SIMFORCING: DISTILLING SIMULATION MOTION PRIORS INTO REAL-DOMAIN ROBOT WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Action-conditioned robot world models must respond precisely to robot trajectories while preserving realistic visual dynamics, yet learning both from heterogeneous robot videos remains challenging. Simulation offers structured motion supervision, but appearance differences hinder direct transfer, and inaccurate simulation predictions can misguide real-video generation. We present SimForcing, a simulation-guided framework that uses simulation both as a source of transferable motion knowledge and as a controllable reference for prediction. First, we transfer motion knowledge from a simulation teacher through latent-motion distillation, aligning temporal changes in latent space to internalize motion priors while mitigating the influence of appearance differences. Second, we introduce multi-block simulation conditioning with condition dropout to exploit predicted simulation trajectories without relying excessively on their accuracy. Our simulation-conditioning classifier-free guidance scheme unifies these two ideas by balancing predictions based on internalized motion knowledge with those additionally guided by simulation latents. The jointly trained student generates both simulation conditions and real-domain videos, requiring no additional world model at inference. On Bridge, SimForcing achieves the best PSNR, SSIM, LPIPS, and FVD among the compared methods without external embodied pretraining. Evaluation on InternData-A1 further supports its applicability across robot datasets. Moreover, using our trained world model to initialize a vision-language-action model improves LIBERO success, suggesting its utility for downstream policy learning.

1 INTRODUCTION

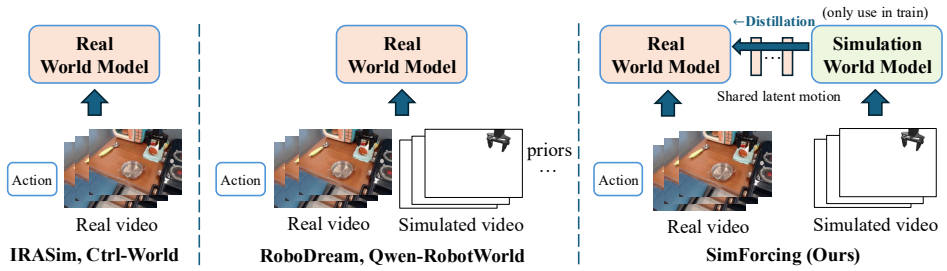


Figure 1: **Comparison of training paradigms for action-conditioned robot world models. Left:** Direct training on real videos and corresponding actions (Zhu et al., 2025; Guo et al., 2026b). **Middle:** Training on real videos augmented with explicit visual priors, such as simulated videos, scene priors, and object priors (Zhang et al., 2026; Ye et al., 2026). **Right:** SimForcing learns a simulation world model from synthetic rollouts and distills its motion priors into a real-world model through latent-space motion supervision, without requiring additional scene or object decomposition.

Recent advances in large-scale video generation and world foundation models (Wan et al., 2025; Chi et al., 2025; Ali et al., 2025; Ma et al., 2026), have expanded the potential of learned visual dynamics for embodied intelligence. Beyond video synthesis, these models can generate training experiences for robot policies (Jang et al., 2025) and support the improvement of vision-language-action (VLA) models through synthetic rollouts (Guo et al., 2026a). Benchmarks such as Worl-

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.

dArena (Shang et al., 2026) further assess their utility as data engines, policy evaluators, and action planners. Despite this progress, visually convincing predictions do not necessarily ensure reliable physical interactions or strong downstream performance (Shang et al., 2026). For robotic manipulation, a key challenge remains: learning precise responses to continuous action trajectories while preserving realistic appearance and temporally coherent scene dynamics.

Existing approaches address this problem through different training paradigms, as illustrated in Fig. 1. One approach learns video prediction directly from real robot videos and their associated actions such as IRASim (Zhu et al., 2025) and Ctrl-World (Guo et al., 2026b). This provides direct supervision for action-conditioned generation, but learning precise action responses and complex visual dynamics jointly from real videos remains challenging. Another approach introduces an explicit visual structure. For example, RoboDream (Ye et al., 2026) anchors generation to rendered robot motion and conditions on scene and object priors. Such compositional inputs provide additional control, while requiring the corresponding priors to be constructed and supplied. Qwen-RobotWorld (Zhang et al., 2026) also needs scene video as prior. However, the potential of simulation data to improve real-video prediction without additional priors remains underexplored.

Simulation provides structured supervision for action-dependent motion, complementing the appearance and interaction diversity of real videos. Transferring this knowledge nevertheless presents two challenges. First, appearance differences complicate direct matching between simulated and real observations. Second, predicted simulation trajectories are imperfect; a real-domain model that relies too heavily on them may inherit their errors. These challenges motivate transferring motion knowledge through temporal differences in latent space to mitigate the influence of appearance differences, while controlling the model's reliance on potentially inaccurate simulation predictions.

We present **SimForcing**, a novel world modeling framework that addresses these challenges through motion knowledge transfer and controllable simulation guidance. To transfer motion knowledge across domains with different appearances, we first train a mixture-of-transformers world model on synthetic rollouts with trajectory augmentation, learning action-dependent motion from diverse robot trajectories. The resulting model initializes the student and serves as a frozen teacher. Our key transfer mechanism is latent-motion distillation, which aligns temporal differences between adjacent teacher and student video latents rather than directly matching their appearances. This motion-focused supervision internalizes the teacher's simulation-derived priors within the student, directly enhancing its ability to predict real-world video dynamics. Joint flow-matching and motion-distillation objectives on paired simulated and real videos further develop prediction capabilities in both domains within the same student, without requiring additional scene or object decomposition.

To benefit from explicit simulation predictions without becoming overly dependent on their accuracy, we introduce multi-block simulation conditioning with condition dropout. Injecting simulation latents into multiple video transformer blocks provides a motion reference throughout generation, while latent corruption and condition dropout train the student to handle imperfect or absent simulation guidance. At inference, we introduce a simulation-conditioning classifier-free guidance (CFG) that combines predictions from branches with and without simulation latent input, balancing explicit simulation guidance with the student's internalized motion priors. This design enables the model to exploit reliable simulation predictions while limiting the influence of inaccurate ones. The jointly trained student generates both the simulation conditions and real-domain videos, requiring no additional world model at inference.

Experiments on Bridge show that SimForcing outperforms Ctrl-World, EnerVerse-AC, and a GeniWorld reimplementation across PSNR, SSIM, LPIPS, and FVD, improving both reconstruction fidelity and perceptual similarity. It also achieves lower LPIPS than Cosmos-Predict2.5 with substantially faster inference. Evaluation on InternData-A1 supports its applicability across robot datasets, while ablations demonstrate the complementary benefits of motion distillation and simulation conditioning. Beyond video prediction, initializing a downstream VLA with the pretrained world-model experts improves average LIBERO success from 82.6% to 89.2% under action-only training relative to Wan DiT initialization, with gains also observed in the other training settings.

Our contributions are threefold:

- We introduce SimForcing, a robot world model framework for transferring motion knowledge from simulation to real-world video prediction in latent space, internalizing simulation-derived motion priors within the world model.

- We propose a controllable mechanism for exploiting simulation predictions while limiting overreliance on inaccurate guidance. Our simulation-conditioning CFG scheme combines this explicit guidance with the motion knowledge internalized through distillation.
- We outperform state-of-the-art methods on Bridge across PSNR, SSIM, LPIPS, and FVD, and demonstrate the framework's applicability to InternData-A1 dataset. LIBERO experiments preliminarily validate the pretrained video expert's utility for policy learning.

2 Related Work

Embodied and action-conditioned video world models. Large-scale video generators provide increasingly capable visual priors for world modeling (Wan et al., 2025; Agarwal et al., 2025; Ali et al., 2025; Chi et al., 2025), with recent efforts extending pretraining and adaptation toward embodied interaction. Qwen-RobotWorld (Zhang et al., 2026) uses natural-language actions to predict future visual trajectories across manipulation, navigation, and driving, while LingBot-Video (Ma et al., 2026) scales mixture-of-experts video pretraining with robot-oriented data. DreamGen (Jang et al., 2025) adapts video generators to target robot embodiments and recovers action labels through inverse dynamics or latent action models, producing synthetic trajectories for policy learning. These advances establish the value of video priors for embodied intelligence. Our work complements this direction by transferring action-dependent motion knowledge from simulation into a real-domain video generator through latent-space supervision.

Action-conditioned world models connect robot controls to predicted visual outcomes. IRASim (Zhu et al., 2025) learns from historical observations and continuous actions, while AVID (Rigter et al., 2024) adds an action-conditioned adapter to a frozen video diffusion model. Dynamic World Simulation (He et al., 2026) combines action conditioning with a motion-reinforced objective. EnerVerse-AC (Jiang et al., 2025) uses multi-level action conditioning for multiview generation, and Ctrl-World (Guo et al., 2026b) supports policy-in-the-loop simulation with frame-level control and temporal memory. Other methods encode control as spatial visual signals: RoboMaster (Fu et al., 2026) uses collaborative robot–object trajectories, ABot-PhysWorld (Chen et al., 2026b) combines spatial action injection with physics-aligned training, and OSCAR (Wu & Gao, 2026) uses rendered kinematic skeletons across embodiments. FlowWAM (Chen et al., 2026a) adopts optical flow for video prediction and policy learning, while RynnWorld-Teleop (Zhao et al., 2026) uses hand-pose conditioning for real-time teleoperation. These methods improve action representation and rollout consistency. Our work complements them by internalizing simulation-derived motion knowledge through latent-motion distillation and balances it with simulation guidance.

Simulation supervision and real-domain adaptation. Simulation supports robot learning through data augmentation, dynamics transfer, and visual conditioning. Sim-and-Real Co-Training (Maddukuri et al., 2025) mixes synthetic and real demonstrations for policy learning, while RoboTwin 2.0 (Chen et al., 2025) scales synthetic data generation with structured domain randomization. ReDRAW (Lanier et al., 2025) adapts simulation-pretrained latent dynamics through residual corrections, and Simulation Distillation (Levy et al., 2026) transfers simulator priors into world models for real-world adaptation and planning. For video generation, Cosmos-Transfer1 (Alhaija et al., 2025) supports sim-to-real visual translation through spatial controls. GeniWorld (Gu et al., 2026) renders numerical actions into visual conditions, while RoboDream (Ye et al., 2026) combines rendered robot motion with scene and object priors. Our method internalizes simulation-derived motion knowledge through latent-motion distillation. Our simulation-conditioning CFG scheme balances this knowledge with explicit simulation guidance to limit reliance on inaccurate predictions.

3 Method

3.1 Problem formulation

Let $\mathbf{x}_{0:T} = (\mathbf{x}_0, \ldots, \mathbf{x}_T)$ denote a robot video and let $\mathbf{a}_{0:T-1}$ denote its aligned low-level action trajectory. At each time step t, the proprioceptive state is synchronized with $\mathbf{x}_t$, and $\mathbf{a}_t$ is the command applied over the transition from $\mathbf{x}_t$ to $\mathbf{x}_{t+1}$. Each action comprises an end-effector translation

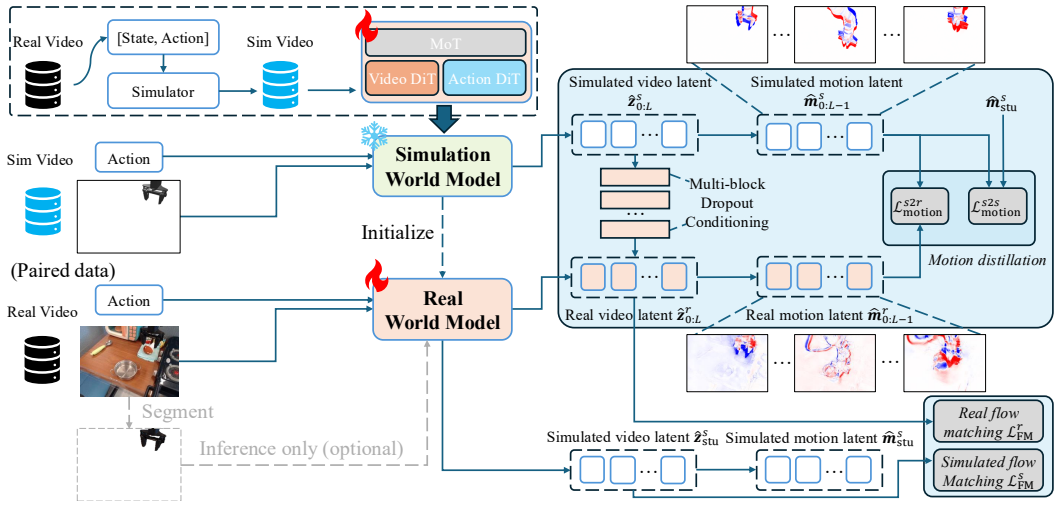


Figure 2: **Overview of SimForcing. Simulation world model training:** A mixture-of-transformers world model learns action-conditioned motion from augmented synthetic rollouts. **Sim-to-real distillation:** The pretrained model initializes a student and serves as a frozen teacher. Latent-motion distillation internalizes motion priors, while multi-block dropout conditioning provides simulation guidance. Joint training enables the final student to generate both simulation conditions and real videos, with CFG balancing explicit guidance and internalized knowledge at inference.

increment, a rotation increment represented as an axis-angle vector, and a gripper command. The proprioceptive state comprises the end-effector position, orientation in Euler angles, and gripper state. Thus, a sequence of $T+1$ video frames is paired with T actions and T proprioceptive states.

Given the initial observation and future actions, our goal is to model

$$p_\theta(\mathbf{x}_{1:T} \mid \mathbf{x}_0, \mathbf{a}_{0:T-1})\,. \tag{1}$$

Proprioceptive inputs are used throughout but omitted from the notation for simplicity. The model predicts future visual observations; it does not predict actions during world-model pretraining. We use superscripts s and r for simulated and real samples. A frozen video autoencoder E compresses the video $\mathbf{x}_{0:T}$ into a latent sequence $\mathbf{z}_{0:L} = E(\mathbf{x}_{0:T})$, where $\mathbf{z}_0$ represents the initial observation and L is the number of future latent time steps. Due to temporal compression, $L < T$.

3.2 Simulation World Model Training

Learning motion responses and visual appearance jointly from real videos is challenging. As shown in Fig. 2, SimForcing first learns action-conditioned robot motion in simulation, then transfers this knowledge to real-video prediction through latent-motion distillation and simulation conditioning.

The backbone combines a Video DiT and an Action DiT through Mixture-of-Transformers (MoT) attention. The Video DiT processes noisy future-video tokens and clean initial-frame tokens, while the Action DiT encodes $\mathbf{a}_{0:T-1}$ as conditioning tokens. Joint attention connects the two streams, with modality-specific feed-forward blocks processing their respective features. Only the video branch receives a diffusion prediction loss.

We construct synthetic action–video pairs by replaying trajectories from real demonstrations in a simulator. Each rollout starts from the corresponding initial robot configuration and renders motion aligned with the action sequence.

To increase motion diversity, we generate three perturbed variants of each trajectory. We write each action as $\mathbf{a}_t = (\mathbf{q}_t, g_t)$, where $\mathbf{q}_t \in \mathbb{R}^6$ contains the translation and axis-angle rotation increments, and g_t is the gripper command. The perturbed action is

$$\widetilde{\mathbf{a}}_t = \left(\mathrm{clip}\left((\mathbf{q}_t + \mathbf{b}) \odot \mathbf{s} + \boldsymbol{\eta}_t\right), g_t\right), \qquad \boldsymbol{\eta}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}), \tag{2}$$

where the bias $\mathbf{b}$ and element-wise scale $\mathbf{s}$ are uniformly sampled once per rollout, while $\boldsymbol{\eta}_t$ is sampled independently at each step. Each perturbed trajectory is replayed from the same initial robot

configuration to render a video paired with the executed actions. For simplicity, we subsequently use $\mathbf{a}_{0:T-1}$ to denote the executed action sequence, including perturbed variants.

Given a synthetic rollout consisting of $T+1$ video frames $\mathbf{x}^s_{0:T}$ and T aligned actions $\mathbf{a}_{0:T-1}$, let $\mathbf{z}^s_0$ denote the initial observation latent and $\mathbf{z}^s = \mathbf{z}^s_{1:L}$ denote the clean future latent sequence. For Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and flow time $\tau \in [0, 1]$, we define the interpolation path and target velocity as

$$\mathbf{z}^s(\tau) = (1-\tau)\mathbf{z}^s + \tau\boldsymbol{\epsilon}, \qquad \mathbf{u}^s = \boldsymbol{\epsilon} - \mathbf{z}^s. \tag{3}$$

The simulation world model S_θ predicts the target velocity conditioned on the initial observation latent and action tokens, with proprioceptive inputs omitted from the notation as described above. We train the entire video-action backbone using the flow-matching objective

$$\mathcal{L}^s_{\text{vid}} = \mathbb{E}\left[\|S_\theta(\mathbf{z}^s(\tau), \tau \mid \mathbf{z}^s_0, \mathbf{a}_{0:T-1}) - \mathbf{u}^s\|_2^2\right]. \tag{4}$$

The expectation is taken over synthetic rollouts, Gaussian noise, and sampled flow times.

The resulting simulation world model serves two purposes: it provides an initialization for action-conditioned video generation in the real domain and serves as a frozen teacher that supplies dynamics supervision during real-domain adaptation.

3.3 Sim-to-Real Distillation Training

The core idea of this part is to transfer motion priors from simulation to the real domain through latent-space distillation. During simulation pretraining, the world model learns to predict future robot motion from the initial observation and action sequence. By excluding object interactions and background complexity, this training focuses on the robot's action-conditioned motion dynamics. The learned latent representations then provide motion supervision and conditioning signals for training the real-domain world model.

Motion Distillation. After simulation pretraining, we freeze the teacher parameters $\bar{\theta}$ and initialize the student's video–action backbone from the teacher. The student parameters θ are optimized during subsequent training, while $\bar{\theta}$ remains fixed. The newly introduced conditioning projections are initialized to zero. We represent latent motion as the difference between temporally adjacent video latents:

$$\mathbf{m}_\ell = \mathbf{z}_{\ell+1} - \mathbf{z}_\ell, \qquad \ell = 0, \ldots, L-1. \tag{5}$$

Here, ℓ indexes the temporally compressed latent sequence, and $\mathbf{z}_0$ is the initial observation latent.

To obtain motion supervision on the fly during training, we follow the flow-matching training procedure, using a single noise injection and velocity prediction for each domain. Specifically, we construct the noisy latents using the same Gaussian noise $\boldsymbol{\epsilon}$ and flow time τ:

$$\mathbf{z}^d(\tau) = (1-\tau)\mathbf{z}^d + \tau\boldsymbol{\epsilon}, \qquad d \in \{s, r\}. \tag{6}$$

We then perform a single forward pass through the frozen simulation teacher $S_{\bar{\theta}}$ and the trainable real-domain student R_θ to predict their respective velocities:

$$\begin{aligned} \hat{\mathbf{u}}^s &= S_{\bar{\theta}}(\mathbf{z}^s(\tau), \tau \mid \mathbf{z}^s_0, \mathbf{a}_{0:T-1}), \\ \hat{\mathbf{u}}^r &= R_\theta(\mathbf{z}^r(\tau), \tau \mid \mathbf{z}^r_0, \mathbf{a}_{0:T-1}; \mathbf{c}^s, g). \end{aligned} \tag{7}$$

Here, $\mathbf{c}^s$ and g denote the simulation condition and its dropout gate, defined below.

Following the interpolation path in Eq. 3, we estimate the clean future latent from each predicted velocity:

$$\hat{\mathbf{z}}^d = \mathbf{z}^d(\tau) - \tau\hat{\mathbf{u}}^d, \qquad d \in \{s, r\}. \tag{8}$$

We compute the predicted latent motion as the difference between adjacent latents:

$$\hat{\mathbf{m}}^d_\ell = \hat{\mathbf{z}}^d_{\ell+1} - \hat{\mathbf{z}}^d_\ell, \qquad \ell = 0, \ldots, L-1, \quad d \in \{s, r\}. \tag{9}$$

Here, $\hat{\mathbf{z}}^d_0 = \mathbf{z}^d_0$ is the clean initial observation latent, and $\hat{\mathbf{m}}^d = (\hat{\mathbf{m}}^d_0, \ldots, \hat{\mathbf{m}}^d_{L-1})$ denotes the predicted motion sequence. We then minimize the squared L_2 distance between the student's and teacher's latent motion sequences:

$$\mathcal{L}^{s2r}_{\text{motion}} = \mathbb{E}\left[\|\hat{\mathbf{m}}^r - \hat{\mathbf{m}}^s\|_2^2\right]. \tag{10}$$

The simulation teacher remains frozen, and gradients are propagated only through the real-domain student. This objective constrains the predicted motion while encouraging temporal consistency in the background, helping suppress background flickering as illustrated in Fig. 2.

Multi-block dropout conditioning. To complement the motion priors distilled into the student, we introduce multi-block dropout conditioning, which provides simulation latents as an optional source of motion guidance. Specifically, we inject the simulation latent into every video transformer block through an independent, zero-initialized 3D convolutional projection. Each projection maps the simulation latent to features aligned with the video tokens, which are added before the feed-forward network:

$$\mathbf{h}_j \leftarrow \mathbf{h}_j + g\, P_j(\mathbf{c}^s), \qquad g \sim \text{Bernoulli}(p), \tag{11}$$

where $\mathbf{h}_j$ denotes the video tokens in block j, P_j is its condition projection, and $\mathbf{c}^s$ is the simulation latent condition. The gate g is sampled independently for each training sample and shared across all blocks, with p denoting the condition retention probability. This design supplies motion information throughout the network. Zero initialization preserves the pretrained backbone's behavior at the start of training, allowing the conditioning pathways to be learned gradually.

To improve robustness to imperfect simulation predictions, we stochastically corrupt the ground-truth simulation latent during training:

$$\mathbf{c}^s = \alpha \mathbf{z}^s + (1-\alpha)\sigma_c \boldsymbol{\eta}, \qquad \boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{12}$$

where α and σ_c are sampled per example, and $\boldsymbol{\eta}$ is independent of the flow-matching noise. This corruption is applied with a fixed probability; otherwise, we use the clean simulation latent. It affects only the conditioning input, leaving the flow-matching and teacher supervision targets unchanged.

Condition dropout prevents the student from relying excessively on simulation latents. When $g = 0$, all simulation-conditioning residuals are disabled, requiring the student to predict future observations using the motion knowledge distilled into its own parameters. When $g = 1$, the student additionally receives simulation guidance at every video block. Both paths retain the initial observation, actions, and motion distillation remains active for all samples regardless of the gate.

To further exploit paired data and consolidate prediction in both domains within a single model, we train the student on both simulation and real videos. The simulation branch uses the simulation initial observation and actions without simulation latent conditioning, while the real branch uses the dropout conditioning described above. Both branches share the same student parameters.

For joint training, let $\hat{\mathbf{u}}^d_{\text{stu}}$ denote the student's velocity prediction in domain $d \in \{s, r\}$. The real-domain prediction is $\hat{\mathbf{u}}^r_{\text{stu}} = \hat{\mathbf{u}}^r$ from Eq. 7, while the simulation-domain prediction disables simulation conditioning:

$$\hat{\mathbf{u}}^s_{\text{stu}} = R_\theta(\mathbf{z}^s(\tau), \tau \mid \mathbf{z}^s_0, \mathbf{a}_{0:T-1}; g = 0)\,. \tag{13}$$

We apply flow matching in both domains:

$$\mathcal{L}^d_{\text{FM}} = \mathbb{E}\left[\left\|\hat{\mathbf{u}}^d_{\text{stu}} - (\boldsymbol{\epsilon} - \mathbf{z}^d)\right\|_2^2\right], \qquad d \in \{s, r\}. \tag{14}$$

We also estimate the student's clean simulation latents as $\hat{\mathbf{z}}^s_{\text{stu}} = \mathbf{z}^s(\tau) - \tau \hat{\mathbf{u}}^s_{\text{stu}}$. Using $\hat{\mathbf{z}}^s_{\text{stu},0} = \mathbf{z}^s_0$, we compute the motion sequence $\hat{\mathbf{m}}^s_{\text{stu}}$ by adjacent differences and align it with the frozen teacher:

$$\mathcal{L}^{s2s}_{\text{motion}} = \mathbb{E}\left[\|\hat{\mathbf{m}}^s_{\text{stu}} - \hat{\mathbf{m}}^s\|_2^2\right]. \tag{15}$$

Here, $\hat{\mathbf{m}}^s$ is the frozen teacher's motion target. Together with real-domain motion distillation, this objective helps preserve the student's simulation prediction capability during joint training.

The complete training objective comprises four terms:

$$\mathcal{L} = \lambda^r_{\text{FM}}\mathcal{L}^r_{\text{FM}} + \lambda^s_{\text{FM}}\mathcal{L}^s_{\text{FM}} + \lambda^{s2r}_{\text{motion}}\mathcal{L}^{s2r}_{\text{motion}} + \lambda^{s2s}_{\text{motion}}\mathcal{L}^{s2s}_{\text{motion}}. \tag{16}$$

The teacher is used only during training. At inference, the student can generate real-domain videos using the following guidance method, eliminating the need for an additional simulation world model.

Table 1: Action-conditioned video prediction on the Bridge (Walke et al., 2023) validation set. Gray rows denote embodied pretrained models. Time denotes inference time per sample. Best results among models without embodied pretraining are in bold.

Model	Time (s) ↓	PSNR ↑	SSIM ↑	FID ↓	FVD ↓	LPIPS ↓
With embodied pretraining						
IRASim (Zhu et al., 2025)	25.9	18.062	0.753	47.27	333.92	0.2102
Cosmos-Predict2.5 (Ali et al., 2025)	20.5	27.216	0.880	23.74	119.38	0.0996
Without embodied pretraining						
Ctrl-World (Guo et al., 2026b)	16.9	20.928	0.761	46.67	235.57	0.1674
EnerVerse-AC (Jiang et al., 2025)	16.6	22.599	0.797	28.67	181.55	0.0933
Wan2.2-TI2V-5B (SFT) (Wan et al., 2025)	**1.8**	22.751	0.838	23.97	168.71	0.0926
GeniWorld (Gu et al., 2026)	4.1	23.436	0.845	**23.62**	148.74	0.0799
SimForcing (Ours)	4.1	**25.077**	**0.858**	25.21	**137.49**	**0.0673**

Simulation-conditioning classifier-free guidance. At inference, we use SAM3 (Carion et al., 2026) to segment the robot arm in the initial real observation and encode the masked observation as $\mathbf{z}_0^s$. Conditioned on $\mathbf{z}_0^s$ and the action trajectory, the student R_θ first generates simulation latents with simulation conditioning disabled. These latents form the condition $\mathbf{c}^s$, which remains fixed throughout real-video sampling. We then evaluate the same student with the conditioning pathways enabled and disabled:

$$\begin{aligned} \hat{\mathbf{u}}_{\text{cond}} &= R_\theta(\mathbf{z}^r(\tau), \tau \mid \mathbf{z}_0^r, \mathbf{a}_{0:T-1}; \mathbf{c}^s, g = 1)\,, \\ \hat{\mathbf{u}}_{\text{uncond}} &= R_\theta(\mathbf{z}^r(\tau), \tau \mid \mathbf{z}_0^r, \mathbf{a}_{0:T-1}; g = 0)\,. \end{aligned} \tag{17}$$

Here, $\mathbf{z}^r(\tau)$ denotes the current real-video latent state during sampling. Both branches retain the initial observation and action trajectory. The guided velocity is

$$\hat{\mathbf{u}}_{\text{cfg}} = \hat{\mathbf{u}}_{\text{uncond}} + w\,(\hat{\mathbf{u}}_{\text{cond}} - \hat{\mathbf{u}}_{\text{uncond}})\,. \tag{18}$$

Setting $w = 0$ uses the student's internalized motion priors without simulation conditioning, while $w = 1$ recovers the conditional prediction; intermediate weights balance the two. The entire process requires only the final student model. Although simulation guidance depends on the quality of the student's own rollouts, w controls reliance on them at inference. Fig. 3 (right) shows consistent improvements over the baseline across the displayed guidance weights.

4 Experiments

4.1 Experimental Setup

We conduct our main evaluation and ablations on Bridge (Walke et al., 2023), assess generalization on InternData-A1 (Tian et al., 2026), and evaluate downstream VLA learning on LIBERO (Liu et al., 2023). For video prediction, models generate future frames from an initial observation and an action trajectory. The baseline combines Wan2.2-TI2V-5B (Wan et al., 2025) with a 1B Action DiT and is trained on real videos. We compare with Cosmos-Predict2.5 (Ali et al., 2025), GeniWorld (Gu et al., 2026), IRASim (Zhu et al., 2025), Ctrl-World (Guo et al., 2026b), and EnerVerse-AC (Jiang et al., 2025). SimForcing learns motion priors from synthetic rollouts and trains a student on paired simulated and real videos under a frozen simulation teacher. We report PSNR and SSIM for reconstruction fidelity, LPIPS for perceptual similarity, and FID and FVD for image- and video-level distributional quality. Higher PSNR and SSIM and lower LPIPS, FID, and FVD are better. For downstream policy learning, we report task success rates on LIBERO. More details can be found in the Appendix A.

4.2 Video Prediction Evaluation

Tab. 1 compares action-conditioned video prediction on Bridge. Among models without embodied pretraining, SimForcing achieves the best PSNR, SSIM, LPIPS, and FVD. Compared with

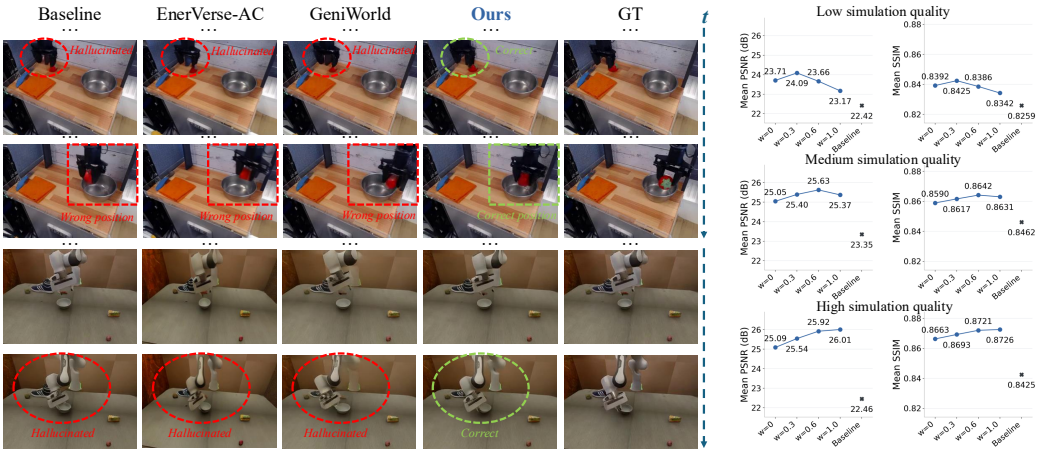


Figure 3: **Left:** Qualitative comparison across methods. The first two rows show examples from Bridge (Walke et al., 2023), and the last two from InternData-A1 (Tian et al., 2026). Our method produces fewer visual hallucinations and more accurate action-conditioned motion. **Right:** Effect of simulation prediction quality on our method's real-video predictions on Bridge.

Table 2: Component ablations on Bridge. FM, TA, MD, and SC denote joint flow matching, trajectory augmentation, motion distillation, and simulation conditioning, respectively. Best are in bold.

	Training components				Evaluation metrics				
ID	FM	TA	MD	SC	PSNR ↑	SSIM ↑	FID ↓	FVD ↓	LPIPS ↓
Wan2.2 (SFT) (Wan et al., 2025)					22.751	0.838	23.97	168.71	0.0926
(a)	✓	✗	✗	✗	24.128	0.851	22.03	141.81	0.0751
(b)	✓	✓	✗	✗	24.523	0.854	21.52	134.56	0.0702
(c)	✓	✗	✓	✗	24.166	0.849	27.46	154.69	0.0768
(d)	✓	✓	✓	✗	24.310	0.852	**19.18**	**100.27**	0.0769
(e)	✓	✓	✗	✓	23.732	0.848	21.83	145.40	0.0772
(f)	✓	✓	✓	✓	**25.077**	**0.858**	25.21	137.49	**0.0673**

GeniWorld, it provides closer agreement with action-conditioned reference videos at the same reported inference time, with a slightly higher FID reflecting a trade-off between prediction fidelity and image-level distributional quality. Both methods perform segmentation at inference, requiring only 0.3 s per sample. Cosmos-Predict2.5 achieves higher PSNR and SSIM and lower FVD, while SimForcing attains lower LPIPS with approximately five times faster inference. These results support the benefits of simulation-derived priors for prediction while maintaining practical inference costs. The qualitative comparisons in Fig. 3 (left) further illustrate these benefits on both Bridge and InternData-A1. In the shown examples, compared to other methods, SimForcing produces fewer visual hallucinations and motion that more closely follows the specified actions.

4.3 Analysis

Component ablations. Tab. 2 shows that joint flow matching and trajectory augmentation improve all five metrics over real-video SFT. With augmentation, motion distillation achieves the lowest FID and FVD, while combining it with simulation conditioning yields the best PSNR, SSIM, and LPIPS. This reflects a trade-off: the full model prioritizes prediction fidelity and perceptual similarity to reference videos over distributional quality. Tab. 3 further shows that latent-motion distillation outperforms latent-value matching on four metrics, supporting temporal differences as a transfer target that mitigates the influence of cross-domain appearance differences. CFG provides additional control over this trade-off: $w = 0.6$ achieves the best PSNR, SSIM, and LPIPS, whereas $w = 1$ yields the lowest FID and FVD. Fig. 3 (right) further illustrates adjustable simulation guidance, with improvements over the baseline across the displayed weights.

Table 3: Ablation studies on distillation targets and CFG weights on the Bridge validation set. Bold indicates the best result within each comparison.

(a) Distillation target

Target	PSNR ↑	SSIM ↑	FID ↓	FVD ↓	LPIPS ↓
Latent value	19.232	0.803	**26.38**	193.15	0.0972
Latent motion	**24.166**	**0.849**	27.46	**154.69**	**0.0768**

(b) CFG weight

w	PSNR ↑	SSIM ↑	FID ↓	FVD ↓	LPIPS ↓
0.0	24.620	0.8549	25.08	145.96	0.0716
0.3	25.014	0.8578	26.68	144.19	0.0687
0.6	**25.077**	**0.8583**	25.21	137.49	**0.0673**
1.0	24.857	0.8567	**24.70**	**133.66**	0.0681

Table 4: Video prediction and component ablations on InternData-A1. Best results are in bold. FM, MD, SC denote joint flow matching, motion distillation, and simulation conditioning, respectively.

Model	FM	MD	SC	PSNR ↑	SSIM ↑	FID ↓	FVD ↓	LPIPS ↓
EnerVerse-AC (Jiang et al., 2025)	–	–	–	23.780	0.782	48.29	378.88	0.1056
Ctrl-World (Guo et al., 2026b)	–	–	–	24.063	0.786	44.42	363.15	0.1099
Wan2.2 (SFT) (Wan et al., 2025)	–	–	–	24.738	0.808	30.94	335.20	0.0827
GeniWorld (Gu et al., 2026)	–	–	–	24.234	0.804	41.54	355.52	0.0962
SimForcing component ablations								
Joint FM only	✓	✗	✗	25.382	**0.813**	**30.30**	317.05	**0.0742**
With motion distillation	✓	✓	✗	25.428	**0.813**	31.99	327.89	0.0761
With simulation conditioning	✓	✗	✓	25.029	0.808	32.06	318.78	0.0806
SimForcing (full)	✓	✓	✓	**25.562**	**0.813**	32.51	**314.21**	0.0765

Evaluation on InternData-A1. Tab. 4 evaluates our framework on InternData-A1, with models trained on its training split. The full model outperforms Ctrl-World, GeniWorld, and EnerVerse-AC across all five metrics and improves over Wan2.2 SFT on all metrics except FID. It achieves the highest PSNR and lowest FVD, and ties for the highest SSIM, while joint flow matching alone yields the lowest FID and LPIPS. These results support the framework's applicability to another robot dataset, with different configurations offering distinct benefits across metrics.

Downstream policy learning. Action-conditioned video prediction and downstream VLA learning share the same model architecture, with all pretrained parameters transferred to initialize the downstream model. The action expert changes from a conditioning branch to an action-prediction branch. Tab. 5 shows that this initialization improves average LIBERO success over Wan2.2 across all three policy-training settings. The largest gain occurs with action-only training, increasing average success from 82.6% to 89.2%, particularly through improvements on Goal and Long. Smaller gains persist with unconditional and joint video–action training, with consistent improvements on Long across settings. These results provide preliminary evidence that the pretrained world-model representations benefit downstream policy learning.

Table 5: LIBERO success rates (%). Arrows indicate improvements over Wan2.2 within each training setting.

Init.	Spatial	Object	Goal	Long	Avg.
FastWAM	97.8	99.2	97.8	95.6	97.6
Action-only training					
Wan2.2	88.0	95.6	82.8	64.0	82.6
Our WM	89.4	95.6	92.4	79.2	89.2↑
Unconditional training					
Wan2.2	96.4	99.6	97.6	92.2	96.5
Our WM	97.2	99.8	98.0	94.8	97.5↑
Video + action training					
Wan2.2	99.4	98.2	98.8	95.8	98.1
Our WM	98.6	99.8	97.8	97.4	98.4↑

5 Conclusion

We present SimForcing, a simulation-guided framework for action-conditioned robot video prediction. A simulation world model serves as a training-time teacher, transferring motion knowledge to a real-domain student through latent-motion distillation. Simulation-conditioning classifier-free guidance complements these internalized priors with controllable guidance from predicted simulation latents, requiring only the final student model at inference. SimForcing achieves the best prediction fidelity and perceptual similarity among the compared methods without embodied pretraining, while downstream experiments provide preliminary evidence of its utility for policy learning.

AI USE STATEMENT

Generative AI tools were used solely for grammar correction and language polishing and were not used for any other tasks. All AI-assisted revisions were reviewed and approved by two authors to ensure accuracy, appropriate wording, and consistency with the intended meaning. The authors take full responsibility for the final content of this work.

REFERENCES

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

Hassan Abu Alhaija, Jose Alvarez, Maciej Bala, Tiffany Cai, Tianshi Cao, Liz Cha, Joshua Chen, Mike Chen, Francesco Ferroni, Sanja Fidler, et al. Cosmos-transfer1: Conditional world generation with adaptive multimodal control. *arXiv preprint arXiv:2503.14492*, 2025.

Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, et al. World simulation with video foundation models for physical ai. *arXiv preprint arXiv:2511.00062*, 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, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

Yixiang Chen, Peiyan Li, Yuan Xu, Qisen Ma, Jiabing Yang, Kai Wang, Jianhua Yang, Dong An, He Guan, Gaoteng Liu, et al. Flowwam: Optical flow as a unified action representation for world action models. *arXiv preprint arXiv:2607.13017*, 2026a.

Yuzhi Chen, Ronghan Chen, Dongjie Huo, Yandan Yang, Dekang Qi, Haoyun Liu, Tong Lin, Shuang Zeng, Junjin Xiao, Xinyuan Chang, et al. Abot-physworld: Interactive world foundation model for robotic manipulation with physics alignment. *arXiv preprint arXiv:2603.23376*, 2026b.

Xiaowei Chi, Peidong Jia, Chun-Kai Fan, Xiaozhu Ju, Weishi Mi, Kevin Zhang, Zhiyuan Qin, Wanxin Tian, Kuangzhi Ge, Hao Li, et al. Wow: Towards a world omniscient world model through embodied interaction. *arXiv preprint arXiv:2509.22642*, 2025.

Xiao Fu, Xintao Wang, Xian Liu, Jianhong Bai, Runsen Xu, Pengfei Wan, Di Zhang, and Dahua Lin. Learning video generation for robotic manipulation with collaborative trajectory control. In *International Conference on Learning Representations*, volume 2026, pp. 144128–144142, 2026.

Chenghao Gu, Hanyang Yu, Jingbo Zhang, Haitao Lin, Wenyao Zhang, Jinghe Wang, Hanglei Jin, Shuzhao Xie, Jingyan Jiang, and Zhi Wang. Geniworld: A generalizable interactive world model for robotic manipulation via visual actions. *arXiv preprint arXiv:2608.06332*, 2026.

Yanjiang Guo, Tony Lee, Lucy Xiaoyang Shi, Jianyu Chen, Percy Liang, and Chelsea Finn. Vlaw: Iterative co-improvement of vision-language-action policy and world model. *arXiv preprint arXiv:2602.12063*, 2026a.

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, 2026b.

Haoran He, Yang Zhang, Liang Lin, Zhongwen Xu, and Ling Pan. Pre-trained video generative models as world simulators. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 4645–4653, 2026.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, et al. Dreamgen: Unlocking generalization in robot learning through video world models. *arXiv preprint arXiv:2505.12705*, 2025.

Yuxin Jiang, Shengcong Chen, Siyuan Huang, Liliang Chen, Pengfei Zhou, Yue Liao, Xindong He, Chiming Liu, Hongsheng Li, Maoqing Yao, et al. Enerverse-ac: Envisioning embodied environments with action condition. *arXiv preprint arXiv:2505.09723*, 2025.

JB Lanier, Kyungmin Kim, Armin Karamzade, Yifei Liu, Ankita Sinha, Kat He, Davide Corsi, and Roy Fox. Adapting world models with latent-state dynamics residuals. *arXiv preprint arXiv:2504.02252*, 2025.

Jacob Levy, Tyler Westenbroek, Kevin Huang, Fernando Palafox, Patrick Yin, Shayegan Omidshafiei, Dong-Ki Kim, Abhishek Gupta, and David Fridovich-Keil. Simulation distillation: Pretraining world models in simulation for rapid real-world adaptation. *arXiv preprint arXiv:2603.15759*, 2026.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Shuailei Ma, Jiaqi Liao, Xinyang Wang, Jingjing Wang, Chaoran Feng, Zijing Hu, Chong Bao, Zichen Xi, Yuqi Gan, Weisen Wang, et al. Scaling mixture-of-experts video pretraining for embodied intelligence. *arXiv preprint arXiv:2607.07675*, 2026.

Abhiram Maddukuri, Zhenyu Jiang, Lawrence Yunliang Chen, Soroush Nasiriany, Yuqi Xie, Yu Fang, Wenqi Huang, Zu Wang, Zhenjia Xu, Nikita Chernyadev, et al. Sim-and-real co-training: A simple recipe for vision-based robotic manipulation. *arXiv preprint arXiv:2503.24361*, 2025.

Marc Rigter, Tarun Gupta, Agrin Hilmkil, and Chao Ma. Avid: Adapting video diffusion models to world models. *arXiv preprint arXiv:2410.12822*, 2024.

Yu Shang, Zhuohang Li, Yiding Ma, Weikang Su, Xin Jin, Ziyou Wang, Lei Jin, Xin Zhang, Yinzhou Tang, Haisheng Su, et al. Worldarena: A unified benchmark for evaluating perception and functional utility of embodied world models. *AI Open*, 7:208–226, 2026.

Yang Tian, Yuyin Yang, Yiman Xie, Zetao Cai, Xu Shi, Ning Gao, Hangxu Liu, Xuekun Jiang, Zherui Qiu, Feng Yuan, et al. Interndata-a1: Pioneering high-fidelity synthetic data for pretraining generalist policy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 976–985, 2026.

Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on robot learning*, pp. 1723–1736. PMLR, 2023.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Zhuoyuan Wu and Jun Gao. Oscar: Omni-embodiment action-conditioned world model for robotics. *arXiv preprint arXiv:2606.04463*, 2026.

Junjie Ye, Rong Xue, Basile Van Hoorick, Runhao Li, Harshitha Rajaprakash, Pavel Tokmakov, Muhammad Zubair Irshad, Vitor Guizilini, and Yue Wang. Robodream: Compositional world models for scalable robot data synthesis. *arXiv preprint arXiv:2606.02577*, 2026.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026.

Jie Zhang, Xiaoyue Chen, Anzhe Chen, Dayiheng Liu, Deqing Li, Gengze Zhou, Hale Yin, Haoqi Yuan, Haoyang Li, Jiahao Li, et al. Qwen-robotworld technical report: Unifying embodied world modeling through language-conditioned video generation. *arXiv preprint arXiv:2606.17030*, 2026.

Haoyu Zhao, Xingyue Zhao, Hangyu Li, Biao Gong, Kehan Li, Siteng Huang, Xin Li, Deli Zhao, and Zhongyu Li. Rynnworld-teleop: An action-conditioned world model for digital teleoperation. *arXiv preprint arXiv:2607.06558*, 2026.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. Irasim: A fine-grained world model for robot manipulation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9834–9844. IEEE, 2025.



Figure A1: Qualitative comparison on Bridge: example 1. From top to bottom: Baseline (Wan et al., 2025), EnerVerse-AC (Jiang et al., 2025), GeniWorld (Gu et al., 2026), our simulator-domain predictions, our real-world predictions, and ground truth. The seven columns show aligned temporal snapshots.

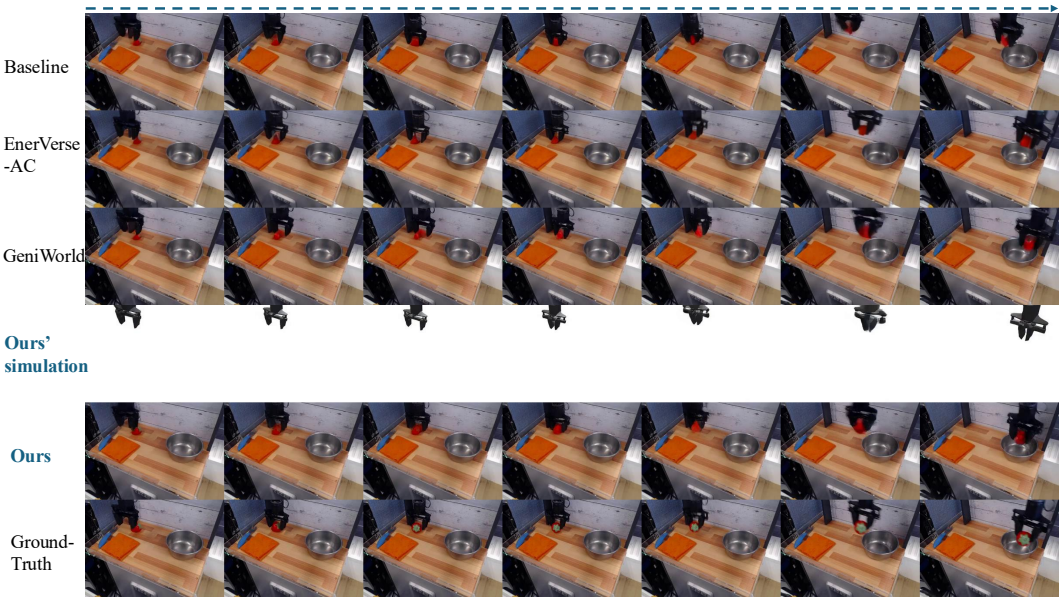


Figure A2: Qualitative comparison on Bridge: example 2. Row order and temporal sampling are identical to Figure A1.

A Additional Qualitative Results

We present additional qualitative comparisons on Bridge (Walke et al., 2023) and InternData-A1 (Tian et al., 2026), with four examples from each dataset.

Each example contains seven uniformly spaced frames from the temporal interval shared by all methods. The comparisons allow inspection of object appearance, robot motion, and agreement



Figure A3: Qualitative comparison on Bridge: example 3. Row order and temporal sampling are identical to Figure A1.

with the ground-truth sequence. We additionally show the simulator-domain predictions produced by our method alongside its real-world predictions.

Across all figures, rows from top to bottom show Baseline (Wan et al., 2025), EnerVerse-AC (Jiang et al., 2025), GeniWorld (Jiang et al., 2025), our simulator-domain predictions, our real-world predictions, and ground truth. Our simulator-domain and real-world predictions use the same temporal indices.

A.1 Qualitative Results on Bridge

Figures A1–A3 present four qualitative examples on Bridge (Walke et al., 2023). Columns correspond to temporally aligned frames.

A.2 Qualitative Results on InternData-A1

Figures A4–A6 present four qualitative examples on InternData-A1 (Tian et al., 2026). Columns correspond to temporally aligned frames.

B Latent Value vs. Latent Motion Distillation.

We compare two distillation objectives: latent value distillation, which matches the student's predicted future latents to those of the frozen teacher, and latent motion distillation, which instead matches their differences between consecutive latent frames. Figure A7 presents the corresponding distillation losses for the real and synthetic branches. On the real branch, the latent value loss remains approximately constant, whereas the latent motion loss decreases during training. On the synthetic branch, both objectives decrease in the later stages, with latent motion exhibiting an initial rise followed by a sustained decline. Since the two objectives supervise different quantities, their absolute loss magnitudes should not be interpreted as a direct comparison of model quality.

C Effect of Training-Time Conditioning Probability

Experimental setup. We evaluate multi-block dropout conditioning on the Bridge validation set by varying the training-time condition retention probability $p \in \{0, 0.3, 0.7, 1.0\}$ and inference-time

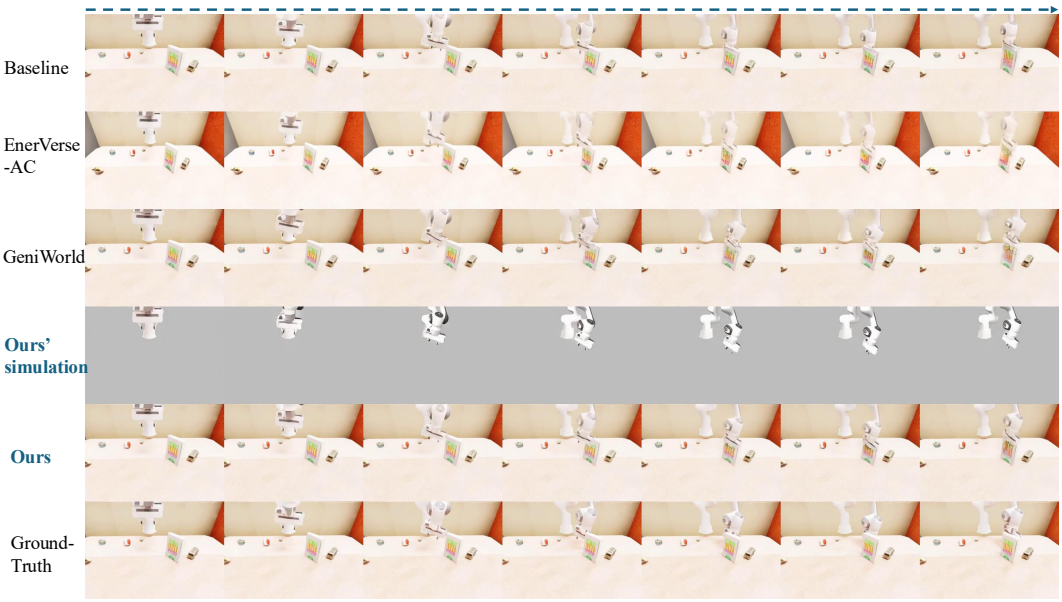


Figure A4: Qualitative comparison on InternData-A1: example 1. From top to bottom: Baseline (Wan et al., 2025), EnerVerse-AC (Jiang et al., 2025), GeniWorld (Gu et al., 2026), our simulator-domain predictions, our real-world predictions, and ground truth. The seven columns show aligned temporal snapshots.

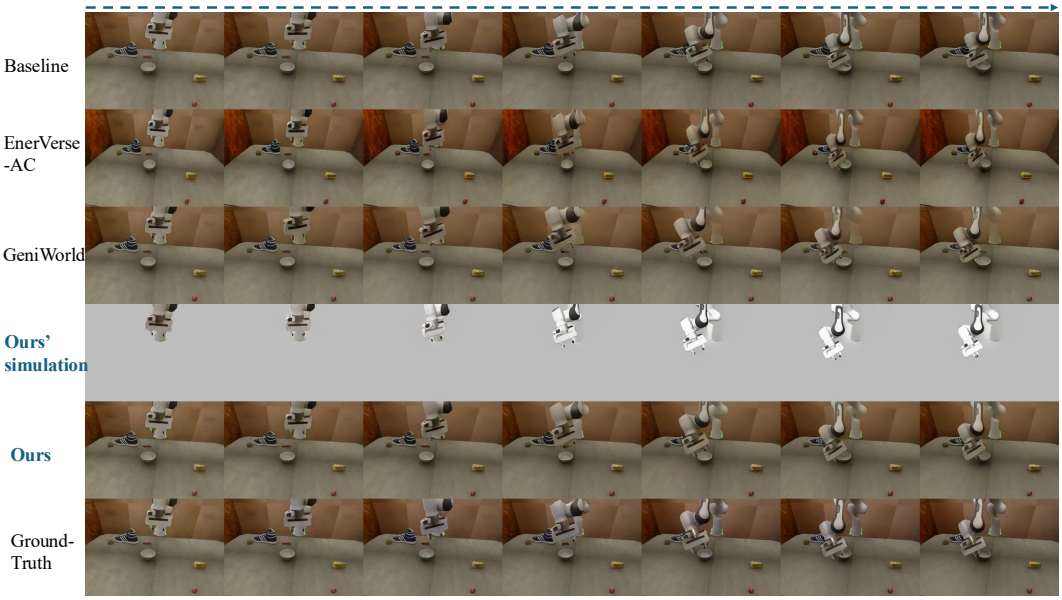


Figure A5: Qualitative comparison on InternData-A1: example 2. Row order and temporal sampling are identical to Figure A4.

CFG weight $w \in \{0, 0.3, 0.6, 1.0\}$. Here, p is the probability of enabling simulation conditioning for each training sample, with the same gate shared across all video transformer blocks. Each checkpoint generates its own simulation conditions and real-video predictions, so the comparison evaluates the complete pipeline. We report PSNR and SSIM on the same evaluation samples. The $p = 0$ checkpoint uses fewer training steps than the others.

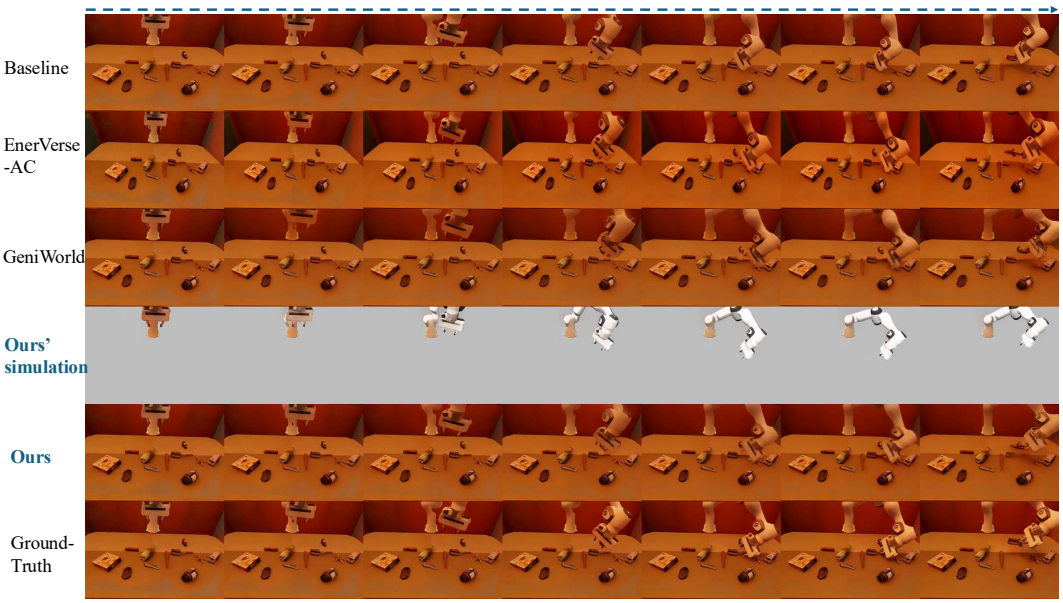


Figure A6: Qualitative comparison on InternData-A1: example 3. Row order and temporal sampling are identical to Figure A4.

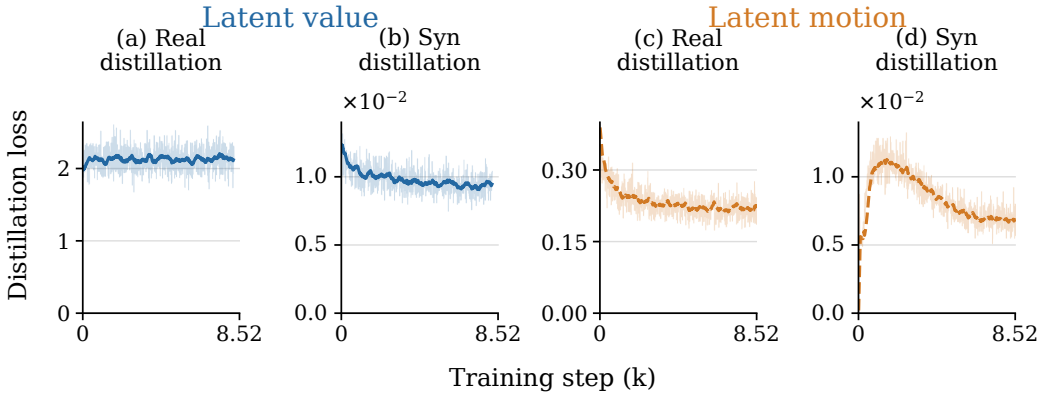


Figure A7: Distillation loss curves for latent value (a–b) and latent motion (c–d) on the real and synthetic branches. Faint curves show logged losses, and bold curves show trailing averages over 250 training steps. The real-branch panels use different vertical scales; the synthetic-branch panels share the same scale.

Effect of multi-block dropout conditioning. Table A1 shows that intermediate condition retention probabilities achieve higher best-observed PSNR and SSIM than $p = 0$ or $p = 1$ in this sweep. For $p = 0$, both metrics remain unchanged across guidance weights. For $p = 1$, removing simulation conditioning at inference substantially reduces prediction quality, suggesting strong dependence on the condition. In contrast, $p = 0.3$ and $p = 0.7$ maintain better performance without conditioning while also benefiting from simulation guidance. These results support the dropout mechanism in multi-block dropout conditioning, which trains the student to predict with and without simulation inputs.

Interaction with CFG. The configuration $(p, w) = (0.7, 0.6)$ achieves the highest PSNR and SSIM among the evaluated settings, supporting our default choice. For $p = 0.7$, $w = 0.6$ outperforms both $w = 0$ and $w = 1$, indicating that moderate simulation guidance improves prediction over either branch alone. For $p = 0.3$, PSNR peaks at $w = 1$ and SSIM at $w = 0.6$, although the

Table A1: Joint evaluation of training-time conditioning probability p and inference-time guidance weight w. Higher PSNR (dB) and SSIM are better. Bold indicates the best result in the entire table for each metric.

	$w = 0$		$w = 0.3$		$w = 0.6$		$w = 1.0$	
p	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
0.0	24.441	0.8524	24.441	0.8524	24.441	0.8524	24.441	0.8524
0.3	24.675	0.8547	24.825	0.8561	24.890	0.8567	24.896	0.8566
0.7	24.620	0.8549	25.014	0.8578	**25.077**	**0.8583**	24.857	0.8567
1.0	21.832	0.8408	23.570	0.8502	24.387	0.8536	24.009	0.8508

differences between these settings are small. These results show that the benefit of CFG depends on the condition retention probability used during training.

D Inference Steps and Simulation Guidance

Evaluation protocol. We investigate the quality–latency trade-off by varying the simulation generation steps N_s, real-video generation steps N_r, and simulation-conditioning CFG weight w. We evaluate all combinations of $N_s, N_r \in \{1, 5, 10, 20\}$ and $w \in \{0, 0.3, 0.6, 1\}$, yielding 64 configurations on the same 100 Bridge validation samples and initial observations. Both generation stages use the same student checkpoint, with masked initial observations as simulation inputs. PSNR and SSIM are averaged over all 100 samples. Latency is averaged over the remaining 99 samples after excluding the first warm-up sample for each configuration. It includes simulation generation, real-video generation, and real-video decoding, but excludes model loading, metric computation, and file saving. Although simulation predictions are shared across the four CFG weights during evaluation, each reported latency includes the full simulation generation cost for one configuration.

Effect of inference steps. Tables A2 and A3 show that increasing real-video generation from one to five or ten steps substantially improves prediction fidelity, while increasing from ten to twenty steps provides only small additional gains. Increasing simulation steps has a much smaller effect and does not consistently improve either metric. The highest observed PSNR and SSIM occur at $(N_s, N_r, w) = (5, 20, 0.6)$, reaching 25.090 dB and 0.8585, respectively. At $w = 0.6$, even one simulation step produces results close to those obtained with larger simulation budgets, suggesting that a coarse predicted simulation reference can already provide useful guidance.

Guidance and latency. The weight $w = 0.6$ yields the highest PSNR and SSIM for every tested step combination. At $w = 0$, both metrics are invariant to N_s, consistent with the absence of simulation conditioning in the real-video prediction branch. Table A4 shows that $(1, 10, 0.6)$ achieves 25.063 dB PSNR and 0.8575 SSIM in 2.008 s per sample: its PSNR is 0.027 dB below the highest observed value, with approximately 47% lower latency than $(5, 20, 0.6)$. For a stronger emphasis on speed, $(1, 10, 1)$ reduces latency to 1.245 s, with 24.838 dB PSNR and 0.8559 SSIM. These results indicate that inference cost can be reduced by allocating fewer steps to simulation generation and avoiding excessive real-video sampling steps.

Timing interpretation and limitations. In the evaluated implementation, $w = 1$ uses a single forward pass per real-video sampling step, whereas the other weights use the two-branch CFG path. The recorded $w = 0$ latency also includes simulation generation; it therefore does not represent an optimized configuration that omits unused simulation predictions and the unused conditional branch. The $N_s = 20$ configurations were evaluated in a separate run, so their timings may be affected by differences in machine load. Results are from a single sweep without repeated-run uncertainty estimates; small metric differences should not be interpreted as statistically established improvements.

Table A2: PSNR (dB) across inference-step budgets and CFG weights on the 100-sample Bridge evaluation. N_s and N_r denote simulation and real-video generation steps. Bold indicates the best result within each row.

Inference steps		CFG weight w			
N_s	N_r	0	0.3	0.6	1
1	1	22.785	23.116	**23.301**	23.206
1	5	24.263	24.697	**24.851**	24.629
1	10	24.576	24.958	**25.063**	24.838
1	20	24.620	24.983	**25.063**	24.843
5	1	22.785	23.130	**23.314**	23.177
5	5	24.263	24.732	**24.847**	24.587
5	10	24.576	24.995	**25.083**	24.847
5	20	24.620	25.019	**25.090**	24.872
10	1	22.785	23.127	**23.307**	23.167
10	5	24.263	24.727	**24.836**	24.574
10	10	24.576	24.990	**25.070**	24.832
10	20	24.620	25.014	**25.077**	24.857
20	1	22.785	23.125	**23.304**	23.163
20	5	24.263	24.725	**24.829**	24.566
20	10	24.576	24.989	**25.062**	24.823
20	20	24.620	25.012	**25.070**	24.847

Table A3: SSIM across inference-step budgets and CFG weights on the 100-sample Bridge evaluation. N_s and N_r denote simulation and real-video generation steps. Bold indicates the best result within each row.

Inference steps		CFG weight w			
N_s	N_r	0	0.3	0.6	1
1	1	0.8231	0.8264	**0.8280**	0.8271
1	5	0.8483	0.8518	**0.8528**	0.8510
1	10	0.8539	0.8569	**0.8575**	0.8559
1	20	0.8549	0.8576	**0.8582**	0.8567
5	1	0.8231	0.8266	**0.8281**	0.8268
5	5	0.8483	0.8520	**0.8527**	0.8504
5	10	0.8539	0.8571	**0.8578**	0.8559
5	20	0.8549	0.8579	**0.8585**	0.8569
10	1	0.8231	0.8266	**0.8281**	0.8267
10	5	0.8483	0.8520	**0.8526**	0.8502
10	10	0.8539	0.8571	**0.8576**	0.8557
10	20	0.8549	0.8578	**0.8583**	0.8567
20	1	0.8231	0.8266	**0.8281**	0.8266
20	5	0.8483	0.8520	**0.8526**	0.8502
20	10	0.8539	0.8571	**0.8576**	0.8556
20	20	0.8549	0.8578	**0.8583**	0.8566

Table A4: Mean inference latency (s/sample) across inference-step budgets and CFG weights.

Inference steps		CFG weight w			
N_s	N_r	0	0.3	0.6	1
1	1	0.644	0.649	0.649	0.571
1	5	1.251	1.258	1.259	0.879
1	10	2.002	2.008	2.008	1.245
1	20	3.519	3.546	3.539	2.012
5	1	0.934	0.941	0.940	0.862
5	5	1.543	1.557	1.553	1.168
5	10	2.298	2.304	2.304	1.536
5	20	3.799	3.802	3.805	2.286
10	1	1.288	1.293	1.292	1.215
10	5	1.890	1.895	1.895	1.512
10	10	2.658	2.664	2.661	1.899
10	20	4.175	4.175	4.179	2.652
20	1	1.946	1.951	1.951	1.876
20	5	2.551	2.557	2.558	2.183
20	10	3.284	3.295	3.295	2.547
20	20	4.772	4.775	4.777	3.287

E Implementation Details

BridgeData V2. We use a paired simulation–real subset of BridgeData V2 (Walke et al., 2023) containing 1,952 robot manipulation trajectories. Each pair consists of temporally aligned simulated and real-world videos sharing the same action sequence, proprioceptive states, and language instruction. We retain trajectories containing at least 21 frames and randomly sample 30 temporal windows per eligible trajectory in each epoch. Each window contains 21 consecutive video frames and 20 aligned actions, with one action per video transition. Both actions and proprioceptive states are seven-dimensional. For evaluation, we select 100 paired samples from the validation set.

InternData-A1. We additionally use the Franka manipulation subset of InternData-A1 (Tian et al., 2026). Each training example pairs a simulation rendering with its corresponding real-camera video, sharing the action sequence, proprioceptive states, and language instruction. The training split contains 2,657 trajectories. We retain paired sequences that support a complete 129-step window and randomly sample 30 windows per eligible trajectory per epoch. Video frames are sampled every four time steps, yielding 33 frames per window, while all 128 actions are retained. Each video transition therefore corresponds to four consecutive actions. The seven-dimensional actions comprise a six-dimensional end-effector pose and a gripper-opening command. The seven-dimensional proprioceptive states comprise the end-effector pose and gripper position. For evaluation, we select 80 paired samples from the validation set.

Data preprocessing. For both datasets, simulated and real-world videos undergo the same aspect-ratio-preserving resizing and center cropping to 224×320 pixels. Pixel values are normalized to $[-1, 1]$. Actions and proprioceptive states are normalized using dataset-level minimum and maximum statistics. The paired videos use identical temporal windows to preserve alignment between visual observations and actions. Language instructions are represented by precomputed text embeddings with a maximum sequence length of 128 tokens.

Model initialization and conditioning. Our video backbone is based on Wan2.2-TI2V-5B (Wan et al., 2025). For each dataset, the student is initialized from an action-conditioned simulation world model, and a frozen copy of the same model serves as the motion teacher. The video backbone contains 30 transformer blocks with a hidden dimension of 3,072 and 24 attention heads. The action transformer contains 30 blocks with a hidden dimension of 1,024.

Motion supervision matches adjacent-frame differences in the estimated clean video latents and is applied to every training sample. The real-world prediction branch additionally receives paired

Table A5: Dataset-specific training settings. Batch sizes count paired simulation–real samples per device.

Setting	BridgeData V2	InternData-A1
Video frames per window	21	33
Actions per window	20	128
Actions per video transition	1	4
Batch size per device	16	10
$\lambda^{r}_{\mathrm{FM}}$	1.0	1.0
$\lambda^{s}_{\mathrm{FM}}$	0.5	0.5
$\lambda^{s2r}_{\mathrm{motion}}$	1.0	1.0
$\lambda^{s2s}_{\mathrm{motion}}$	0.1	1.0

simulation latents with probability 0.7. The residual projections used to inject this condition are initialized to zero. The simulation prediction branch does not receive this auxiliary condition. Initial-frame and action conditions are retained throughout training, and the strength of motion supervision is unchanged when the auxiliary simulation condition is present.

Training objective. We jointly optimize flow matching in both domains and teacher-guided motion supervision:

$$\begin{aligned}\mathcal{L} = \lambda^{r}_{\mathrm{FM}}\mathcal{L}^{r}_{\mathrm{FM}} + \lambda^{s}_{\mathrm{FM}}\mathcal{L}^{s}_{\mathrm{FM}} \\ + \lambda^{s2r}_{\mathrm{motion}}\mathcal{L}^{s2r}_{\mathrm{motion}} + \lambda^{s2s}_{\mathrm{motion}}\mathcal{L}^{s2s}_{\mathrm{motion}}.\end{aligned} \tag{19}$$

The loss weights for the two datasets are summarized in Table A5. No action-prediction loss is used.

Optimization. For both datasets, we train for 20 epochs using AdamW with a learning rate of 10^{-4}, momentum coefficients $(\beta_1, \beta_2) = (0.9, 0.95)$, and weight decay 10^{-2}. The learning rate is linearly warmed up over the first 5% of training steps and subsequently decayed with a cosine schedule. We perform one optimizer update per batch without gradient accumulation. Training uses bfloat16 mixed precision, gradient checkpointing, and gradient clipping with a maximum norm of 1.0. The video autoencoder and motion teacher remain frozen throughout training. We use 1,000 diffusion training timesteps with a timestep-shift factor of 5.0 and set the random seed to 42.

Baseline implementations. For IRASim (Zhu et al., 2025) and Cosmos-Predict2.5 (Ali et al., 2025), we directly evaluate the released checkpoints trained on Bridge, without additional fine-tuning. For Ctrl-World (Guo et al., 2026b) and EnerVerse-AC (Jiang et al., 2025), we use their publicly available code to train on our training splits for a number of optimization steps comparable to SimForcing. As no public implementation of GeniWorld (Gu et al., 2026) was available when we conducted the experiments, we implement its approach based on the paper using Wan2.2-TI2V-5B (Wan et al., 2025). Our GeniWorld implementation shares the same backbone architecture as SimForcing to facilitate a controlled comparison.

F Downstream Policy Learning on LIBERO

Experimental setup. We evaluate the transferability of our pretrained world model to robotic manipulation on the four LIBERO suites: Spatial, Object, Goal, and Long. As summarized in Table 5, we compare Wan2.2-based initialization with our world-model initialization under three downstream training settings. The policy uses a video expert and an action expert coupled through attention. Its observations comprise an external-camera image and a wrist-camera image, concatenated spatially, together with the task instruction and the current proprioceptive state. The action expert predicts a sequence of end-effector motion and gripper commands. For the Wan2.2 baseline, the video expert is initialized from Wan2.2 and the action expert from the corresponding Wan2.2-derived ActionDiT initialization. For Our WM, both experts inherit the compatible weights of our pretrained world-model checkpoint. Thus, this comparison evaluates the transfer of the pretrained two-expert model, rather than a replacement of the visual backbone alone.

The following three training settings follow FastWAM (Yuan et al., 2026); please refer to the original paper for further details.

Action-only training. In this setting, the video expert is frozen and serves as a visual feature extractor for the current observation. Its layer-wise keys and values provide visual context for the action expert, which learns to denoise action trajectories conditioned on this context, the task instruction, and proprioception. Only the action expert and the proprioceptive projection are optimized. Future video frames are not used as prediction targets, and no video-generation loss is applied. This setting tests whether the pretrained model provides a useful initialization for policy learning when downstream adaptation is restricted to the action pathway.

Unconditional training. Here, both experts are optimized using video and action flow-matching objectives. Future video latents and action trajectories are independently corrupted with noise, while the initial observation remains clean and provides visual conditioning. The video expert predicts future visual dynamics without receiving action tokens or ground-truth actions. The action expert attends to its own action tokens and the video tokens of the initial observation, but cannot attend to future video tokens. The initial observation tokens are also prevented from attending to future frames, avoiding indirect access to future observations through the visual context. Consequently, video prediction supplies an auxiliary learning objective for the visual representation, while action prediction depends only on information available at decision time. The term *unconditional* refers specifically to the absence of action conditioning in the video branch; visual, language, and proprioceptive conditioning are retained. At inference, actions can be generated from the current observation without generating a future video.

Video + action training. This setting retains the video and action flow-matching objectives and jointly optimizes both experts, but expands the action expert's attention to the complete video token sequence. Action prediction can therefore use both the current observation and the evolving representations of future video frames. The coupling is directional: action tokens attend to video tokens, whereas video tokens do not attend to action tokens. During training, the future video tokens are noisy versions of demonstration latents. During inference, future video latents and actions are generated together from noise, with the initial observation fixed. Thus, the additional visual context available to the policy comes from predicted futures rather than ground-truth future observations.

Results and interpretation. Our world-model initialization improves the reported average success rate in all three settings: from 82.6% to 89.2% under action-only training, from 96.5% to 97.5% under unconditional training, and from 98.1% to 98.4% under video + action training. These correspond to gains of 6.6, 1.0, and 0.3 percentage points, respectively. The largest improvement occurs under action-only adaptation, including gains of 9.6 points on Goal and 15.2 points on Long. This result supports the usefulness of the transferred initialization even when the visual expert is frozen. The gains are smaller when both experts receive downstream video and action supervision. Video + action training with Our WM achieves the highest reported average success rate, 98.4%, compared with 97.6% for the FastWAM reference. The improvements are not uniform across suites: in the video + action setting, gains on Object and Long offset decreases on Spatial and Goal. Overall, the results support improved downstream transfer across the three training settings, without implying an improvement on every individual suite.