing	7
Q-Former	5	Control-side overhead	6
Action expert	8	End-to-end refresh	91

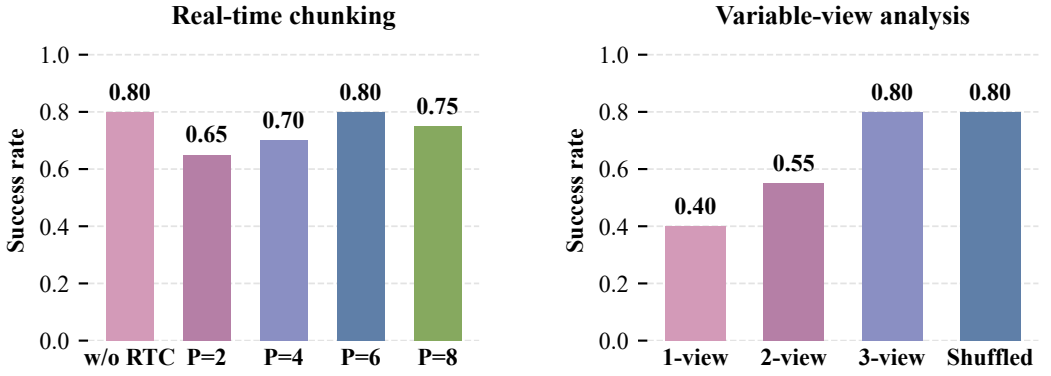


Figure 6: Action-prefix and view-availability analyses. The left panel compares inference without real-time chunking (w/o RTC) with prefix lengths $P = 2, 4, 6, 8$. The right panel uses the same checkpoint to compare one, two, and three camera views and a shuffled three-view order. Bars report success fractions.

and 80% for one, two, three, and shuffled views, respectively (Fig. 6, right). Additional views improve the point estimate, while the tested order permutation leaves it unchanged. The aggregate sweep records do not specify task selection, trial denominators, or seed variation, so these results are descriptive.

Action-prefix sweep. Success is 80% without RTC and 65%, 70%, 80%, and 75% for $P = 2, 4, 6, 8$ (Fig. 6, left). The reported point estimates favor $P = 6$ among the tested prefixes. Success alone does not measure execution continuity, as the no-RTC setting also reaches 80%.

Runtime breakdown. Table 5 reports the module and deployment timing breakdown for the single-H100 system with $H = 16$. The VAE, video backbone, Q-Former, and action expert sum to 32 ms. Including 14 ms of model-side overhead gives a 46 ms cached response; the first call adds 13 ms of T5 encoding. Network transfer, data processing, and control-side overhead raise the deployed refresh to 91 ms. These timings differ from the 30 Hz clock that executes queued actions.

At 30 Hz, prefixes of two, four, six, and eight actions cover approximately 67, 133, 200, and 267 ms. Only the first is below the reported 91 ms refresh. With $P = 6$, the committed prefix provides a nominal margin of 109 ms. This budget explains the deployment choice without establishing a bound on every query or an observed queue-underflow rate.

5 Discussion and Conclusion

X-World transfers video representations to robot control through residual multi-view adaptation and a fixed-length token interface. Removing video adaptation or cross-view fusion produces the largest observed success drops among the tested removals. The training-stage comparison also exposes an interaction: action pretraining helps after video adaptation, but reduces performance when adaptation is skipped. The distribution-shift results are similarly selective, with gains on robot and language perturbations accompanied by losses on camera and noise shifts. Together, these findings motivate treating the visual and action interfaces as separate design choices when transferring a video prior.

Limitations. Evaluation covers LIBERO, its distribution-shift extension, and one real-world robot platform; it does not establish transfer to an unseen embodiment or a scaling law. The training-stage ablations change the available supervision as well as the training procedure, so they do not isolate the benefit of freezing the backbone. The Q-Former comparison uses last-layer features, confounding compression with access to multiple layers; its 0.6-point average difference requires repeated runs and a matched feature baseline before drawing a stronger conclusion. Baseline pretraining data and compute are not controlled, and the available results do not quantify training-seed variability. The LIBERO-Plus category denominators and aggregate weighting are unspecified, limiting audit of the reported total score. Real-world success estimates are based on 20 trials per task. The latency profile provides a nominal queue budget; tail latency, queue underflow, and action-boundary continuity remain unquantified.

ETHICS STATEMENT

This work uses public human and robot video datasets and self-collected robot teleoperation data for research on language-instructed manipulation. Public data are used according to their licenses. Teleoperation recordings are handled as potentially identifying data, and any redistribution should preserve the applicable consent and privacy requirements. Physical robot experiments require supervised operation, workspace safeguards, and an emergency stop; success in simulation or a qualitative rollout does not establish general hardware safety.

REPRODUCIBILITY STATEMENT

The paper reports the training objectives, updated modules, input resolution, prediction horizons, feature-extraction path, and evaluation settings. The Appendix lists data sources, frame-rate interpolation, native action representations, and the available aggregate results. Figure scripts use those aggregates and the original qualitative frames. Full model dimensions, optimization configurations, post-training budgets, and per-run evaluation logs are not supplied here; these omissions limit independent reproduction and uncertainty analysis.

REFERENCES

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. *arXiv preprint*, 2025.

Jinyue Bian, Zhaoxing Zhang, Zhengyu Liang, Shiwei Zheng, Shengtao Zhang, Rong Shen, Chen Yang, and Anzhou Hou. VLA-LPAF: Lightweight perspective-adaptive fusion for vision-language-action to enable more unconstrained robotic manipulation. *arXiv preprint*, 2025.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint*, 2024.

Kevin Black, Manuel Y Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. *arXiv preprint*, 2025a.

Kevin Black, Allen Z Ren, Michael Equi, and Sergey Levine. Training-time action conditioning for efficient real-time chunking. *arXiv preprint*, 2025b.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. RT-1: Robotics transformer for real-world control at scale. *arXiv preprint*, 2022.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint*, 2023.

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*, 2025a.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. UniVLA: Learning to act anywhere with task-centric latent actions. *arXiv preprint*, 2025b.

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

Lawrence Yunliang Chen, Kush Hari, Karthik Dharmarajan, Chenfeng Xu, Quan Vuong, and Ken Goldberg. Mirage: Cross-embodiment zero-shot policy transfer with cross-painting. *arXiv preprint*, 2024a.

Lawrence Yunliang Chen, Chenfeng Xu, Karthik Dharmarajan, Muhammad Zubair Irshad, Richard Cheng, Kurt Keutzer, Masayoshi Tomizuka, Quan Vuong, and Ken Goldberg. RoVi-Aug: Robot and viewpoint augmentation for cross-embodiment robot learning. *arXiv preprint*, 2024b.

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*, 2025.

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

Ria Doshi, Homer Walke, Oier Mees, Sudeep Dasari, and Sergey Levine. Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation. *arXiv preprint*, 2024.

Yilun Du, Mengjiao Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Brian Ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B Tenenbaum, et al. Video language planning. *arXiv preprint*, 2023a.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *NeurIPS*, 2023b.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. LIBERO-Plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint*, 2025.

GEAR Team, Allison Azzolini, Johan Bjorck, Valts Blukis, Fernando Castañeda, Rahul Chand, Yan Chang, Danyi Chen, Nikita Cherniadev, Xingye Da, et al. GR00T N1.6: An improved open foundation model for generalist humanoid robots. *NVIDIA Research*, December 2025.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4D: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. LTX-Video: Realtime video latent diffusion. *arXiv preprint*, 2024.

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*, 2024.

Chia-Yu Hung, Qi Sun, Pengfei Hong, Amir Zadeh, Chuan Li, U-Xuan Tan, Navonil Majumder, and Soujanya Poria. NORA: A small open-sourced generalist vision language action model for embodied tasks. *arXiv preprint*, 2025.

InternVLA-M1 Contributors. InternVLA-M1: A spatially guided vision-language-action framework for generalist robot policy. *arXiv preprint*, 2025.

Guanhua Ji, Harsha Polavaram, Lawrence Yunliang Chen, Sandeep Bajamahal, Zehan Ma, Simeon Adebola, Chenfeng Xu, and Ken Goldberg. OXE-AugE: A large-scale robot augmentation of OXE for scaling cross-embodiment policy learning. *arXiv preprint*, 2025.

Tao Jiang, Tianyuan Yuan, Yicheng Liu, Chenhao Lu, Jianning Cui, Xiao Liu, Shuiqi Cheng, Jiyang Gao, Huazhe Xu, and Hang Zhao. Galaxea open-world dataset and G0 dual-system VLA model. *arXiv preprint*, 2025.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. OpenVLA: An open-source vision-language-action model. *arXiv preprint*, 2024.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *RSS*, 2025.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, et al. Cosmos Policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint*, 2026.

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Wensi Ai, Benjamin Martinez, et al. BEHAVIOR-1K: A human-centered, embodied AI benchmark with 1,000 everyday activities and realistic simulation. *arXiv preprint*, 2024.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, et al. Causal world modeling for robot control. *arXiv preprint*, 2026.

Zhixuan Liang, Yizhuo Li, Tianshuo Yang, Chengyue Wu, Sitong Mao, Tian Nian, Liuao Pei, Shunbo Zhou, Xiaokang Yang, Jiangmiao Pang, et al. Discrete diffusion VLA: Bringing discrete diffusion to action decoding in vision-language-action policies. *arXiv preprint*, 2025.

Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Jingbin Cai, Si Liu, Jianlan Luo, et al. Genie Envisioner: A unified world foundation platform for robotic manipulation. *arXiv preprint*, 2025.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *NeurIPS*, 2023.

Qi Lv, Weijie Kong, Hao Li, Jia Zeng, Zherui Qiu, Delin Qu, Haoming Song, Qizhi Chen, Xiang Deng, and Jiangmiao Pang. F1: A vision-language-action model bridging understanding and generation to actions. *arXiv preprint*, 2025.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. CALVIN: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 2022.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint*, 2024.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open X-Embodiment: Robotic learning datasets and RT-X models. In *ICRA*, 2024.

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

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

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*, 2025.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint*, 2025.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. SpatialVLA: Exploring spatial representations for visual-language-action model. In *RSS*, 2025.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Shuhan Tan, Kairan Dou, Yue Zhao, and Philipp Krähenbühl. Interactive post-training for vision-language-action models. *arXiv preprint*, 2025.

Lirui Wang, Xinlei Chen, Jialiang Zhao, and Kaiming He. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers. In *NeurIPS*, 2024.

Youpeng Wen, Junfan Lin, Yi Zhu, Jianhua Han, Hang Xu, Shen Zhao, and Xiaodan Liang. VidMan: Exploiting implicit dynamics from video diffusion model for effective robot manipulation. In *NeurIPS*, 2024.

Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, et al. RoboMIND: Benchmark on multi-embodiment intelligence normative data for robot manipulation. *arXiv preprint*, 2024.

Shihan Wu, Xuecheng Liu, Shaoxuan Xie, Pengwei Wang, Xinghang Li, Bowen Yang, Zhe Li, Kai Zhu, Hongyu Wu, Yiheng Liu, et al. RoboCOIN: An open-sourced bimanual robotic data collection for integrated manipulation. *arXiv preprint*, 2025.

Liudi Yang, Yang Bai, George Eskandar, Fengyi Shen, Mohammad Altillawi, Dong Chen, Ziyuan Liu, and Abhinav Valada. CoVAR: Co-generation of video and action for robotic manipulation via multi-modal diffusion. *arXiv preprint*, 2025.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. *arXiv preprint*, 2024.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint*, 2026.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. CoT-VLA: Visual chain-of-thought reasoning for vision-language-action models. In *CVPR*, 2025a.

Zhenyu Zhao, Hongyi Jing, Xiawei Liu, Jiageng Mao, Abha Jha, Hanwen Yang, Rong Xue, Sergey Zakharor, Vitor Guizilini, and Yue Wang. Humanoid everyday: A comprehensive robotic dataset for open-world humanoid manipulation. *arXiv preprint*, 2025b.

Jinliang Zheng, Jianxiong Li, Dongxiu Liu, Yinan Zheng, Zhihao Wang, Zhonghong Ou, Yu Liu, Jingjing Liu, Ya-Qin Zhang, and Xianyuan Zhan. Universal actions for enhanced embodied foundation models. In *CVPR*, 2025.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *ICLR*, 2026.

A Data and Training Details

This section summarizes the data sources used across the three training stages of *X-World* and the sampling strategy used for heterogeneous action pretraining.

A.1 Stage 1: Video pre-training

We use a mixture of robot videos and human videos to provide diverse visual interaction data for video pre-training. All sources are converted into a unified variable-view format, with up to three views retained for each sample. Table 6 summarizes the resulting data sources.

Table 6: Datasets for Stage 1 video adaptation. Robot and human videos supply future-frame supervision, while Views gives the number of available camera streams per training example.

Dataset	Views	Summary
RoboTwin 2.0 (Chen et al., 2025)	3	Domain-randomized dual-arm benchmark designed for robust bimanual manipulation and transfer.
CALVIN (Mees et al., 2022)	2	Long-horizon language benchmark built around static and gripper observations for sequential manipulation.
AgiBotWorld (Bu et al., 2025a)	3	Quality-controlled manipulation data collected in realistic settings with broad skill coverage.
Behavior-1K (Li et al., 2024)	3	Household simulation benchmark emphasizing realistic environments and long-horizon daily activities.
RoboCOIN (Wu et al., 2025)	3	Bimanual multi-embodiment data with hierarchical annotations for coordinated manipulation skills.
Galaxea (Jiang et al., 2025)	3	Open-world robot data from everyday scenes with fine-grained subtask language annotations.
Open-X-Embodiment (O'Neill et al., 2024)	1-3	Aggregated real-robot data spanning diverse embodiments, camera layouts, and manipulation behaviors.
RoboMIND (Wu et al., 2024)	1-3	Unified multi-embodiment teleoperation data covering diverse tasks, objects, and robot platforms.
Ego4D (Grauman et al., 2022)	1	Large-scale egocentric daily-life videos capturing diverse activities and environments from a first-person view.
Humanoid-Everyday (Zhao et al., 2025b)	1	Open-world humanoid manipulation data with multimodal sensing for whole-body interaction tasks.

A.2 Stage 2: Action pre-training

We use heterogeneous VLA trajectories collected from multiple benchmarks and robot platforms for action pre-training. These datasets differ in embodiment, observation setup, and heterogeneous state and action interfaces, providing broad supervision across diverse manipulation settings. Table 7 summarizes the main VLA data sources.

Table 7: Datasets for Stage 2 action pretraining. Robot trajectories provide action supervision, while Views gives the available camera count per example; dataset-specific projections accommodate native state and action formats.

Dataset	Views	Summary
RoboTwin 2.0 (Chen et al., 2025)	3	Domain-randomized dual-arm trajectories for robust bimanual manipulation and transfer.
CALVIN (Mees et al., 2022)	2	Long-horizon language-conditioned trajectories with static and gripper observations.
AgiBotWorld (Bu et al., 2025a)	3	Large-scale manipulation trajectories collected in realistic settings with diverse skills.
Behavior-1K (Li et al., 2024)	3	Household embodied benchmark providing long-horizon trajectories in realistic simulation.
RoboCOIN (Wu et al., 2025)	3	Bimanual multi-embodiment trajectories with hierarchical annotations for coordinated skills.
Galaxea (Jiang et al., 2025)	3	Open-world manipulation trajectories with fine-grained subtask language supervision.
Open-X-Embodiment (O'Neill et al., 2024)	1-3	Aggregated cross-embodiment trajectories covering diverse robots, scenes, and tasks.
RoboMIND (Wu et al., 2024)	1-3	Unified multi-embodiment teleoperation trajectories across diverse tasks and platforms.

A.3 TEMPERATURE-SCALED MULTI-DATASET SAMPLING

To avoid over-sampling the largest sources while still accounting for dataset scale, we use temperature-scaled sampling when drawing training examples from the Stage 2 mixture. The sampling weight of dataset d is defined as

$$w_d \propto \alpha_d \cdot N_d^{1/T}, \tag{7}$$

where N_d denotes the total number of dataset steps, i.e., the number of sampleable state and action transitions in dataset d, α_d is the base mixture weight, and T is the sampling temperature. We set $\alpha_d = 1.0$ and use $T = 2.0$ for all datasets, so the effective reweighting scales with the square root of the dataset step count before normalization.

A.4 STAGE 3: REAL-WORLD POST-TRAINING DATA

We use self-collected teleoperation data for post-training on the six real-world tasks described in Sec. 4.2. We collect 30 minutes of demonstrations per task with synchronized three-view observations. The platform, control interface, evaluation protocol, and task definitions are described in Sec. 4.2.

B IMPLEMENTATION DETAILS

This section provides additional implementation details for variable-view processing, the token sequence interface, heterogeneous state and action alignment, and asynchronous real-time chunking.

B.1 VARIABLE-VIEW VIDEO PROCESSING

We add a view embedding to encode view identity without assuming a fixed semantic ordering, and randomize view order during training to improve robustness to heterogeneous camera setups. Each training sample is represented with three view slots. When only one or two views are available, we replicate an existing view to fill the missing slots and apply independent diffusion noise to each replica, which avoids padding and prevents wasting computation on masked placeholders. Cross-view attention masks block interactions between replicated and original view slots, ensuring that replicated views do not affect the original views and vice versa.

B.2 FPS-AWARE TEMPORAL ROPE SCALING

Our video diffusion backbone builds on LTX-Video (HaCohen et al., 2024) and ingests videos collected at different frame rates. To align temporal positional encoding with the physical sampling rate of each sample, we pass the per-sample frame rate fps to the backbone and use it to construct the RoPE interpolation tuple (Su et al., 2024) in both training and inference:

$$\text{rope\_interpolation\_scale} = \left(\frac{r_t}{\mathrm{fps}},\ r_s,\ r_s\right), \tag{8}$$

where r_t and r_s denote the temporal and spatial compression ratios of the LTX-Video VAE. For multi-view inputs, the same sample-level fps is repeated across views. In the transformer, this tuple scales the temporal, height, and width coordinates before rotary embedding is applied, so clips with different frame rates share the same architecture while their temporal coordinates remain aligned with the actual sampling rate.

B.3 Q-FORMER AND TOKEN SEQUENCE INTERFACE

The video diffusion backbone produces spatiotemporal features throughout its denoising transformer blocks. We denote the exported hidden states by $\{h_\ell\}_{\ell=1}^{L}$ and distill them into a fixed-length token sequence using a layer-aligned Q-Former. At layer ℓ, the current query tokens cross-attend to the corresponding backbone hidden states h_ℓ before being passed to the next layer. The final token sequence forms a compact world prior whose size is independent of the number of input views and video tokens, and the action expert attends to this fixed interface when generating action chunks.

B.4 Action representation and dataset-specific modules

VLA datasets have heterogeneous state and action interfaces, including different action parameterizations, units, and coordinate frames. Rather than forcing a single shared action space, we use lightweight dataset-specific modules around a shared action-generation backbone, following the design in X-VLA (Zheng et al., 2026). Each dataset has a state projector, an action input projection for noised action chunks, and an output head that decodes shared representations into executable actions in its native parameterization. For the real-world platform, actions are represented as relative joint position targets, while absolute joint positions are provided as proprioceptive state.

B.5 Asynchronous real-time chunking

We maintain a FIFO action queue $\mathcal{Q}$ and execute and remove its front action at each control step. Let $R \triangleq |\mathcal{Q}|$ be the remaining queue length, P the prefix threshold, and E the number of newly predicted suffix actions appended after each refresh. When $R \leq P$, the policy uses the current observations and reuses the remaining R actions as a committed prefix to generate a refreshed action chunk. It discards the overlapping R predicted actions and appends the next E suffix actions to $\mathcal{Q}$. The robot continues executing committed actions while the policy updates future actions in the background, decoupling inference updates from action execution and avoiding waiting gaps.

In the reference deployment implementation, P corresponds to prefix_steps, E corresponds to execution_steps, and delay is computed as $\min(\text{remain\_steps}, \text{prefix\_steps})$. Thus, delay gives the number of remaining queued actions reused as the action prefix and dropped from the overlapping part of the returned prediction.

C Additional Video Predictions

The complete benchmark comparisons, real-world setup, task definitions, action-prefix and view-availability analyses, and runtime breakdown are included in Sec. 4. This appendix provides the additional qualitative video predictions.

Video adaptation. Figure 7 shows one-, two-, and three-view predictions after Stage 1 adaptation. The first column is the observed frame; subsequent columns show predicted future frames 8, 16, and 24 of the $K = 24$ clip. Rows within each example are synchronized camera views.

Real-world post-training. Figure 8 shows three-view predictions after real-world post-training for table clearing and block stacking. The temporal and camera conventions match Fig. 7. These examples illustrate video prediction appearance and do not establish its quantitative relationship to control success.

D Supplementary Videos

The supplementary package contains six real-world videos corresponding to the six tasks described in Sec. 4.2. The policy uses synchronized three-view observations; the videos display the third-person overview. Representative image sequences are included in Fig. 4.

- `01_stack_bowl_and_block.mp4`: Stack bowl and block.
- `02_put_bottle_into_trash_can.mp4`: Put bottle into trash can.
- `03_pour_water_from_teapot.mp4`: Pour water from teapot.
- `04_move_bowl_into_tray.mp4`: Move bowl into a tray.
- `05_stack_two_blocks.mp4`: Stack two blocks.
- `06_clear_the_table.mp4`: Clear the table.

The videos are qualitative examples and are not used to compute the reported success rates.

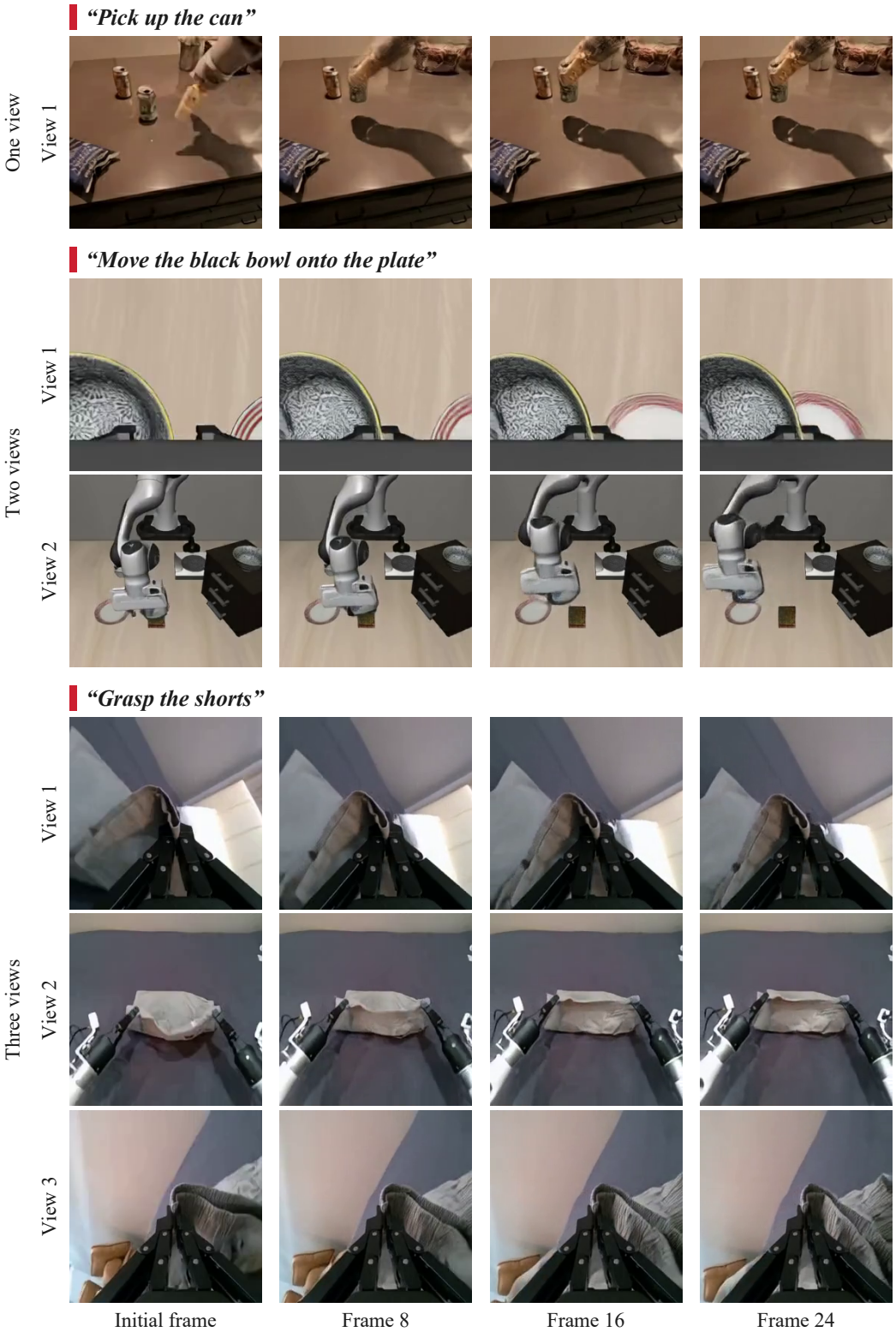


Figure 7: Future-video predictions after Stage 1 adaptation. The three examples use one, two, and three input camera views. Within each example, the first column contains observed frames and the remaining columns show predictions at frames 8, 16, and 24 of the 24-frame horizon; multiple rows correspond to synchronized views of the same sequence.

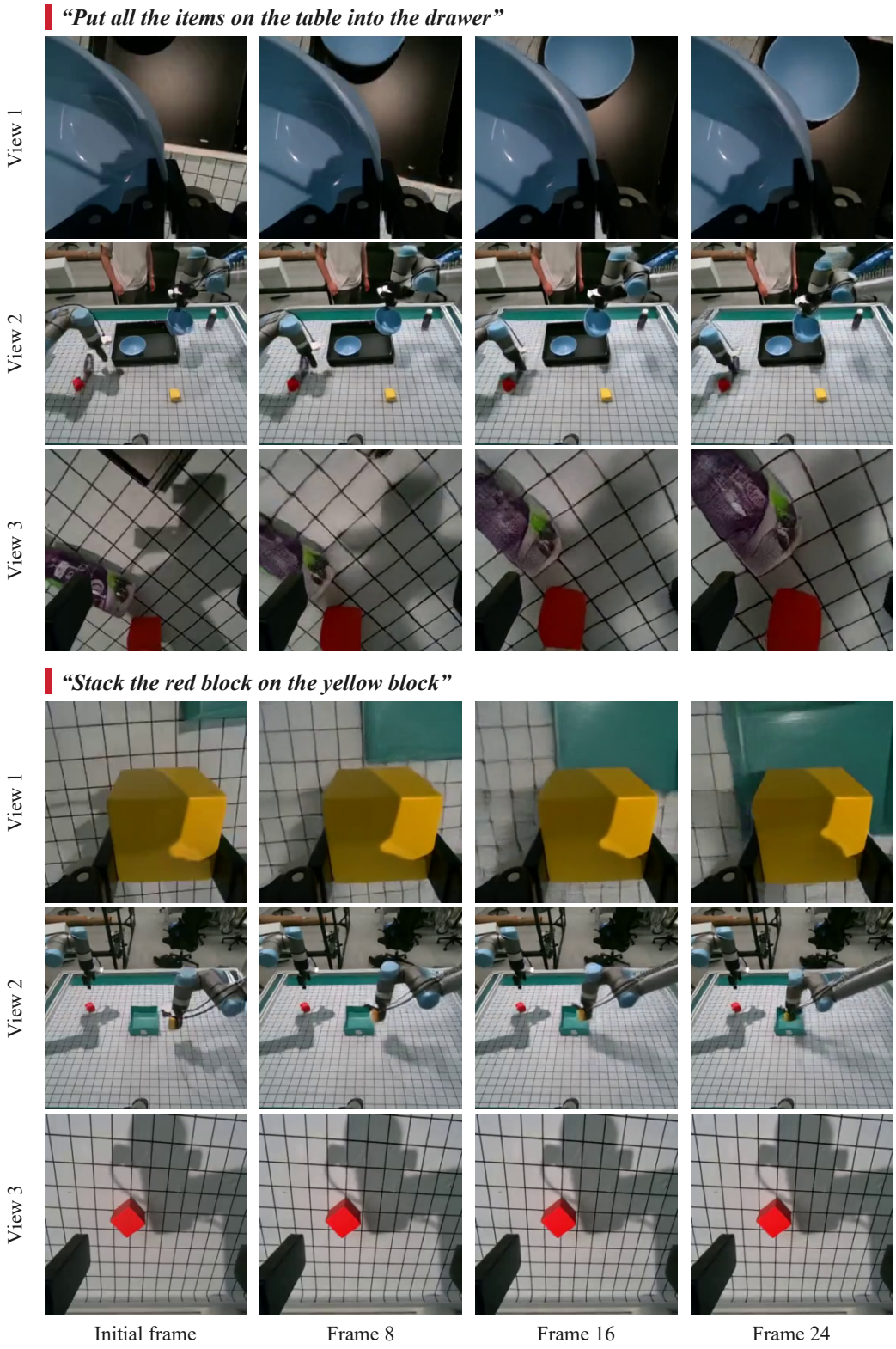


Figure 8: Future-video predictions after real-world post-training. Table clearing (top) and block stacking (bottom) each show three synchronized camera views. Observed frames occupy the first column, followed by predictions at frames 8, 16, and 24, so each row traces the predicted evolution of one camera view.