WORLDLINE: ACTION-DRIVEN VISUAL SIMULATION FOR ROBOTIC MANIPULATION

Anonymous authors
Paper under double-blind review

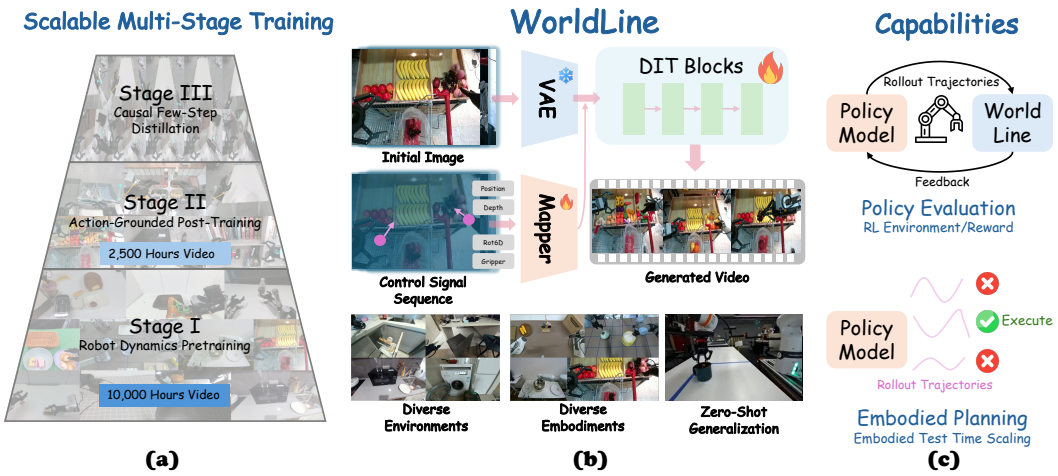

Figure 1: Overview of **WorldLine**. (a) A scalable three-stage training pipeline: **Stage I** learns robot dynamics from large-scale action-free videos; **Stage II** grounds heterogeneous robot controls through action-conditioned post-training; and **Stage III** distills the model for causal few-step rollout. (b) WorldLine maps an initial observation and an image-space control sequence to future videos across diverse environments and robot embodiments, including zero-shot settings. (c) WorldLine support policy evaluation and planning by predicting action outcomes before real-world execution.

ABSTRACT

Real-world robot learning is constrained by the cost of collecting experience and evaluating candidate behaviors. Video generation models offer a scalable foundation for visual simulators that predict action outcomes before physical execution. Yet they often favor visual plausibility over accurate action following and coherent robot–object dynamics, while action-conditioned simulators depend on scarce, embodiment-specific data that are difficult to share across incompatible control spaces. We introduce **WorldLine**, an action-driven visual simulator that decouples transferable dynamics learning from heterogeneous action grounding. WorldLine learns manipulation dynamics from more than 10,000 hours of action-free robot videos and grounds them using over 2,000 hours of action trajectories across more than ten embodiments. An image-space action representation provides a shared control interface across embodiments, while multi-view and failure-enriched training with relational regularization improves interaction-sensitive prediction. Robot-focused few-step distillation enables efficient causal rollout while preserving action-critical motion. Across held-out and out-of-domain settings, WorldLine maintains strong visual quality and robot-motion agreement; on failed trajectories, it improves robot-mask IoU by 0.1626 over the strongest baseline. It predicts trajectory success with 74% mean accuracy across RoboTwin and AgiBot, one percentage point above the strongest baseline. Without RoboTwin training or adaptation, its rollouts improve task success by up to 21.4 percentage points over direct policy execution. Together, these capabilities make WorldLine a scalable and efficient visual simulator for policy evaluation and embodied planning.

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

1 INTRODUCTION

Robot learning demands large-scale interaction data Dasari et al. (2020); Black et al. (2025b); Ye et al. (2026), yet real-world collection is costly. Conventional simulators Todorov et al. (2012); Makoviychuk et al. (2021) reduce this burden but require handcrafted assets and dynamics, limiting scalability across environments and embodiments. Video generation models Brooks et al. (2024); Wan et al. (2025) offer a data-driven visual prior, but their pretraining objectives primarily emphasize visual realism and temporal consistency without explicitly enforcing faithful responses to robot controls. Action-conditioned visual simulation therefore requires *action fidelity*: generated trajectories must accurately follow commanded direction, timing, and gripper state while preserving coherent robot–object dynamics over time Guo et al. (2026); Agarwal et al. (2026); Wang et al. (2026).

Learning a visual simulator that is both action-faithful and physically coherent at scale is difficult. Such a model requires large and diverse robot experience, yet action supervision is scarce and embodiment-specific. Pooling trajectories across robots is further complicated by native control signals that differ in dimensionality, kinematics, coordinate systems, and interfaces, preventing direct sharing Alzayer et al. (2026); Wu et al. (2026c); Zhen et al. (2026).

We introduce **WorldLine**, an action-driven visual simulator built on the insight that robot–object dynamics and precise action grounding can be learned separately, allowing dynamics learning to scale beyond scarce action-labeled trajectories. WorldLine first adapts a pretrained video model to more than 10,000 hours of action-free robot videos to learn a transferable prior over robot motion and object interaction. It then uses over 2,000 hours of action trajectories spanning more than ten robot embodiments to learn how heterogeneous controls drive future motion. Rather than forcing incompatible controls into a common vector space, WorldLine projects their spatial effects, including end-effector pose and gripper state, into each camera view, providing a shared image-space interface that preserves action geometry across embodiments. This action grounding enables predicted motion to faithfully follow commanded actions.

Accurate control following alone is not enough; predictions should also distinguish successful and unsuccessful interaction outcomes. WorldLine uses synchronized multi-view videos to provide complementary observations of robot and object motion, while failure trajectories expose the model to unsuccessful outcomes rather than biasing generation toward successful completion. Relational regularization further promotes temporal and cross-view consistency. Finally, robot-focused few-step distillation turns the model into an efficient causal simulator without sacrificing action-critical motion, enabling policy evaluation and embodied planning.

We evaluate WorldLine from three complementary perspectives: action-conditioned video generation, policy evaluation, and embodied planning. For action-conditioned generation, WorldLine maintains strong visual quality and robot-motion agreement across held-out and out-of-domain settings. On failed trajectories, it improves robot-mask IoU by 0.1626 over the strongest baseline. For policy evaluation, WorldLine rollouts predict trajectory success with 74% mean accuracy across RoboTwin Chen et al. (2026) and AgiBot Bu et al. (2025), one percentage point above the strongest baseline. For zero-shot RoboTwin planning, using these rollouts to select among policy-generated trajectories improves task success by 19.1 and 21.4 percentage points for $\pi_{0.5}$ Black et al. (2025a) and LingBot-VLA Wu et al. (2026a), respectively, over direct policy execution. Together, these results demonstrate WorldLine's scalability and practical value for robot evaluation and planning.

Our contributions are threefold:

- **Scalable cross-embodiment learning.** We introduce a decoupled framework that learns transferable robot–object dynamics from large-scale action-free videos and grounds heterogeneous robot controls through a geometry-preserving image-space interface.
- **Interaction-oriented modeling.** We combine synchronized multi-view observations, failure-enriched trajectories, and relational regularization to improve action-conditioned outcome modeling across viewpoints, environments, and embodiments. Few-step distillation further enables efficient causal rollout while preserving action-critical robot motion.
- **Broad empirical validation.** We evaluate WorldLine on action-conditioned video generation, policy evaluation, and best-of-N trajectory selection, demonstrating generalization to unseen environments and embodiments and consistent improvements over direct policy execution.

Table 1: Training-data comparison across action-conditioned robot video models in scale, diversity, and action supervision. Embod. denotes the number of robot embodiments, Failure denotes the duration of failed trajectories, and NR denotes unreported information.

	Scale		Diversity		Supervision	
Method	**Video-only**	**Action data**	**Embod.**	**Tasks**	**Action repr.**	**Failure**
CTRL-World Guo et al. (2026)	N/A	450 h	1	86	Native Action	60 h
GE-Sim-V2 Qiu et al. (2026)	N/A	2,500 h	1	NR	Image-space map	NR
Masked Visual Action Alzayer et al. (2026)	N/A	450 h	1	86	Masked Action	60 h
Worldline(ours)	**10,000 h**	**2,500 h**	**>10**	**>3,500**	Image-space map	**200 h**

2 Related Work

Video generation for robot learning. Video generation models have supported robot learning through visual prediction, synthetic data generation, and simulation Finn & Levine (2017); Dasari et al. (2020); Yang et al. (2024). Existing work uses generated videos to facilitate policy learning and augment training data Wang et al. (2026); Zhu et al. (2024), or to predict the outcomes of candidate actions for planning and policy evaluation. The latter requires generated trajectories to respond accurately to robot controls and preserve coherent robot–object interactions, rather than merely appear visually plausible Guo et al. (2026). WorldLine targets this predictive setting, with an emphasis on scalable dynamics learning and action-faithful simulation.

Action conditioning and scalable visual simulation. Existing visual simulators represent robot controls using native action vectors Agarwal et al. (2026), learned latent actions Ye et al. (2025); Bruce et al. (2024), or spatial conditions Wu et al. (2026c); Alzayer et al. (2026). Native actions provide direct control but are tied to embodiment-specific kinematics and coordinate systems. Latent actions can capture motion from unlabeled videos but require additional grounding before they can represent executable controls. Spatial conditions provide a geometrically aligned interface, but action representation alone does not resolve the data-scaling problem. Existing simulators still rely primarily on limited action-labeled trajectories whose control spaces and interaction distributions vary across embodiments O'Neill et al. (2024); Wu et al. (2026b). Most methods therefore learn visual dynamics and action semantics jointly from the same trajectories, coupling simulation coverage to available action supervision. WorldLine instead decouples these objectives by learning transferable robot–object dynamics from large-scale action-free videos and subsequently grounding heterogeneous controls through a shared image-space representation. This design allows action-free videos and embodiment-specific action trajectories to provide complementary supervision.

3 Data Pipeline

Data composition. WorldLine separates dynamics learning and action grounding across complementary data sources: action-free videos provide interaction coverage, while geometrically verified action trajectories provide precise control supervision. Figure 2 summarizes the data pipeline, while Table 1 compares its scale, diversity, and supervision with action-conditioned video models.

Action-free dynamics corpus. We curate more than 10,000 hours of robot videos from six collections, including AgiBotWorld Bu et al. (2025), RoboCOIN Wu et al. (2025), the RoboMIND series Wu et al. (2024); Hou et al. (2025), and Galaxea Jiang et al. (2025). The corpus spans more than ten robot embodiments and over 3,500 manipulation tasks. We retain one primary task-facing view per trajectory and filter out static, reset, corrupted, and otherwise uninformative clips.

Action-grounding corpus. For Stage II, we aggregate over 2,000 hours of action-labeled trajectories across more than ten robot embodiments. Each trajectory provides temporally aligned controls and calibrated camera geometry. We use robot kinematics to project end-effector motion into each camera view and discard samples with invalid calibration, temporal misalignment, or inconsistent projections. The verified trajectories are converted into camera-aligned action maps encoding projected position, depth, Rot6D orientation, and gripper state. We additionally construct a synchronized multi-view subset containing a head camera and up to two wrist cameras, and include approximately 200 hours of failure trajectories to broaden viewpoint and outcome coverage.

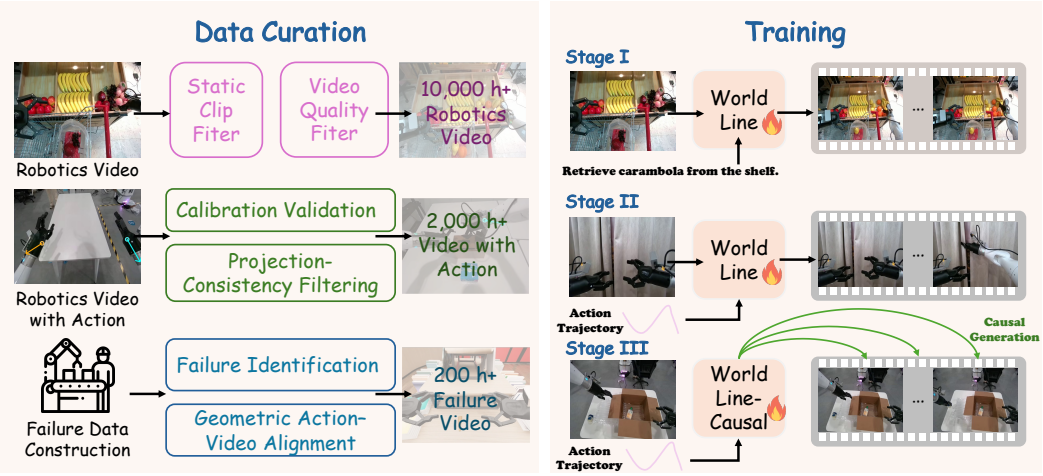

Figure 2: **Data construction and three-stage training of WorldLine.** Left: action-free videos are filtered for dynamics pretraining; action-labeled trajectories undergo calibration and projection checks; and failure trajectories are identified and paired with geometrically aligned controls. Right: Stage I learns manipulation dynamics, Stage II grounds heterogeneous controls with multi-view and failure-enriched training, and Stage III distills the model for efficient causal rollout.

Evaluation separation. We reserve a scene-disjoint subset of AgiBotWorld for in-domain evaluation and exclude it from model training. DROID is excluded from training and used only to evaluate out-of-distribution generalization across unseen environments, tasks, and robot embodiments.

4 Method

4.1 Overview

Given initial observations $\mathbf{x}_0^{1:V}$ from V views and a sequence of robot controls $\mathbf{q}_{1:T}$, WorldLine maps the controls to camera-aligned action maps $\mathbf{A}_{1:T}^{1:V} = \mathcal{G}(\mathbf{q}_{1:T})$ and predicts future observations:

$$\hat{\mathbf{x}}_{1:T}^{1:V} \sim p_\theta\big(\cdot \mid \mathbf{x}_0^{1:V}, \mathbf{A}_{1:T}^{1:V}\big)\,. \tag{1}$$

As illustrated in Figure 2, WorldLine is trained in three stages. Stage I learns transferable manipulation dynamics from large-scale action-free robot videos. Stage II grounds heterogeneous controls in image space while promoting interaction coherence through multi-view, failure-enriched, and relational training. Stage III distills the model for efficient causal rollout.

4.2 Stage I: Robot Dynamics Pretraining

We initialize WorldLine from a pretrained Cosmos3-Nano video model and adapt it to the action-free robot video corpus described in Section 3. Given future video latents $\mathbf{z} = \mathbf{z}_{1:T}$, an initial-frame latent $\mathbf{z}_0$, and task text y, we optimize the standard flow-matching objective

$$\mathcal{L}_{\text{pre}} = \mathbb{E}_{\mathbf{z},\boldsymbol{\epsilon},\sigma}\left[\left\|\mathbf{v}_\theta\left(\mathbf{z}_\sigma, \sigma \mid \mathbf{z}_0, \mathbf{e}(y)\right) - (\boldsymbol{\epsilon} - \mathbf{z})\right\|_2^2\right], \qquad \mathbf{z}_\sigma = (1-\sigma)\mathbf{z} + \sigma\boldsymbol{\epsilon}, \tag{2}$$

where $\boldsymbol{\epsilon}$ is Gaussian noise, σ the noise level, and $\mathbf{e}(y)$ the task-text embedding. Stage I learns a transferable prior over robot–object dynamics without action supervision to initialize Stage II.

4.3 Stage II: Action-Grounded Post-Training

Stage I learns broad manipulation dynamics but does not ground how robot controls determine future observations. Stage II converts this prior into an action-grounded simulator using geometrically verified action–video trajectories. We remove task-text conditioning, leaving generation conditioned only on the initial observations and control sequence.

4.3.1 CROSS-EMBODIMENT SPATIAL ACTION CONDITIONING

Native robot controls are not directly aligned across embodiments because they differ in dimensionality, coordinate conventions, and kinematics. For each control $\mathbf{q}_t$, we use embodiment-specific kinematics to obtain the commanded end-effector pose and transform it into each camera frame. Let $\mathbf{u}_t^v = \pi_v(\mathbf{p}_t)$ denote the projection of its 3D position $\mathbf{p}_t$ into view v. We construct a Gaussian heatmap with fixed bandwidth σ_g centered at $\mathbf{u}_t^v$:

$$\mathbf{h}_t^v(\mathbf{u}) = \exp\left(-\frac{\|\mathbf{u} - \mathbf{u}_t^v\|_2^2}{2\sigma_g^2}\right). \tag{3}$$

Using the same heatmap, we spatially encode the corresponding depth d_t^v, Rot6D orientation $\mathbf{r}_t^v \in \mathbb{R}^6$, and gripper state g_t. The resulting nine-channel action map at latent resolution $H_\ell \times W_\ell$ is

$$\mathbf{A}_t^v = \mathrm{Concat}\big(\mathbf{h}_t^v, d_t^v\mathbf{h}_t^v, \{r_{t,k}^v\mathbf{h}_t^v\}_{k=1}^6, g_t\mathbf{h}_t^v\big) \in \mathbb{R}^{9\times H_\ell \times W_\ell}, \qquad \mathbf{A}^v = \{\mathbf{A}_t^v\}_{t=1}^T. \tag{4}$$

The action maps depend only on the control sequence, robot kinematics, and camera calibration, without using future observations. We encode them with a lightweight residual branch and add the resulting tokens to the video-latent embeddings:

$$\widetilde{\mathbf{A}}^v = \mathbf{A}^v + \mathrm{Conv3D}(\mathbf{A}^v), \qquad \mathbf{E}^v = \mathbf{E}_{\mathrm{video}}^v + \mathbf{W}_{\mathrm{act}}\,\mathrm{Patchify}(\widetilde{\mathbf{A}}^v). \tag{5}$$

Because the action and video tokens share the same patch layout, their spatial and temporal coordinates remain aligned. Representing controls through their camera-space effects provides a shared conditioning interface while preserving action geometry across embodiments.

4.3.2 MULTI-VIEW AND FAILURE-ENRICHED TRAINING

Action grounding alone does not ensure coherent interactions: a single camera may miss gripper–object contact and object motion, while success-heavy data underrepresent failure outcomes. We jointly train on synchronized head and wrist views, whose tokens share temporal positions but retain view-specific spatial coordinates through RoPE. Each view receives camera-projected action maps, and the resulting multi-view flow-matching loss is denoted by $\mathcal{L}_{\mathrm{FM}}^{\mathrm{mv}}$. We apply the same objective to successful and failed trajectories without outcome labels, exposing the model to unsuccessful interaction outcomes. Further construction details are provided in Appendix A.

4.3.3 RELATIONAL DYNAMICS REGULARIZATION

Multi-view and failure-enriched data broaden interaction coverage, but the flow-matching objective does not explicitly preserve relational dynamics across time and viewpoints Zhang et al. (2026). We therefore regularize intermediate DiT features using a frozen V-JEPA2 Assran et al. (2025) applied to the target videos. Let $\mathbf{S}_t^v$ and $\mathbf{T}_t^v$ denote the student and teacher feature tokens at time t and view v. For row-normalized token features, their pairwise cosine-similarity matrix is $\mathcal{R}(\mathbf{F}, \mathbf{G}) = \overline{\mathbf{F}}\,\overline{\mathbf{G}}^\top$.

We match student and teacher relations between consecutive frames within each view and between synchronized frames across views:

$$\begin{aligned}
\mathcal{L}_{\mathrm{intra}} &= \mathbb{E}_{v,t}\left[\left\|\mathcal{R}(\mathbf{S}_t^v, \mathbf{S}_{t+1}^v) - \mathcal{R}(\mathbf{T}_t^v, \mathbf{T}_{t+1}^v)\right\|_1\right], \\
\mathcal{L}_{\mathrm{cross}} &= \mathbb{E}_{(v,v'),t}\left[\left\|\mathcal{R}(\mathbf{S}_t^v, \mathbf{S}_t^{v'}) - \mathcal{R}(\mathbf{T}_t^v, \mathbf{T}_t^{v'})\right\|_1\right].
\end{aligned} \tag{6}$$

The complete action-grounding objective is

$$\mathcal{L}_{\mathrm{post}} = \mathcal{L}_{\mathrm{FM}}^{\mathrm{mv}} + \lambda_{\mathrm{intra}}\mathcal{L}_{\mathrm{intra}} + \lambda_{\mathrm{cross}}\mathcal{L}_{\mathrm{cross}}. \tag{7}$$

These objectives transfer the teacher's temporal and cross-view relational structure to WorldLine, promoting coherent robot–object dynamics. Further details are provided in Appendix B.3.

4.4 STAGE III: ROBOT-FOCUSED CAUSAL DISTILLATION

Stage II generates full trajectories from a fixed action sequence through iterative ODE sampling, preventing online control updates and making repeated rollout costly. We distill it into a block-autoregressive student that accepts new controls at each block and generates the future in four denoising steps. Conditioned only on the causal history $\mathbf{h}_b$ and current action maps $\mathbf{A}_b$, the student is trained through causal adaptation, trajectory regression, and distribution matching Yin et al. (2024a).

Table 2: Action-conditioned video prediction on held-out AgiBotWorld success/failure trajectories and DROID, averaged over five runs. † denotes DROID-trained methods, making DROID in-domain. Arrows indicate metric direction; green and blue mark the best and second-best results.

Model	In-Domain								DROID Evaluation			
	AgiBotWorld (Successful)				AgiBotWorld (Failure)				DROID			
	PSNR↑	SSIM↑	LPIPS↓	Robot IoU↑	PSNR↑	SSIM↑	LPIPS↓	Robot IoU↑	PSNR↑	SSIM↑	LPIPS↓	Robot IoU↑
Video Generation Models												
Cosmos3-Nano Agarwal et al. (2026)	12.54	0.4752	0.5654	0.2804	10.35	0.4723	0.6000	0.1444	16.97	0.6516	0.3492	0.2049
Cosmos3-Super Agarwal et al. (2026)	12.57	0.4685	0.5674	0.2975	10.73	0.4786	0.6216	0.1755	17.07	0.6524	0.3457	0.1978
Action-Conditioned Models												
Ctrl-World† Guo et al. (2026)	15.68	0.6457	0.4690	0.2863	14.73	0.6881	0.4140	0.1917	**19.11**	**0.7502**	0.3010	0.1686
GE-Sim-V2 Qiu et al. (2026)	**18.90**	0.7334	0.2803	0.6300	15.67	0.7160	0.3285	0.3079	16.97	0.6771	0.3766	0.1959
OpenDW-0.5 Dexmal (2026)	16.67	0.6681	0.4639	0.4821	13.10	0.6364	0.5204	0.2187	15.53	0.6247	0.5012	0.0827
Masked Visual Actions† Alzayer et al. (2026)	14.53	0.6378	0.4208	0.3649	13.48	0.7143	0.3414	0.2083	17.05	0.7104	0.3635	0.1965
Ours												
WorldLine	18.67	**0.7931**	**0.1695**	**0.6539**	**19.19**	**0.8135**	**0.1913**	**0.4705**	19.04	0.7480	0.2968	**0.2739**
WorldLine (Causal 4-Step)	18.31	0.7206	0.2876	0.5923	18.32	0.7910	0.2241	0.4265	18.90	0.7232	**0.2705**	0.1986

Few-step distillation can overemphasize static backgrounds Zhu et al. (2026). Let $\overline{\mathbf{M}}$ denote the latent-resolution robot mask and $\widehat{\mathbf{z}}$ and $\mathbf{z}_T$ the student and teacher latents. We define

$$\mathcal{L}_{\mathrm{RF}} = \left\|\overline{\mathbf{M}} \odot (\widehat{\mathbf{z}} - \mathbf{z}_T)\right\|_2^2 + \lambda_\Delta \left\|\overline{\mathbf{M}} \odot (\Delta_t \widehat{\mathbf{z}} - \Delta_t \mathbf{z}_T)\right\|_1 . \tag{8}$$

This preserves robot appearance and motion alongside trajectory and distribution matching. The model enables low-latency causal rollout, online updates, and mask-free inference.

5 Experiments

We evaluate WorldLine's capabilities through three sets of experiments: action-conditioned prediction across in- and out-of-domain settings, policy evaluation, and embodied planning. We additionally assess causal rollout efficiency and ablate the key data, representation, and training components.

5.1 Action-Conditioned Video Generation

Evaluation protocol. Given an initial observation and recorded action sequence, each model predicts the future video. We evaluate on scene-disjoint AgiBotWorld Bu et al. (2025) trajectories with successful and failed executions, and on DROID Khazatsky et al. (2024), which is excluded from WorldLine training, adaptation, and checkpoint selection. The settings test generalization to unseen in-domain scenes and fully out-of-domain environments and robot embodiments, respectively.

Metrics. We report PSNR and SSIM for pixel-level fidelity and LPIPS for perceptual similarity. Because static backgrounds dominate full-frame metrics, we measure head-view robot-mask IoU for robot motion. We use SAM 3 Carion et al. (2026) to segment robots in generated and ground-truth frames and compute overlap. The projected end-effector location only initializes segmentation, while generated and ground-truth masks are inferred independently; thus, robot-mask IoU measures image-space robot-motion agreement. Metrics are reported on held-out AgiBotWorld and out-of-domain DROID; details appear in Appendix C.1.

Quantitative results. Table 2 compares WorldLine with video generation and action-conditioned simulation baselines. On successful AgiBotWorld trajectories, WorldLine ranks first in SSIM, LPIPS, and robot IoU and second in PSNR. On failures, it leads all four metrics and improves robot IoU by 0.1626 over the strongest baseline. Although DROID is out-of-domain for WorldLine and in-domain for Ctrl-World and Masked Visual Actions, WorldLine achieves the highest robot IoU (0.2739), exceeding the best prior result by 0.0690, with near-best visual quality. Its causal variant ranks second on failure metrics and achieves the best DROID LPIPS.

Qualitative results. Figure 3 shows how the quantitative gains translate into more accurate robot motion and resulting object configurations. Across AgiBotWorld and DROID, causal WorldLine aligns robot motion with the input controls while preserving the resulting gripper–object configuration. Competing methods instead exhibit robot drift, mismatched gripper–object configurations, or insufficient motion, particularly under DROID's unseen environments and embodiments.



Figure 3: **Qualitative action-conditioned video generation.** Given initial frame and actions, each model predicts a frame on held-out AgiBotWorld (top) and out-of-domain DROID (bottom). Causal WorldLine better matches ground-truth robot motion and gripper–object configuration.

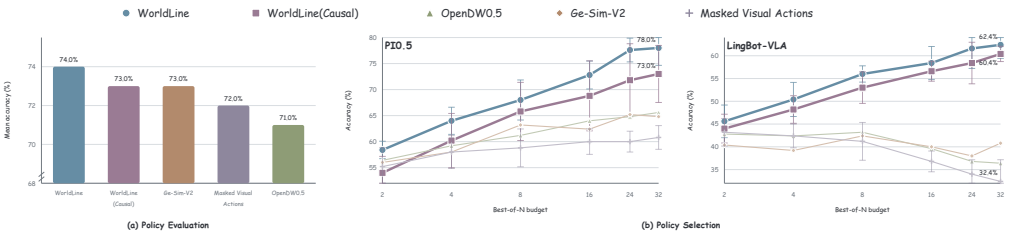

Figure 4: **WorldLine for policy evaluation and selection.** (a) Mean policy-evaluation accuracy over five generation seeds on RoboTwin and AgiBot. (b) RoboTwin task success rate under different rollout budgets for $\pi_{0.5}$ and LingBot-VLA.

5.2 Downstream Robotic Applications

Beyond video generation, a visual simulator must capture task-relevant action consequences to support decisions; we therefore evaluate WorldLine on policy evaluation and embodied planning.

5.2.1 Policy Evaluation

Evaluation protocol. We evaluate task-success prediction from rollouts on RoboTwin and AgiBot. From an initial observation and action sequence, each simulator generates a video that Qwen3VL-8B classifies as successful or unsuccessful. Accuracy is averaged over five generation seeds.

Results. Figure 4(a) shows that WorldLine achieves the highest mean accuracy of 74%, followed by causal WorldLine and GE-Sim-V2 at 73%, Masked Visual Actions at 72%, and OpenDW-0.5 at 71%. Causal WorldLine remains within one point of Stage II at lower latency.

5.2.2 Embodied Planning

Evaluation protocol. We evaluate out-of-domain rollout-based planning on RoboTwin Chen et al. (2026), which is excluded from WorldLine training, adaptation, and checkpoint selection. Across 10 tasks with five scenes each, $\pi_{0.5}$ Black et al. (2025a) and LingBot-VLA Wu et al. (2026a) each sample $N \in \{2, 4, 8, 16, 24, 32\}$ candidate action trajectories. Each simulator predicts their outcomes, Qwen3VL-8B scores task completion, and the highest-scoring trajectory is executed.

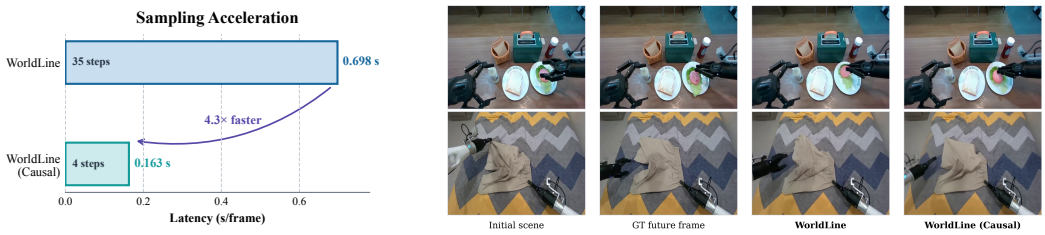

Figure 5: **Causal distillation efficiency and quality.** Left: sampling steps and per-frame latency under identical settings. Right: Stage-II and causal predictions on AgiBot. The causal model accelerates rollout while preserving appearance, robot geometry, and object evolution.

Baselines and metrics. We compare Stage-II and causal WorldLine with GE-Sim-V2, OpenDW-0.5, and Masked Visual Actions using shared candidates and Qwen3VL-8B Bai et al. (2025) selector. We report per-budget success and direct-execution gains over five independently sampled banks.

Results. Figure 4(b) shows WorldLine generally improves with larger budgets, whereas prior simulators scale weakly. Since Qwen3VL-8B is not an oracle, gains need not be monotonic. At $N = 32$, WorldLine raises success from 58.9% to 78.0% for $\pi_{0.5}$ and from 41.0% to 62.4% for LingBot-VLA, gains of 19.1% and 21.4%. Causal WorldLine reaches 73.0% and 60.4%, retaining most gains at lower cost. Without RoboTwin training, WorldLine provides reliable improvements across policies.

5.3 Causal Few-Step Acceleration

Evaluation protocol. We compare Stage-II and four-step causal WorldLine at batch size one under identical 129-frame settings to isolate speedup, reporting sampling steps and per-frame latency.

Results. Causal distillation reduces sampling from 35 to four steps, cutting 129-frame generation from 90 to 21 seconds and per-frame latency from 0.698 to 0.163 seconds, a $4.3\times$ speedup. Figure 5 shows that this acceleration preserves the main robot–object evolution in two AgiBot examples.

5.4 Ablation Studies

Evaluation protocol. We evaluate each variant using LPIPS and robot-mask IoU on held-out AgiBotWorld and DROID, plus RoboTwin planning success. For planning, Qwen3VL-8B selects from the same 32 trajectories sampled by $\pi_{0.5}$ and LingBot-VLA for every variant, matching the $N = 32$ setting in the main planning experiment. The variant without action-free data excludes the Stage-I corpus but uses the same total training-compute budget as the full model.

Data. Under matched total training compute, Table 3 shows that removing action-free data degrades visual quality and planning. This comparison supports the value of the additional interaction coverage provided by the action-free corpus, rather than an advantage caused merely by greater training compute. Head-view-only training also performs worse, showing that wrist views provide complementary close-range robot and object-motion cues. Failure trajectories further expose incomplete and unsuccessful outcomes, helping the model distinguish unsuccessful consequences rather than defaulting to task-completion patterns. Success-only data nearly preserves $\pi_{0.5}$ performance but reduces LingBot-VLA success by 10.8 points, highlighting failure coverage.

Action encoding and grounding. Replacing image-space action maps with FiLM-injected native action vectors reduces robot IoU on both datasets and planning success for both policies, showing the limitation of action conditioning without spatial grounding (Table 3). Among projection-based encodings, WorldLine outperforms GE-Sim-V2 and Masked Visual Actions under matched RoboTwin training and a 32-rollout budget as shown in Figure 6. Gains across both policies suggest spatial grounding improves action-outcome alignment beyond any single policy's prior.

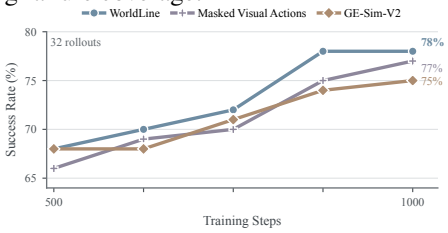

Figure 6: Matched action-encoding comparison on RoboTwin.

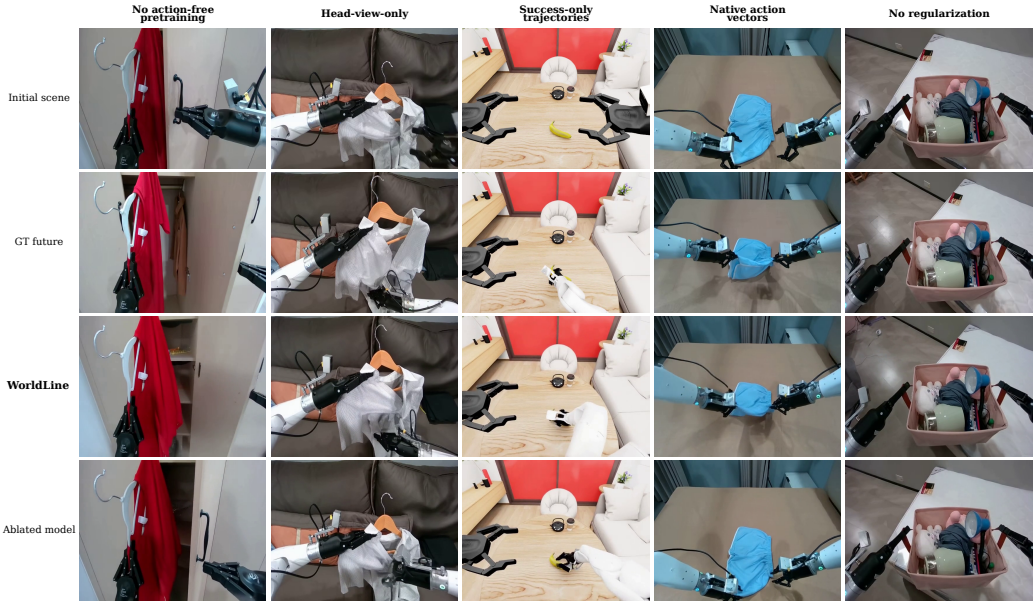

Figure 7: **Qualitative comparison of WorldLine ablations.** We compare the full model with variants that remove key data, representation, and training components, highlighting their effects on robot motion and robot–object interaction outcomes.

Table 3: **Ablation of WorldLine components.** LPIPS and robot-mask IoU are evaluated on held-out AgiBotWorld and out-of-domain DROID. RoboTwin planning uses 32 rollouts; gains are over direct execution. LPIPS is omitted for head-view-only. The no-action-free variant matches the full model's training-compute budget. Green and blue mark best and second-best results.

	AgiBotWorld		DROID		RoboTwin: $\pi_{0.5}$		RoboTwin: LingBot-VLA	
Variant	LPIPS↓	Robot IoU↑	LPIPS↓	Robot IoU↑	Success↑	Gain↑	Success↑	Gain↑
Full WorldLine	**0.1804**	**0.5622**	**0.2968**	**0.2739**	**78.0±1.4%**	**+19.1%**	**62.4±3.6%**	**+21.4%**
Data								
No action-free data (compute-matched)	0.2924	0.5277	0.3028	0.2614	73.2±3.3%	+14.3%	60.4±3.0%	+19.4%
Head-view-only data	–	0.4348	–	0.2665	72.8±1.10 %	+13.9%	62.0 ±3.2%	+21.0%
Success-only trajectories	0.2401	0.4721	0.3028	0.2566	77.2±2.3%	+18.3%	51.6±3.0%	+10.6%
Action Control								
Native action vectors	0.3118	0.3639	0.4537	0.1171	67.2±2.3%	+8.3%	53.2±3.3%	+12.2%
Training Strategy								
No regularization	0.2477	0.4593	0.3386	0.2303	77.2±3.0%	+18.3%	62.0±2.4%	+21.0%

Training strategy. Removing relational regularization substantially lowers robot IoU but barely affects planning success as shown in Table 3. By preserving feature relations, the regularizer limits local geometric drift and improves robot-motion consistency across viewpoints and time, which matters more for action-faithful prediction than coarse outcome ranking. Figure 7 qualitatively compares these effects on robot motion and interaction outcomes. The metric gap suggests relational regularization primarily stabilizes geometric consistency, while task-success ranking remains robust.

6 Conclusion

We presented WorldLine, an action-driven visual simulator that scales dynamics learning beyond scarce action-labeled data by separating action-free pretraining from image-space action grounding. This design learns transferable robot–object dynamics while grounding heterogeneous controls across embodiments and improving action-conditioned robot-motion and outcome prediction. Across held-out and fully out-of-domain settings, WorldLine follows commanded actions, predicts trajectory success, and improves embodied planning through rollout selection. Four-step causal distillation retains these capabilities at substantially lower latency. Thus, action-driven visual simulation can scale across embodiments while remaining useful for downstream robotic decisions.

AI USE STATEMENT

AI-assisted tools were used solely for language editing and manuscript-level quality control. This included improving clarity and readability, correcting grammatical and typographical errors, and identifying potential factual inaccuracies, ambiguous statements, and inconsistencies across sections. All suggestions were reviewed and verified by the authors before incorporation. AI tools were not used to generate experimental results. The authors take full responsibility for this paper.

REFERENCES

Niket Agarwal, Arslan Ali, Jon Allen, Martin Antolini, Adeline Aubame, Alisson Azzolini, Junjie Bai, Maciej Bala, Yogesh Balaji, Josh Bapst, et al. Cosmos 3: Omnimodal world models for physical ai. *arXiv preprint arXiv:2606.02800*, 2026.

Hadi Alzayer, Wenlong Huang, Haonan Chen, Christopher Luey, Lvmin Zhang, Maneesh Agrawala, Gordon Wetzstein, Li Fei-Fei, Yilun Du, Jiajun Wu, et al. Masked visual actions for unified world modeling. *arXiv preprint arXiv:2607.19343*, 2026.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, brian ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 27–30 Sep 2025a. URL https://proceedings.mlr.press/v305/black25a.html.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. In *RSS 2025*, 2025b.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators, 2024. URL https://openai.com/research/video-generation-models-as-world-simulators.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Maria Elisabeth Bechtle, Feryal Behbahani, Stephanie C.Y. Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=bJbSbJskOS.

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

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. In *International conference on learning representations*, volume 2026, pp. 138846–138923, 2026.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=itonej9GIV.

Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. Robonet: Large-scale multi-robot learning. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura (eds.), *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 885–897. PMLR, 30 Oct–01 Nov 2020. URL https://proceedings.mlr.press/v100/dasari20a.html.

Dexmal. OpenDW: An open-source world model for action-conditioned embodied intelligence. GitHub repository, 2026. URL https://github.com/dexmal/opendw. Accessed September 26, 2026.

Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 2786–2793. IEEE, 2017.

Yanjiang Guo, Lucy Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 6121–6138, 2026.

Chengkai Hou, Kun Wu, Jiaming Liu, Zhengping Che, Di Wu, Fei Liao, Guangrun Li, Jingyang He, Qiuxuan Feng, Zhao Jin, et al. Robomind 2.0: A multimodal, bimanual mobile manipulation dataset for generalizable embodied intelligence. *arXiv preprint arXiv:2512.24653*, 2025.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *Advances in Neural Information Processing Systems*, 38:167283–167308, 2026.

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 arXiv:2509.00576*, 2025.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance GPU based physics simulation for robot learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=fgFBtYgJQX_.

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: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024.

Boxiang Qiu, Liliang Chen, Yue Liao, Nan Wang, Lintao Wang, Jiayi Luo, Wenzhi Zhao, Shengcong Chen, Di Chen, Ye Li, et al. Ge-sim 2.0: A roadmap towards comprehensive closed-loop video world simulators for robotic manipulation. *arXiv preprint arXiv:2605.27491*, 2026.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Yixuan Wang, Rhythm Syed, Fangyu Wu, Mengchao Zhang, Aykut Onol, Jose Barreiros, Hooshang Nayyeri, Tony Dear, Huan Zhang, and Yunzhu Li. Interactive world simulator for robot policy training and evaluation. In *Robot World Models*, 2026. URL https://openreview.net/forum?id=x4qnERKsNI.

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 arXiv:2412.13877*, 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 arXiv:2511.17441*, 2025.

Wei Wu, Fan Lu, Yunnan Wang, Shuai Yang, Shi Liu, Fangjing Wang, Qian Zhu, He Sun, Yong Wang, Shuailei Ma, et al. A pragmatic vla foundation model. *arXiv preprint arXiv:2601.18692*, 2026a.

Xionghao Wu, Yijun Yang, Shiyang Zhou, Haoze Sun, Jianhui Liu, Songsong Yu, Jiyao Zhang, Wenbo Li, Bo Wang, Guoqing Ma, et al. Zimablue: Evolving generalizable world action models through scalable video pre-training. *arXiv preprint arXiv:2609.00188*, 2026b.

Zhenyu Wu, Xiuwei Xu, Yukun Zhou, Yifan Li, Qiuping Deng, Xiaofeng Wang, Zheng Zhu, Bingyao Yu, Ziwei Wang, Jiwen Lu, et al. imac: Translating actions into motion and contact images for embodied world models. *arXiv preprint arXiv:2606.09813*, 2026c.

Sherry Yang, Yilun Du, Seyed Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 45210–45234, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/c4d66eae503694424123b93ac0fbaf17-Paper-Conference.pdf.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, Lars Liden, Kimin Lee, Jianfeng Gao, Luke Zettlemoyer, Dieter Fox, and Minjoon Seo. Latent action pretraining from videos. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 28213–28239, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/45d74e190008c7bff2845ffc8e3facd3-Paper-Conference.pdf.

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 arXiv:2602.15922*, 2026.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6613–6623. IEEE, 2024b.

Peiwen Zhang, Yufan Deng, Shangkun Sun, Juncheng Ma, Duomin Wang, Jonas Du, Zilin Pan, Ye Huang, Hao Liang, Songyan Huang, et al. Physisforcing: Physics reinforced world simulator for robotic manipulation. *arXiv preprint arXiv:2606.28128*, 2026.

Haoyu Zhen, Zixian Gao, Qiao Sun, Yilin Zhao, Yuncong Yang, Yilun Du, Pengsheng Guo, Tsun-Hsuan Wang, Yi-Ling Qiao, and Chuang Gan. Action images: End-to-end policy learning via multiview video generation. *arXiv preprint arXiv:2604.06168*, 2026.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. Irasim: Learning interactive real-robot action simulators. *arXiv preprint arXiv:2406.14540*, 2(5):6, 2024.

Hongzhou Zhu, Min Zhao, Guande He, Hang Su, Chongxuan Li, and Jun Zhu. Causal forcing: Autoregressive diffusion distillation done right for high-quality real-time interactive video generation. *arXiv preprint arXiv:2602.02214*, 2026.

A Data Construction Details

A.1 Stage-Wise Data Composition

WorldLine uses complementary data in its first two training stages. Stage I learns general robot motion and object-interaction dynamics from a diverse collection of robot videos. For each trajectory, we select a primary task-facing view, typically a head, top, or front camera that clearly observes the robot workspace. Although some source datasets contain robot states or actions, Stage I does not provide these signals to the model and treats all samples as action-free videos.

Stage II uses robot trajectories with sufficiently reliable state or action sequences and camera geometry. Unlike Stage I, it focuses on grounding robot motion through spatial action conditions and jointly processes synchronized head, left-wrist, and right-wrist observations when all required views are available.

A.2 Video Sampling and Motion Filtering

Training clips are sampled dynamically from valid trajectory segments rather than being pre-segmented into fixed, non-overlapping clips. The sampled clips contain between 33 and 129 frames, allowing the model to observe interactions at different temporal scales. Stage I uses a larger temporal stride to cover longer motion, whereas Stage II samples more densely to preserve fine-grained action effects.

Following Figure 2, we apply source-specific motion and video-quality criteria to remove static, corrupted, and chassis-motion-dominated clips. Videos are resized to source-specific training resolutions, with camera intrinsics adjusted accordingly.

A.3 Action Geometry and Multi-View Construction

For Stage II, we retain trajectories that pass calibration validation and projection-consistency filtering, as summarized in Figure 2. Visual observations and controls are aligned by frame index, and forward kinematics transforms each end-effector pose into the corresponding camera frame. The projected position defines a Gaussian spatial condition that also encodes depth, Rot6D orientation, and gripper state, producing the nine-channel action representation described in the main paper. Invalid or out-of-view projections are masked, and wrist-camera poses are updated from robot kinematics when dynamic extrinsics are available.

Stage II uses three semantic camera slots corresponding to the head or top camera, left wrist, and right wrist. All views within a trajectory share the same sampled frame indices and clip length. Trajectories missing a required wrist view are excluded from multi-view training rather than automatically converted into single-view samples.

A.4 Failure Data and Evaluation Separation

Failure trajectory construction. In addition to successful demonstrations, we construct a dedicated failure dataset by collecting controlled robot executions that do not successfully complete the intended manipulation task. These trajectories capture diverse unsuccessful interactions, such as missed grasps, object slippage, unintended collisions, and incomplete manipulation. We retain the corresponding multi-view observations, robot states or actions, and camera geometry, and process them using the same temporal sampling and action-grounding pipeline as the remaining Stage-II data. During training, failure trajectories are treated as ordinary action–video pairs without providing explicit outcome labels. The simulator must therefore learn the visual consequences of unsuccessful actions directly from the observed trajectories rather than from a predefined failure category.

Evaluation separation. For evaluation, we construct a scene-disjoint AgiBotWorld split such that the evaluation scenes and their associated trajectories are excluded from training. DROID is not used in any WorldLine training stage, model adaptation, or checkpoint selection. We directly evaluate the standard WorldLine checkpoint on DROID. Compared with the predominantly bimanual training data, DROID contains substantially different camera viewpoints and single-arm manipulation set-

tings, providing an out-of-distribution evaluation of viewpoint and embodiment-configuration generalization.

B Implementation Details

B.1 Model Initialization and Stage-I Training

WorldLine is built upon the pretrained Cosmos3-Nano video generation model. All existing model components are initialized from pretrained weights. Stage I performs full-parameter training on the action-free robot video corpus. The model is conditioned on the initial observation and task description and is optimized using the flow-matching objective described in the main paper. This stage adapts the pretrained video prior to robot motion and manipulation dynamics without consuming robot actions.

B.2 Stage-II Action-Grounded Post-Training

Stage II is initialized from the Stage-I model and removes task-text conditioning. All existing modules inherit their weights from the preceding checkpoint, while the newly introduced action-conditioning branch is zero-initialized. The zero initialization preserves the behavior of the action-free model at the beginning of post-training and allows action control to be introduced progressively.

For each view, the nine-channel action maps are rendered directly at the video latent resolution. For a 480×640 input, the corresponding action tensor has shape $9 \times T_\ell \times 30 \times 40$. We use a fixed Gaussian bandwidth of $\sigma_g = 10$ pixels on the 384×512 reference canvas. The action encoder applies a single $\mathrm{Conv3D}(9, 9)$ layer with a $1 \times 3 \times 3$ kernel and adds its output to the input action maps. The resulting representation is spatially patchified using 2×2 patches and projected from 36 to 4096 dimensions. The action tokens are then added to the spatially and temporally aligned video tokens before the first DiT block. No numerical-action FiLM modulation is used.

The three camera views are ordered as head, left wrist, and right wrist. They share the same temporal coordinates, while their RoPE height coordinates begin at fixed semantic slots of 0, 480, and 992, respectively. During Stage II, we optimize the generation-specific MoE modules, time embeddings, video-latent input and output projections, and the action-conditioning branch, while keeping the remaining pretrained components fixed. We use AdamW with betas $(0.9, 0.95)$, weight decay 10^{-6}, and a base learning rate of 1.5×10^{-5}. The newly initialized action branch uses a 66-times learning-rate multiplier. Training uses a per-device batch size of one, gradient accumulation of two, and gradient clipping at 0.2.

B.3 Relational Regularization

We use a frozen V-JEPA2 ViT-g/384 encoder as the relational teacher. We choose V-JEPA2 because its self-supervised representations capture semantic motion and interaction structure beyond low-level pixel similarity, providing stable targets for modeling robot–object dynamics without requiring additional annotations.

Teacher features are obtained by concatenating the normalized outputs of its last four Transformer blocks. Student features are extracted from the output of the 18th DiT block. Because the teacher and student have different feature dimensions, we do not directly regress their feature values. Instead, both representations are temporally aggregated into eight bins and spatially pooled to an 8×8 grid. Relational supervision is then applied to their cosine-similarity matrices.

The intra-view objective aligns token relations between adjacent temporal bins within the same camera view. At each optimization step, one of the head, left-wrist, and right-wrist views is selected in a round-robin manner. The cross-view objective aligns the head view with one wrist view and alternates between the left and right wrists.

Rather than using fixed loss coefficients, we dynamically scale the relational objectives according to exponential moving averages of their ratios to the flow-matching loss. The target ratios are 0.03 for intra-view regularization and 0.005 for cross-view regularization, with maximum scales of 0.20 and

0.10, respectively. The scales are gradually introduced during the first 250 optimization steps. The V-JEPA2 teacher remains frozen throughout training.

B.4 Causal Distillation and Rollout

We convert the non-causal Stage-II model into an efficient block-autoregressive simulator through three successive stages: causal teacher-forcing adaptation, four-step trajectory regression, and score-based distribution matching. Each stage is initialized from the checkpoint produced by the preceding stage.

B.4.1 Causal Teacher-Forcing Adaptation

We first introduce causal attention and block-wise generation while retaining the Stage-II flow-matching objective. The model receives one initial latent frame as the observation condition and generates four latent frames per block. Given the temporal compression ratio of the video tokenizer, each generated block corresponds to 16 RGB frames. During training, each block is conditioned on ground-truth preceding blocks and the action maps covering the same temporal interval. This stage teaches the model to generate future blocks under a causal attention pattern while preserving the action-grounded dynamics learned by the Stage-II model.

B.4.2 Four-Step Trajectory Regression

We next use the frozen causal teacher to construct offline four-step denoising trajectories. Each block is generated using four explicit Euler updates with the noise schedule

$$1.0 \rightarrow \frac{15}{16} \rightarrow \frac{5}{6} \rightarrow \frac{5}{8} \rightarrow 0. \tag{9}$$

At each noisy node, the student regresses the next node on the teacher trajectory. An additional terminal objective preserves the final clean prediction:

$$\mathcal{L}_{\text{traj}} = \lambda_{\text{next}} \sum_{k=0}^{K-1} \left\| \widehat{\mathbf{z}}^{(k+1)} - \mathbf{z}_T^{(k+1)} \right\|_2^2 + \lambda_{\text{terminal}} \left\| \widehat{\mathbf{z}}^{\text{clean}} - \mathbf{z}_T^{\text{clean}} \right\|_2^2 + \mathcal{L}_{\text{aux}}, \tag{10}$$

where $\mathbf{z}_T$ denotes the trajectory generated by the frozen teacher, $\lambda_{\text{next}} = 1.0$, and $\lambda_{\text{terminal}} = 0.5$. This stage compresses the teacher's iterative denoising trajectory into four student updates.

B.4.3 Score-Based Distribution Matching

Trajectory regression encourages the student to follow individual teacher trajectories but does not directly align the overall generated distribution Huang et al. (2026); Zhu et al. (2026). We therefore further optimize the student using score-based distribution matching Yin et al. (2024b;a). Given a student-generated clean latent $\mathbf{G}_\theta$, we add noise to obtain $\mathbf{x}_t$. A frozen teacher estimates the flow field of the real data distribution, while a trainable fake-score model estimates the flow field of the generated distribution:

$$\mathbf{g}_{\text{DMD}}(\mathbf{x}_t, t) = \mathbf{v}_{\text{real}}(\mathbf{x}_t, t) - \mathbf{v}_{\text{fake}}(\mathbf{x}_t, t). \tag{11}$$

The student is updated using the detached pseudo-target

$$\mathcal{L}_{\text{DMD}} = \left\| \mathbf{G}_\theta - \text{sg}(\mathbf{G}_\theta - \mathbf{g}_{\text{DMD}}) \right\|_2^2. \tag{12}$$

The fake-score model is separately trained with a flow-matching objective on student-generated samples. We perform one student update and five fake-score updates every six optimization iterations.

The distribution-matching stage additionally uses latent-anchor, boundary-consistency, perceptual, and robot-focused auxiliary objectives:

$$\begin{aligned} \mathcal{L}_{\text{SF-DMD}} = \mathcal{L}_{\text{DMD}} &+ \lambda_{\text{anchor}} \mathcal{L}_{\text{anchor}} + \lambda_{\text{boundary}} \mathcal{L}_{\text{boundary}} \\ &+ \lambda_{\text{LPIPS}} \mathcal{L}_{\text{LPIPS}} + \lambda_{\text{robot}} \mathcal{L}_{\text{robot}} + \lambda_{\Delta\text{robot}} \mathcal{L}_{\Delta\text{robot}}, \end{aligned} \tag{13}$$

where the corresponding weights are 0.005, 0.05, 0.005, 0.005, and 0.1.

The robot-focused objectives use binary masks constructed by projecting robot links, gripper branches, and end-effectors into each camera view using forward kinematics and camera calibration. The masks are temporally expanded over neighboring frames and spatially dilated at the latent resolution. Robot-region reconstruction and temporal-motion losses are normalized by the valid mask area. The left- and right-wrist losses are assigned twice the weight of the head-view loss.

B.4.4 Block-Autoregressive Rollout

After each block is generated, its clean key-value features are inserted into the causal cache. The next block is initialized from independent Gaussian noise, while temporal continuity is maintained through the cached history rather than by repeating the final frame of the previous block. For long-horizon rollout, the cache retains one persistent sink block and the three most recent generated blocks. This fixed four-block history bounds memory and computation as the generated sequence becomes longer. The three camera views are generated jointly within each block.

B.5 Inference Configuration

Stage-II WorldLine uses 35 denoising steps, whereas Causal WorldLine uses four steps for each generated block. Both models use an explicit Euler solver for the rectified-flow probability-flow ODE and do not use classifier-free guidance. Unless otherwise specified, inference uses a random seed of 42, a per-view resolution of 480×640, and a batch size of one trajectory. The head, left-wrist, and right-wrist views are generated jointly rather than being concatenated into a single image input.

A 129-frame rollout consists of one observed initial frame followed by eight generated blocks. Latency is measured end-to-end on a single NVIDIA H200 GPU using FP16 precision and batch size one. The measurement includes VAE encoding, video generation, and VAE decoding, while excluding MP4 encoding and disk writing. Stage-II WorldLine requires 90 seconds for a 129-frame rollout, corresponding to 0.698 seconds per frame. Causal WorldLine requires 21 seconds, corresponding to 0.163 seconds per frame and a $4.3\times$ speedup.

C Experimental Details

C.1 Action-Conditioned Video Generation

Evaluation sets. We evaluate action-conditioned video generation on the scene-disjoint AgiBotWorld test set and the fully out-of-domain DROID test set. The AgiBotWorld test scenes and trajectories are excluded from training. DROID is not used for WorldLine training, adaptation, or checkpoint selection and differs from the predominantly bimanual training data in both camera viewpoints and robot embodiment. Each method receives the same initial observation and action sequence and predicts the corresponding future observations. CTRL-World and Masked Visual Actions are evaluated using models trained on DROID.

Full-frame metrics. PSNR, SSIM, and LPIPS are computed frame by frame using their standard implementations, including the conditioning frame at $t = 0$. Each metric is averaged temporally within each trajectory and view, then equally across trajectory–view pairs from the head, left-wrist, and right-wrist cameras.

Robot-mask IoU. Robot-mask IoU is evaluated only on the head-camera view. Unlike wrist cameras, which move together with the end effectors and often contain partial or close-range views of the robot, the head camera provides a stable global view of the workspace and more consistently captures the spatial relationship between the robot and manipulated objects. It therefore enables a more comparable measurement of action-conditioned robot motion across trajectories.

We use SAM 3 to segment the robot in each generated and ground-truth video, using the end-effector projections only to initialize segmentation and then propagating the masks over time. The two videos are segmented independently; ground-truth masks are never used to prompt predicted videos. Given

the ground-truth and predicted robot masks M_t and $\widehat{M}_t$, frame-level IoU is

$$\mathrm{IoU}_t = \frac{|M_t \cap \widehat{M}_t|}{|M_t \cup \widehat{M}_t|}. \tag{14}$$

Frames in which both masks are empty are excluded rather than assigned an IoU of one. If only one mask is empty, the frame receives an IoU of zero. We first average valid frame-level IoUs within each video and then average over the evaluation videos. Videos whose masks are empty in every frame are excluded from this average.

C.2 Embodied Planning

Policies and tasks. We evaluate embodied planning in RoboTwin using one fixed checkpoint of $\pi_{0.5}$ and one fixed checkpoint of LingBot-VLA, each trained on RoboTwin. In contrast, WorldLine is directly evaluated in RoboTwin without using any RoboTwin data for training, adaptation, or checkpoint selection. The policy checkpoints are held fixed across all simulators and rollout budgets. They are not required to represent fully converged policy performance because the purpose of this experiment is to measure the relative improvement obtained from simulator-based trajectory selection over direct execution of the same policy. We evaluate 10 manipulation tasks with five scenes per task.

Candidate generation. For each task and scene, a policy generates a bank of up to 32 candidate action trajectories. Every simulator receives the same initial observation and candidate actions. WorldLine renders each candidate as a predicted video. We sample 16 frames from each video: ten frames are distributed over the complete trajectory, while six additional frames are distributed over its final 40%. This sampling preserves the overall action progression while providing denser evidence for placement, release, object dropping, and terminal-state evaluation.

Hierarchical candidate selection. We use Qwen3VL-8B to select one trajectory from each candidate bank. Candidate identities are hidden from the model and replaced with temporary letter labels. The candidate ordering is deterministically shuffled for each scene. For the 32-candidate setting, candidates are processed using a hierarchical bracket: Qwen3VL-8B selects two candidates from each group of four, progressively reducing the bank from 32 to 16, from 16 to 8, and from 8 to 4. The four finalists are then scored independently from 0 to 100 according to their predicted likelihood of completing the task. The highest-scoring candidate is selected. Tied finalists are compared again using anonymous labels.

For each of five runs, each policy independently samples a new bank of 32 candidate trajectories using a distinct generation seed. All simulators evaluate the same bank within each run. Rollout budgets of 2, 4, 8, 16, 24, and 32 use prefixes of the run-specific candidate ordering and the corresponding selection procedure. Reported error bars denote standard deviation across the five independently sampled banks. Ground-truth success labels are accessed only after selection and are never provided to Qwen3VL-8B.

Qwen3VL-8B configuration. Qwen3VL-8B is evaluated in BF16 using deterministic decoding with `do_sample=False` and at most 256 generated tokens. The multimodal conversation contains a single user message and no separate system prompt. The following prompt is used for each four-way comparison. The value of `select_count` is two during the bracket rounds and one during a tie-breaking comparison.

Prompt P1: Candidate Comparison

```
Task instruction: {instruction}
Candidate A: <video A>
Candidate B: <video B>
Candidate C: <video C>
Candidate D: <video D>

Compare these candidates and select the {select_count} most likely to actually complete
the task. Prioritize correct final object state, then valid grasp/contact and placement.
Reject partial progress, drops, and physical inconsistencies. Ignore visual quality unless
it prevents judging. You must return exactly {select_count} candidate label(s), even if
every candidate appears to fail; in that case choose the least unsuccessful. Reply only
as JSON using the displayed letter labels:
{"selected_labels": ["A"], "reason": "<brief comparative reason>"}
```

The final four candidates are scored independently using the following prompt. The integer score is used as the primary ranking criterion, while `task_completed` and the textual fields are retained for analysis.

Prompt P2: Independent Candidate Scoring

```
Task instruction: {instruction}
Candidate video:

Judge this candidate independently. Determine whether the predicted trajectory ends
with the task actually completed. Check the initial and final object state, valid
grasp/contact, correct placement, drops, and physical inconsistencies. Mere contact or
partial progress is not completion. Ignore visual quality unless it prevents judging.
Give a 0-100 task-success score. Reply only as JSON:
{"task_completed": <true or false>, "score": <integer>,
 "final_state": "<brief description>", "failure_reason": "<brief reason or empty>"}
```

Selector validation on ground-truth rollouts. To isolate ranking from video-generation errors, we apply Qwen3VL-8B directly to ground-truth RoboTwin rollouts from $\pi_{0.5}$ using the same selection protocol. Selected-trajectory success reaches 69.0%, 79.3%, 82.8%, 82.8%, 85.9%, and 89.3% for $N \in \{2, 4, 8, 16, 24, 32\}$, respectively. This increase shows that the selector can identify successful trajectories when given accurate visual outcomes, supporting its use in rollout-based planning.

C.3 Policy Evaluation

Evaluation set. Policy evaluation uses balanced sets of successful and failed robot trajectories. For RoboTwin, we sample 50 base trajectories, comprising 25 successful and 25 failed executions drawn from $\pi_{0.5}$ and LingBot-VLA. For AgiBot, we likewise sample 50 trajectories with an equal number of successful and failed executions. All visual simulators are evaluated on the same matched base trajectories. RoboTwin data are not used for WorldLine training, adaptation, or checkpoint selection.

For each base trajectory, every simulator generates five outcome videos using distinct generation seeds. Qwen3VL-8B decoding and the subsequent decision rules remain deterministic, so variation across runs arises from video generation.

WorldLine receives the initial observations and recorded action sequence and generates the corresponding future video. WorldLine itself does not output a success label. Instead, Qwen3VL-8B extracts structured visual evidence from each generated trajectory, after which deterministic rules map the extracted evidence to a success or failure prediction.

Observation sampling. For each trajectory, we sample 12 temporal observations at normalized positions

$$p_i = \sqrt{\frac{i}{N-1}}, \qquad i = 0, \ldots, N-1. \tag{15}$$

This schedule allocates more observations to the latter part of the trajectory, where task completion and failure are typically more distinguishable. For multi-view trajectories, all views are sampled at the same normalized positions. The sampled images are resized to a height of 360 pixels. For grasp tasks, synchronized views from the same time step are concatenated horizontally. For lift and state-change tasks, synchronized views are provided as separate image inputs in temporal order.

Evidence-based trajectory evaluation. Instructions are routed into three task families: grasp, lift, and state change. Qwen3VL-8B is instructed to extract observable evidence rather than directly predict success or failure. In particular, wrist-camera motion alone is not treated as evidence of object displacement; object motion must be verified in the head view or through clear geometric separation from its support.

The extracted evidence is converted into a binary prediction using deterministic task-specific rules. A grasp requires the target to remain held across at least two late observations. A lift additionally requires visible target displacement in the head or wide view and separation from the original support. A state-change task requires the instructed final state to be visibly completed. A wrong object or visible drop results in failure, and insufficient or null evidence is not interpreted optimistically.

Qwen3VL-8B uses BF16, deterministic decoding, at most 256 generated tokens, and `enable_thinking=False`. The following common prompt is used to extract visual evidence. The output schema is selected according to the routed task family.

Prompt P3: Visual Evidence Extraction

```
Task instruction: {instruction}

Chronological observations follow. At each time step, the labeled views are
synchronized images of the same instant.

Extract observable facts only; do not decide success or failure. Track the target object
across the synchronized views and compare its initial, intermediate, and final states.
The hand_left and hand_right views are cameras mounted on the moving grippers: apparent
target motion, growth, or centering in a hand-camera view may be caused only by camera
motion and is not proof that the object moved. Verify object displacement in the wider
head view or by clear separation from its support and surrounding landmarks. Require
agreement between views when the head view is ambiguous.

Treat wording such as 'attempt to grasp X' as success only if X is actually grasped. For
a grasp-only task, require the target to remain visibly enclosed by the gripper across at
least two late observations; the target need not leave its support. For a lift or pick-up
task, require visible controlled motion of the object with the gripper and separation from
its support. Contact, gripper closure, or temporary occlusion alone is not sufficient.
State the exact time-step transition that proves any claimed motion. For placement,
insertion, folding, pouring, pressing, or turning, inspect whether the corresponding final
state is visible. Use null for a fact that is genuinely hidden in every view. Do not infer
facts from purposeful motion.

For press or click tasks:
A press or click is a momentary action: set required_final_state_visible true when the
instructed arm or gripper visibly reaches and presses the correct button, bell top, or
switch, even if it springs back or the surrounding object never moves. Mere approach
without visible contact or pressing is insufficient.

Return only the schema corresponding to the task family.

// grasp
{"target_held_for_two_late_steps": true,
 "wrong_object_or_drop": false,
 "observations": "...",
 "confidence": 0}

// lift
{"head_view_confirms_target_displacement": true,
 "target_held_for_two_late_steps": true,
 "target_on_original_support_at_end": false,
 "wrong_object_or_drop": false,
 "observations": "...",
 "confidence": 0}

// state_change
{"required_final_state_visible": true,
 "wrong_object_or_drop": false,
 "observations": "...",
 "confidence": 0}

Task instruction: {instruction}
Do not include a verdict.
```

Verification of unsuccessful predictions. When the initial evidence does not establish task completion, we apply a conditional verification procedure. For synchronized multi-view trajectories, we first inspect six higher-resolution observation groups at normalized positions $[0, 0.65, 0.78, 0.88, 0.95, 1]$, with images resized to a height of 480 pixels. This verification reuses the same evidence rubric and task-specific schema.

For applicable state-change cases, we then check whether the already extracted observations logically establish task completion without introducing new visual claims.

Prompt P4: Logical Evidence Reconciliation

```
Check the logical consistency of already-extracted visual observations; do not infer
anything that they do not explicitly state.
Task instruction: {instruction}
```

```
selected: {observations}
first pass: {observations}
high-resolution verification: {observations}
Do these observations themselves directly establish that the instructed task was
completed? A visibly completed momentary action such as pressing a button counts.
Placement, insertion, scooping, pouring, or transfer requires an explicitly observed
completed effect. Approach, contact, purposeful motion, an incomplete attempt, or a
negated completion does not count. Reply only with
{"observations_prove_completion": <true|false>, "reason": "<brief reason>"}
```

Finally, unresolved lift and state-change cases are re-examined using the temporal motion in the complete video. Twelve frames are sampled from each available view and provided as temporally ordered video inputs.

Prompt P5: Temporal Video Verification

```
Re-check the unsuccessful image-based judgment using the temporal motion in each camera
sequence. The views cover the same trajectory. Track the exact instructed target, support,
and destination. For lifting, require controlled target motion with the gripper and visible
separation or changed height relative to the support. For placement or insertion, require
the target to be released in the instructed destination; approach or temporary occlusion
is not completion.
Earlier observation: {observations}

Reply only with observable motion evidence:
{"task_effect_completed": <true|false|null>,
 "observations": "<exact visible transition; at most 30 words>"}
```

A trajectory is counted as correctly evaluated when the final prediction agrees with the recorded success or failure label. For each generation seed, accuracy is computed by pooling the 100 trajectories from RoboTwin and AgiBot; the reported 74% is the mean across five seeds.

C.4 Ablation Protocol

All ablation variants use the same training-data splits, inference configurations, evaluation cases, metric implementations, candidate trajectories, and Qwen3VL-8B procedures as the corresponding main experiments. For the planning ablations, each variant evaluates the same 32 candidate trajectories generated by the fixed $\pi_{0.5}$ and LingBot-VLA checkpoints, matching the $N = 32$ setting in the main planning experiment. The reported planning success rates are averaged over five evaluation runs.

For the no-action-free-data ablation, we exclude the Stage-I action-free corpus entirely while matching the full WorldLine model's total training-compute budget. Consequently, its comparison with the full model controls for total compute and isolates the contribution of access to the action-free data more directly than an unmatched-training baseline.

D Additional Quantitative Results

D.1 Representative-Task Planning Results

Table 4 reports planning performance on a representative subset of RoboTwin tasks. The tasks are selected to cover different manipulation types and interaction patterns, while excluding tasks for which direct execution is saturated at either 0% or 100% for either policy. The subset is not selected according to the magnitude of performance improvement. All methods use the same policy checkpoints, candidate trajectories, rollout budgets, and Qwen3VL-8B selector described in the experimental protocol.

Analysis. The task-level results show that the aggregate planning gains are not driven by a single interaction type. For $\pi_{0.5}$, WorldLine increases the average success rate over the five selected tasks from 53.3% under direct execution to 77.6%, a gain of 24.3 percentage points. The largest improvements occur on Handover Block and Stack Two Blocks, where success increases by 48.5 and 54.5 points, respectively. For LingBot-VLA, WorldLine improves the subset average from 26.4% to 48.0% (+21.6 points), with particularly large gains on Open Laptop (+42.5 points) and Handover

Table 4: Planning success rates on a representative subset of RoboTwin tasks (%). The selected tasks cover different manipulation types. All methods evaluate the same candidate trajectories.

Method	Handover Block	Lift Pot	Open Laptop	Stack Two Blocks	Turn Switch	Average
			$\pi_{0.5}$			
Direct Execution	7.5	83.1	85.6	37.5	52.5	53.3
GE-Sim-V2	32.0	96.0	84.0	56.0	48.0	63.2
OpenDW-0.5	32.0	84.0	80.0	48.0	52.0	59.2
Masked Visual Actions	0.0	100.0	80.0	68.0	56.0	60.8
WorldLine	56.0	80.0	100.0	92.0	60.0	77.6
WorldLine (Causal 4-Step)	8.0	84.0	100.0	64.0	72.0	65.6
			LingBot-VLA			
Direct Execution	10.0	4.4	57.5	3.8	56.3	26.4
GE-Sim-V2	8.0	0.0	76.0	0.0	44.0	25.6
OpenDW-0.5	0.0	0.0	32.0	0.0	88.0	24.0
Masked Visual Actions	8.0	0.0	20.0	4.0	60.0	18.4
WorldLine	40.0	4.0	100.0	20.0	76.0	48.0
WorldLine (Causal 4-Step)	8.0	0.0	88.0	36.0	88.0	44.0

Block (+30.0 points). These improvements span object transfer, articulated-object manipulation, and multi-object stacking, supporting the use of predicted outcomes for candidate selection across distinct interaction patterns.

The causal four-step variant retains substantial average gains over direct execution, reaching 65.6% for $\pi_{0.5}$ and 44.0% for LingBot-VLA. Its benefits are strongest on tasks with visually identifiable state transitions, including Open Laptop, Stack Two Blocks, and Turn Switch. Performance is not uniform across all tasks: neither WorldLine variant improves Lift Pot for LingBot-VLA, and the causal variant provides little improvement on Handover Block for $\pi_{0.5}$. This variation indicates that planning quality depends on whether the predicted rollout exposes sufficiently discriminative evidence for ranking the candidate actions.

E Additional Qualitative Results

Supplementary video visualizations are available on an anonymous project page.[1]

E.1 Additional Comparisons on AgiBotWorld and DROID

Figure 8 provides additional comparisons between WorldLine and prior video-generation and action-conditioned simulation methods. We include held-out AgiBotWorld examples covering both successful and unsuccessful interactions, together with DROID examples involving unseen environments, camera configurations, and robot embodiments. Each example uses the same initial observation and action sequence for all methods. The visualization presents the ground-truth future alongside the corresponding predictions and enlarges regions containing action-critical robot–object interactions.

E.2 Three-View WorldLine Predictions on AgiBotWorld

Figure 9 provides additional comparisons between WorldLine predictions and ground-truth trajectories from held-out AgiBotWorld scenes. For each example, synchronized head, left-wrist, and right-wrist views are shown at multiple time steps. This layout enables a detailed examination of whether the predicted robot motion, contact, and object-state changes remain temporally aligned across the global workspace view and the two moving wrist cameras.

E.3 Three-View WorldLine Predictions on DROID

Figure 10 presents additional comparisons between WorldLine predictions and recorded DROID trajectories. DROID is not used during WorldLine training, adaptation, or checkpoint selection.

[1] Supplementary Videos. After opening the anonymous page, turn `JS Off` to `JS On` to view the videos.

For each example, we show temporally aligned head, left-wrist, and right-wrist views, allowing the predicted robot motion and object evolution to be examined from both the global workspace camera and the moving end-effector cameras. The three-view layout also reveals whether the generated interaction remains consistent across synchronized viewpoints under an unseen robot embodiment.

E.4 Long-Horizon Causal Rollouts

Figure 11 compares long-horizon Causal WorldLine rollouts with the corresponding ground-truth trajectories. Each example presents synchronized head, left-wrist, and right-wrist observations sampled across the generated sequence. The visualization exposes how robot motion, object-state changes, and cross-view consistency evolve across successive block-autoregressive generation steps.

E.5 Qualitative Ablations

Figure 12 visualizes the effects of the main data, action-control, and training-strategy ablations on three AgiBotWorld examples. Each comparison uses the same initial observation and action sequence and places the ground-truth future, full WorldLine prediction, and selected ablation variants side by side. The examples focus on differences in commanded robot motion, contact accuracy, object displacement, and failure-state generation that may not be fully captured by aggregate image metrics.

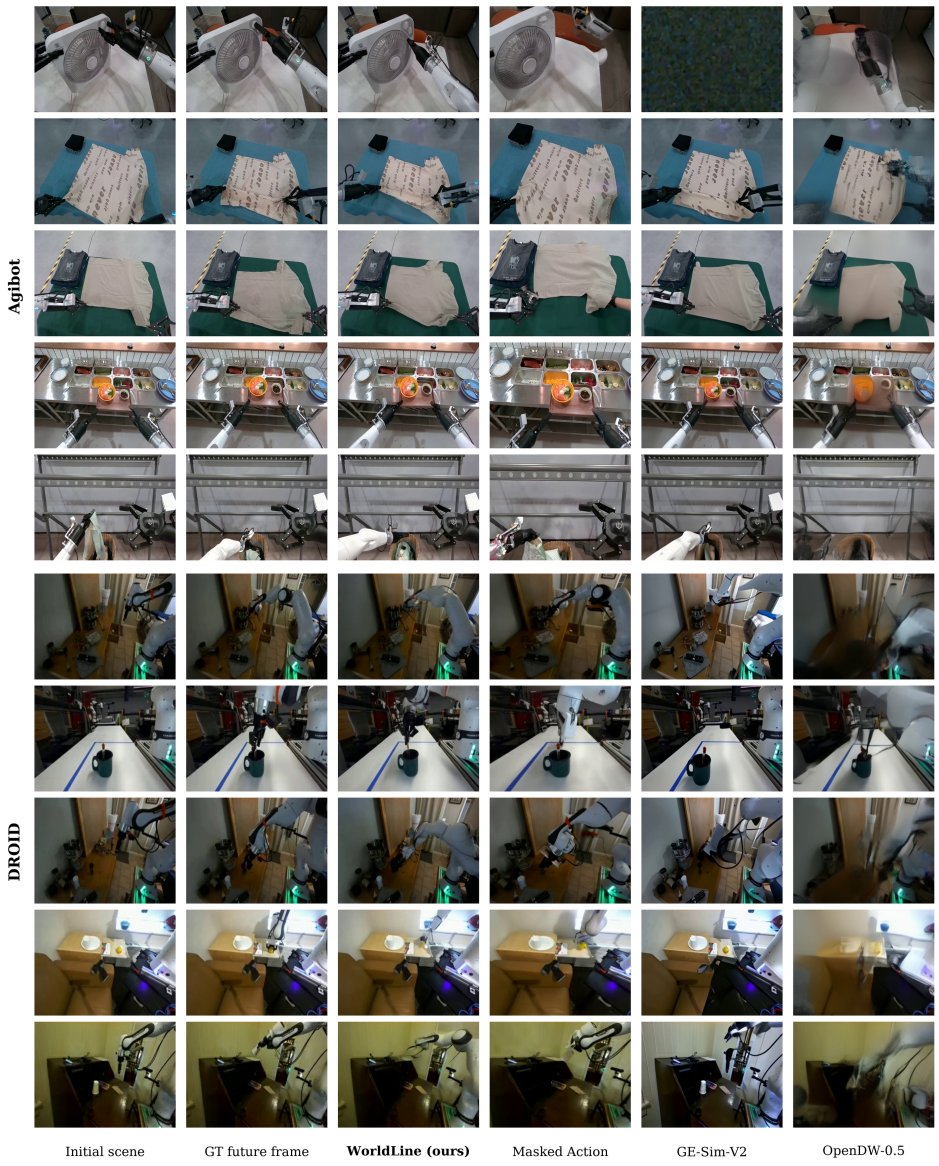

Figure 8: **Additional qualitative comparisons on AgiBotWorld and DROID.** Each example shows the initial observation, action condition, ground-truth future, predictions from comparison methods, and the WorldLine prediction. AgiBotWorld examples are drawn from held-out scenes and include both successful and unsuccessful interactions, while DROID examples test transfer to unseen environments, viewpoints, and robot embodiments. Enlarged regions focus on commanded robot motion, contact, and the resulting object-state changes.

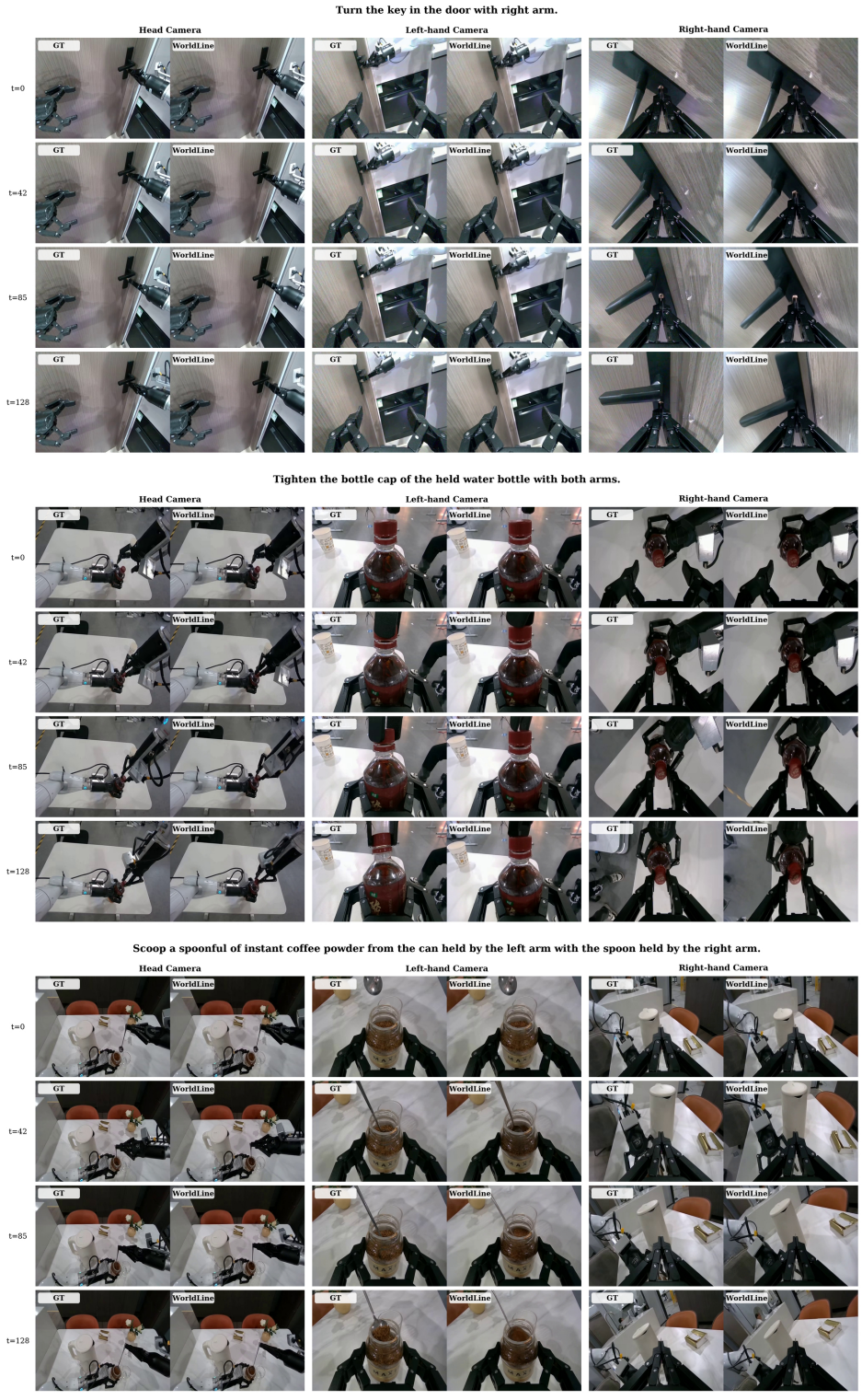

Figure 9: **Three-view comparison between WorldLine predictions and ground-truth AgiBot-World trajectories.** Each example presents synchronized head, left-wrist, and right-wrist observations at multiple time steps from a held-out AgiBotWorld scene. The comparison shows robot motion, contact, and object-state evolution across the global and wrist-mounted camera views.

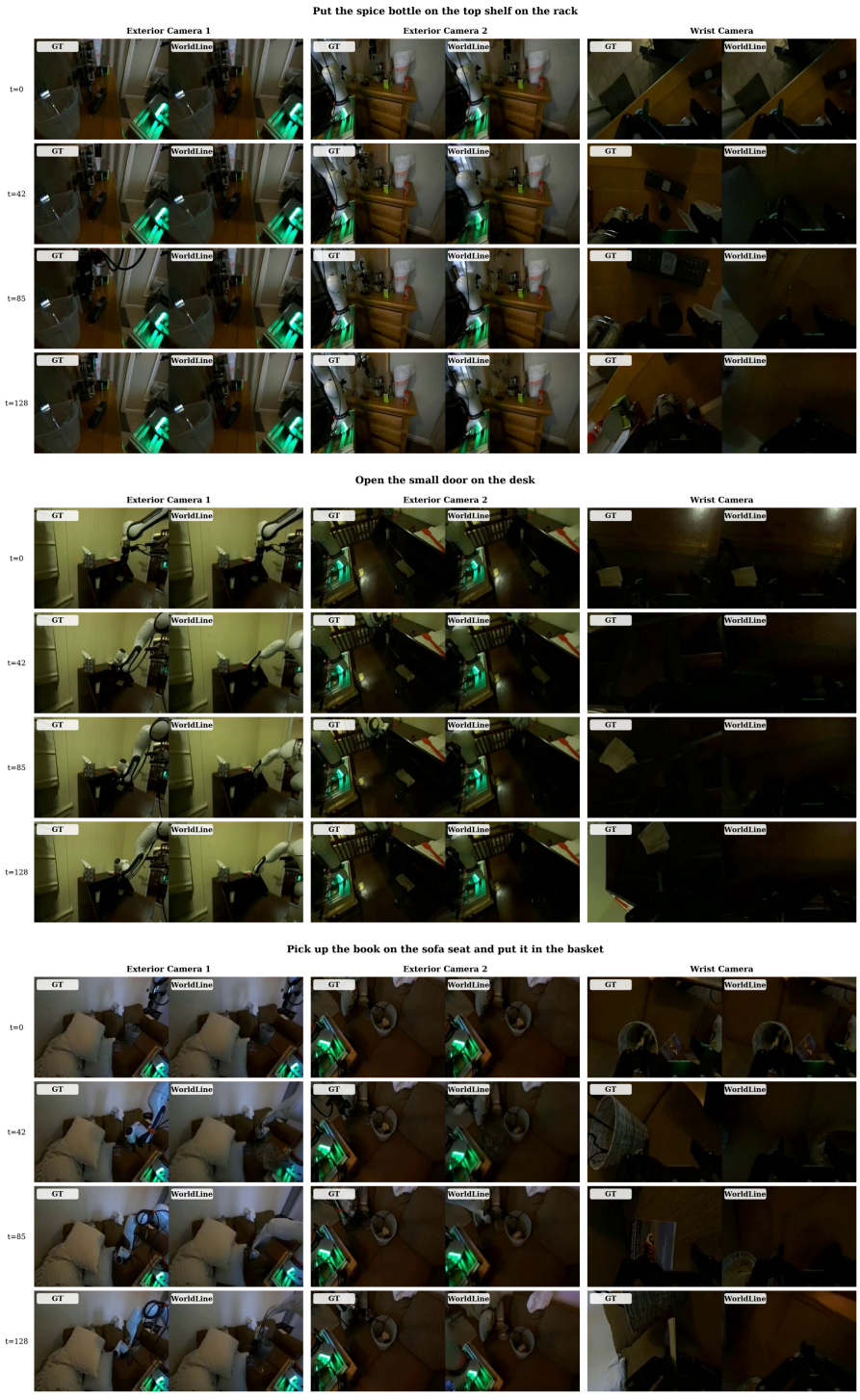

Figure 10: **Three-view comparison between WorldLine predictions and ground-truth DROID trajectories.** Each example presents synchronized head, left-wrist, and right-wrist observations at multiple time steps. WorldLine is evaluated directly on DROID without training or adaptation on DROID data. The comparison exposes the temporal alignment of robot motion, contact, and object-state evolution across global and wrist-mounted camera views.

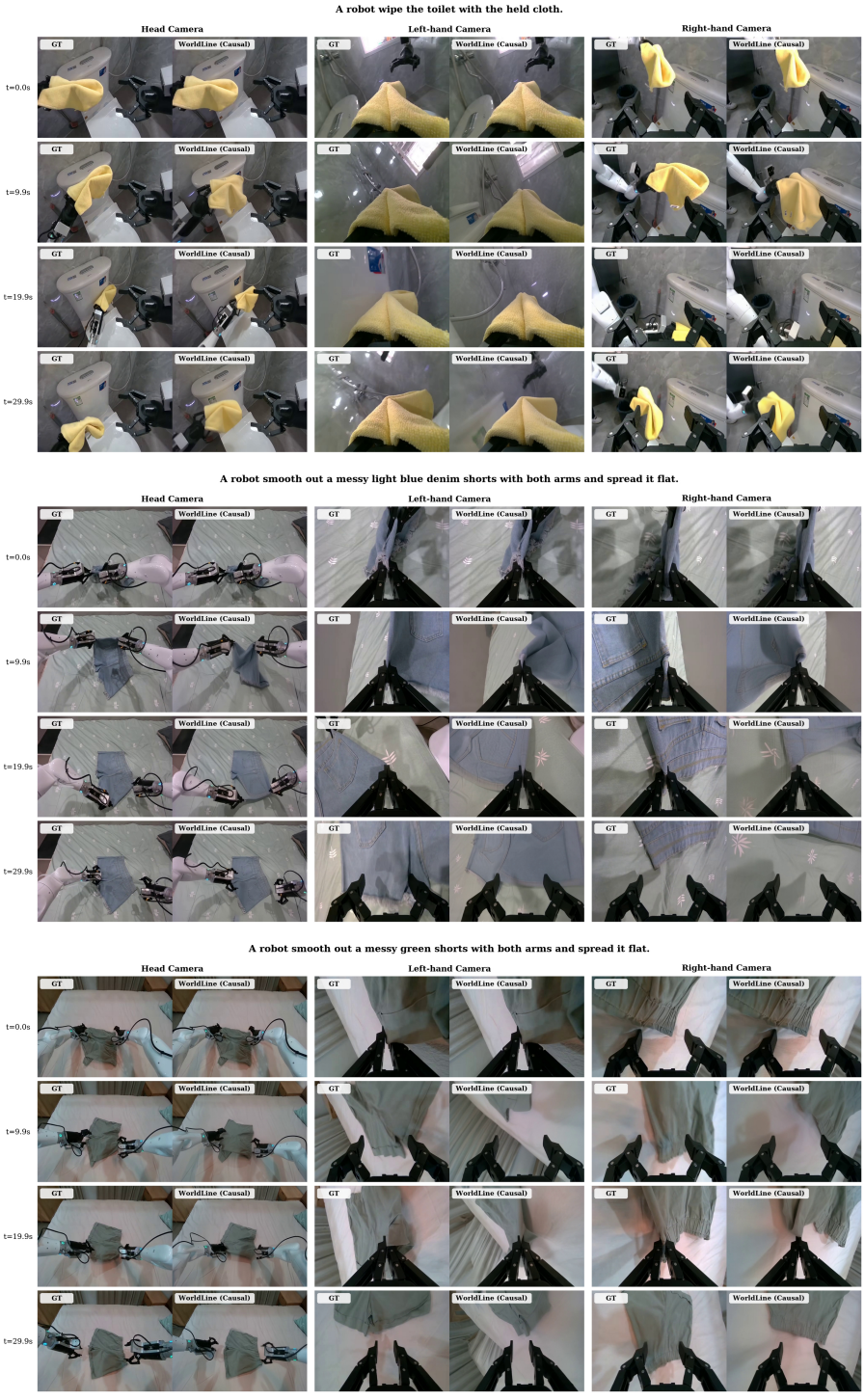

Figure 11: **Long-horizon three-view rollouts generated by Causal WorldLine.** We compare the causal predictions with the corresponding ground-truth trajectories using synchronized head, left-wrist, and right-wrist observations sampled across the rollout. The visualization highlights long-term robot motion, object-state evolution, temporal continuity, and cross-view consistency under block-autoregressive generation.

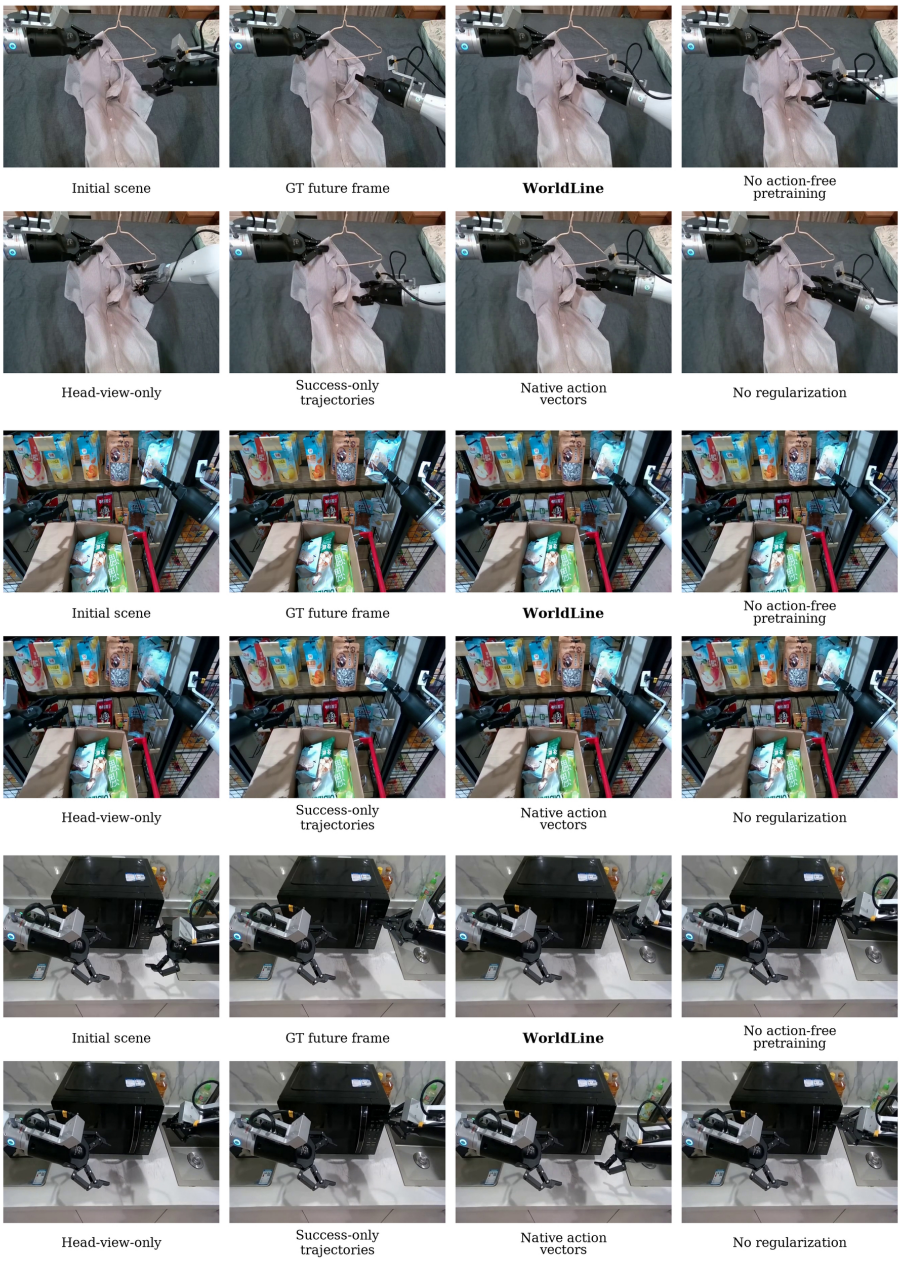

Figure 12: **Qualitative effects of the main WorldLine components on AgiBotWorld.** Across three examples, we compare the full model with representative ablations of the training data, action-conditioning interface, and relational regularization. All variants receive the same initial observation and action sequence. Enlarged regions highlight differences in robot pose, contact, object motion, and the generation of unsuccessful interaction outcomes.

F LIMITATIONS AND BROADER IMPACT

F.1 LIMITATIONS

Although WorldLine generalizes across unseen scenes, environments, and robot embodiments, its coverage is still shaped by the diversity of interactions in the training data. Rare manipulation patterns, deformable objects, unusual camera configurations, and substantially longer action horizons may benefit from broader and more targeted supervision. In downstream planning, the achievable improvement is also bounded by the diversity of trajectories proposed by the underlying policy and the quality of the multimodal selector. Future work may extend WorldLine with richer interaction data, longer-horizon training, uncertainty-aware prediction, and robot-specific outcome evaluators.

F.2 BROADER IMPACT

WorldLine can reduce the amount of repeated physical interaction required for robot policy development by enabling candidate behaviors to be visualized, evaluated, and selected before execution. This capability may lower data collection costs, reduce hardware wear, support failure analysis, and make cross-embodiment robot learning more accessible. For deployment in safety-sensitive settings, simulator-based evaluation should complement standard real-world validation and human oversight. The collection and use of large-scale robot videos should also follow appropriate privacy, licensing, and data-governance practices, while continued improvements in training and inference efficiency can reduce computational cost.