SKELETON-GROUNDED LATENT ACTION MODEL FOR ROBOT WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Robot world models simulate how environments evolve in response to robot actions. Latent action models enable learning dynamics from videos by encoding visual transitions as proxy actions. However, reconstruction-driven latent actions mix robot motion with object responses and other scene changes, complicating their correspondence with robot controls. We propose **SkeLAM**, a skeleton-guided latent action model for controllable robot world modeling. SkeLAM decomposes latent actions into a kinematic component grounded by robot skeleton supervision and a residual component retaining complementary visual dynamics. Cross-modal swap reconstruction encourages RGB-derived and skeleton-derived kinematic representations to be interchangeable during video reconstruction. The world model incorporates both components through separate conditioning pathways, while branch dropout reduces its dependence on future-derived residual information. A lightweight adapter maps robot actions to the kinematic space, and subsequent world-model fine-tuning adapts generation to these predicted conditions. At inference, the model generates future observations from the current observation and robot actions. On DROID, SkeLAM reduces end-effector ADE by 19.5% and LPIPS by 11.1% compared with DreamDojo. Motion probing supports kinematic specialization, while qualitative results demonstrate zero-shot human-to-robot motion transfer in generated rollouts. Policy evaluation shows agreement between predicted success rates and those measured in real robot executions.

1 INTRODUCTION

Robot world models simulate how environments evolve in response to robot actions, supporting policy evaluation and model-based planning (Yang et al., 2024; Wu et al., 2024). Accurate simulation requires modeling robot motion and resulting scene changes. Latent action models encode visual transitions as proxy actions (Bruce et al., 2024; Ye et al., 2025), allowing world models to learn dynamics from large-scale videos without explicit action annotations (Gao et al., 2026). These latent actions are learned through future-observation reconstruction, with the current observation providing visual context (Gao et al., 2025; Liang et al., 2026). For action-conditioned simulation, inferred latent actions serve as supervision for mapping robot controls into the learned latent action space.

Learning this mapping is challenging since reconstruction objectives encourage latent actions to encode robot motion together with object responses and other scene changes (Nikulin et al., 2025; Zhang et al., 2025). For example, similar end-effector motions may produce different object responses under different contact states (Gu et al., 2026; Xu et al., 2026), introducing scene-dependent variation beyond robot motion into the adaptation target. These interaction outcomes are useful for world modeling, since accurate simulation requires predicting how robot motion affects the surrounding scene. This motivates a dedicated kinematic component for action adaptation, together with a residual component retaining complementary transition information for dynamics learning.

To this end, we propose SkeLAM, a skeleton-guided latent action model for controllable robot world modeling. Robot skeletons describe robot configuration and motion while abstracting scene appearance. Related robot world models use skeleton sequences or kinematic renderings as direct conditioning signals for generation (Wang et al., 2025b; Wu & Gao, 2026; Xu et al., 2026). SkeLAM uses paired RGB and skeleton transitions to learn kinematic and residual components from RGB observations. Skeleton supervision grounds the kinematic latent in robot motion, while the residual

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.

latent supplies complementary information for RGB reconstruction. Together, the two components jointly support reconstruction, providing a dedicated target for action adaptation while retaining information for dynamics learning. During representation learning, skeletons supervise the RGB latents, while the deployed interface uses RGB observations and robot actions.

We further introduce cross-modal swap reconstruction, which replaces the RGB kinematic latent with its paired skeleton counterpart while retaining the RGB residual. Reconstructing the same transition under this substitution encourages compatible motion conditioning in the RGB decoder, extending cross-modal correspondence beyond alignment in latent space. The world model learns from both components through separate conditioning pathways, while branch dropout prepares it for the zero-residual condition used at deployment. A lightweight adapter maps robot actions to the kinematic space, and subsequent world-model fine-tuning adapts generation to these predicted conditions. At inference, generation uses the current observation and robot actions.

To assess motion accuracy and visual fidelity, we evaluate SkeLAM on DROID (Khazatsky et al., 2024). Compared with DreamDojo, SkeLAM reduces end-effector ADE by 19.5% and LPIPS by 11.1%. Motion probing shows that the kinematic component approaches the performance of the concatenated representation with half the latent dimensions, while latent interventions and component ablations support its role in generation. Qualitative results further demonstrate zero-shot human-to-robot motion transfer in generated rollouts. Open-loop policy evaluation shows agreement between predicted success rates and those measured in real robot executions.

Our contributions are summarized as follows:

- We propose SkeLAM, a skeleton-guided latent action model that separates the conditioning roles of robot motion and complementary visual dynamics, providing a dedicated kinematic target for robot action adaptation.
- We introduce cross-modal swap reconstruction to encourage compatible RGB and skeleton motion conditioning during generation. Combined with branch dropout, this design supports robot action conditioning through the kinematic pathway learned from visual transitions.
- Experiments on DROID demonstrate gains in motion accuracy and visual fidelity, supported by representation analyses and component ablations. We further show zero-shot human-to-robot motion transfer and agreement between predicted and measured policy success rates.

2 Related Work

Action-conditioned robot world models. Robot world models support interactive simulation (Yang et al., 2024; Wu et al., 2024; Wang et al., 2025a) and policy evaluation (Zhu et al., 2025; Guo et al., 2026; Liao et al., 2025). Control representations include action embeddings, skeleton sequences (Wang et al., 2025b; Wu & Gao, 2026), and kinematic renderings (Xu et al., 2026). Concurrent studies further explore spatially grounded world modeling (Gu et al., 2026; Yang et al., 2026b). SkeLAM uses skeletons as training supervision for kinematic latents, which an adapter predicts from robot actions to condition generation.

Latent action learning from videos. Latent action models encode visual transitions for world modeling (Bruce et al., 2024; Gao et al., 2025; 2026; Garrido et al., 2026) and robot policy learning (Ye et al., 2025; Liang et al., 2026; Yang et al., 2026a). Analyses of reconstruction-based learning identify sensitivity to distracting dynamics and ambiguity in action correspondence (Nikulin et al., 2025; Zhang et al., 2025). Motion and geometry supervision further structure these representations through optical flow constraints (Bu et al., 2026) and shared RGB–depth representations (Govind et al., 2026), alongside concurrent advances in latent representations for policy learning (Liu et al., 2026a; Shen et al., 2026; Xie et al., 2026; Liu et al., 2026b). SkeLAM learns a dedicated kinematic component for action adaptation while retaining visual dynamics in a residual component.

3 Methodology

SkeLAM learns structured latent actions for video-based world modeling and subsequent adaptation to robot controls. As illustrated in Figure 1, skeleton supervision grounds a kinematic component in robot motion, while a residual component retains complementary visual dynamics. The world model

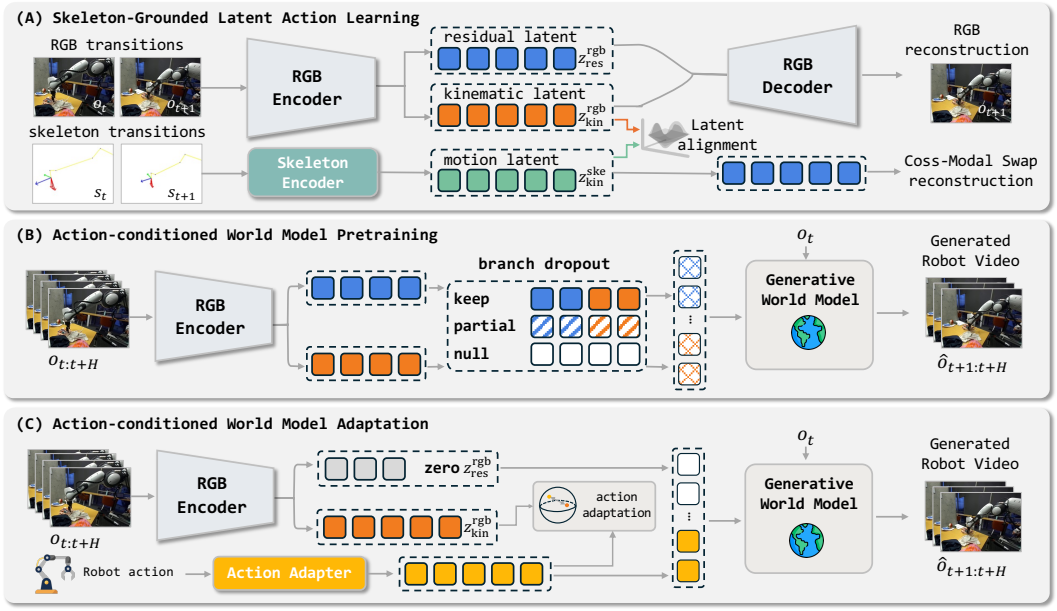


Figure 1: **Overview of SkeLAM.** (A) Skeleton supervision and cross-modal swap reconstruction structure RGB transition latents into kinematic and residual components. (B) Separate projections condition the world model: keep retains both branches, partial drops one, and null drops both. (C) An action adapter predicts kinematic latents, with the null residual at inference.

learns from both components, with branch dropout preparing it for action-conditioned inference. An action adapter is then trained to predict kinematic latents from robot actions, followed by world-model fine-tuning using these predictions.

Preliminary. Let o_t and a_t denote the visual observation and robot action at time t. An action-conditioned world model represents the distribution of H future observations:

$$p(o_{t+1:t+H} \mid o_t, a_{t:t+H-1}) \,. \tag{1}$$

To learn dynamics from videos, latent action models infer proxy actions z_t from visual transitions (o_t, o_{t+1}). These representations are learned by reconstructing o_{t+1} from o_t and z_t, and subsequently provide conditioning signals for world-model training.

3.1 Skeleton-Grounded Latent Actions

Latent decomposition. To learn a kinematic representation for robot action adaptation while retaining complementary information for RGB reconstruction, we encode each RGB transition into kinematic latents $z^{\text{rgb}}_{\text{kin},t}$ and residual latents $z^{\text{rgb}}_{\text{res},t}$. Further, to supervise the kinematic latent, we use skeleton images depicting the corresponding robot joint configurations.

Given paired RGB observations (o_t, o_{t+1}) and their corresponding skeleton images (s_t, s_{t+1}), Transformer encoders parameterize Gaussian posteriors for the RGB kinematic and residual latents, and the skeleton motion latent:

$$\begin{aligned} \left(z^{\text{rgb}}_{\text{kin},t}, z^{\text{rgb}}_{\text{res},t}\right) &\sim q^{\text{rgb}}_{\phi}(z_{\text{kin}}, z_{\text{res}} \mid o_t, o_{t+1}), \\ z^{\text{ske}}_{\text{kin},t} &\sim q^{\text{ske}}_{\eta}(z_{\text{kin}} \mid s_t, s_{t+1}). \end{aligned} \tag{2}$$

where $z^{\text{rgb}}_{\text{kin},t}, z^{\text{ske}}_{\text{kin},t} \in \mathbb{R}^{d_{\text{kin}}}$ and $z^{\text{rgb}}_{\text{res},t} \in \mathbb{R}^{d_{\text{res}}}$. The d_{kin} and d_{res} denotes the dimension of the kinematic and residual latents, respectively. We initialize the RGB and skeleton branches through conditional reconstruction:

$$\begin{aligned} \hat{o}_{t+1} &= D_{\text{rgb}}\left(o_t, z^{\text{rgb}}_{\text{kin},t}, z^{\text{rgb}}_{\text{res},t}\right), \\ \hat{s}_{t+1} &= D_{\text{ske}}\left(s_t, z^{\text{ske}}_{\text{kin},t}\right). \end{aligned} \tag{3}$$

Here, skeleton reconstruction establishes the motion reference used for subsequent supervision of the RGB kinematic representation.

Skeleton grounding. To ground the RGB kinematic representation in robot motion, we freeze the trained skeleton encoder and decoder. We reconstruct the next skeleton from the RGB kinematic posterior mean through the frozen decoder and align it with the paired skeleton motion representation. Let $u_t = \mathrm{norm}(\mu^{\mathrm{rgb}}_{\mathrm{kin},t})$ and $v_t = \mathrm{sg}[\mathrm{norm}(\mu^{\mathrm{ske}}_{\mathrm{kin},t})]$, where μ is the posterior mean, norm denotes ℓ_2 normalization, and sg denotes stop-gradient. The grounding and alignment objectives are:

$$\mathcal{L}_{\mathrm{ground}} = \mathbb{E}\left[\ell_{\mathrm{ske}}\left(D_{\mathrm{ske}}(s_t, \mu^{\mathrm{rgb}}_{\mathrm{kin},t}), s_{t+1}\right)\right],$$
$$\mathcal{L}_{\mathrm{align}} = \mathbb{E}\left[1 - u_t^{\top} v_t + \frac{\|u_t - v_t\|_2^2}{d_{\mathrm{kin}}}\right]. \tag{4}$$

Here, ℓ_{ske} is the skeleton reconstruction loss detailed in Appendix B.3.

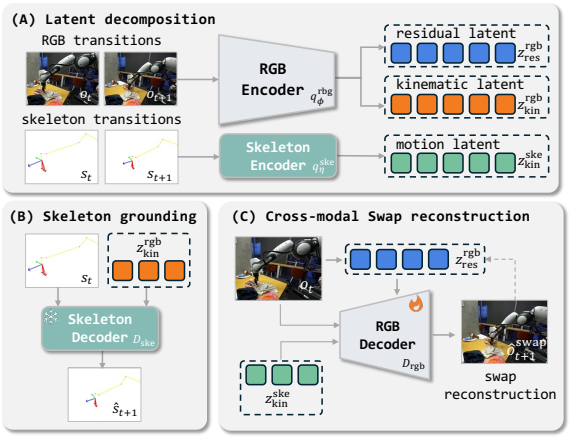


Figure 2: **Skeleton-grounded latent action learning.** (A) RGB and skeleton transition encoding. (B) Grounding through a frozen skeleton decoder. (C) Swap reconstruction using the paired skeleton mean and the same RGB residual sample.

Cross-modal swap reconstruction. To extend motion supervision from the skeleton branch to the RGB decoder, we introduce cross-modal swap reconstruction. We replace the RGB kinematic input with the posterior mean of the paired skeleton motion latent, while retaining the residual sample used in RGB reconstruction:

$$\hat{o}^{\mathrm{swap}}_{t+1} = D_{\mathrm{rgb}}\left(o_t, \mathrm{sg}\left[\mu^{\mathrm{ske}}_{\mathrm{kin},t}\right], z^{\mathrm{rgb}}_{\mathrm{res},t}\right),$$
$$\mathcal{L}_{\mathrm{swap}} = \mathbb{E}\left[\mathrm{MAE}\left(\hat{o}^{\mathrm{swap}}_{t+1}, o_{t+1}\right)\right]. \tag{5}$$

The RGB and swapped reconstructions share the decoder, current observation, and target, encouraging compatible motion conditioning across modalities. Gradients propagate to ϕ through the residual latent, while the skeleton branch remains frozen.

Latent-action objective. The RGB objective combines reconstruction MSE with KL regularization of the kinematic and residual posteriors toward standard Gaussian priors:

$$\mathcal{L}_{\mathrm{rgb}} = \mathbb{E}_{\mathcal{D}, q^{\mathrm{rgb}}_{\phi}}\left[\mathrm{MSE}(\hat{o}_{t+1}, o_{t+1})\right] + \beta \mathbb{E}_{\mathcal{D}}\left[D_{\mathrm{KL}}\left(q^{\mathrm{rgb}}_{\phi,\mathrm{kin}} \,\|\, \mathcal{N}(0, I)\right) + D_{\mathrm{KL}}\left(q^{\mathrm{rgb}}_{\phi,\mathrm{res}} \,\|\, \mathcal{N}(0, I)\right)\right]. \tag{6}$$

Here $\mathcal{D}$ denotes the paired training data and β controls the KL regularization strength. The component posteriors $q^{\mathrm{rgb}}_{\phi,\mathrm{kin}}$ and $q^{\mathrm{rgb}}_{\phi,\mathrm{res}}$ are the marginals of $q^{\mathrm{rgb}}_{\phi}(\cdot \mid o_t, o_{t+1})$.

With the skeleton branch frozen, we optimize the RGB encoder and decoder using:

$$\mathcal{L}_{\mathrm{latent}} = \mathcal{L}_{\mathrm{rgb}} + \lambda_{\mathrm{ground}}\mathcal{L}_{\mathrm{ground}} + \lambda_{\mathrm{align}}\mathcal{L}_{\mathrm{align}} + \lambda_{\mathrm{swap}}\mathcal{L}_{\mathrm{swap}} + \lambda_{\mathrm{var}}\mathcal{L}_{\mathrm{var}}. \tag{7}$$

The regularizer $\mathcal{L}_{\mathrm{var}}$ penalizes low per-dimension standard deviations and cross-covariance between the RGB latent components. Grounding and latent alignment are introduced after conditional reconstruction initialization, followed by swap reconstruction. Initialization losses, regularizer definitions, loss weights, and the optimization schedule are provided in Appendix B.3.

3.2 Action-Conditioned World Modeling

Structured world-model conditioning. To preserve a kinematic conditioning pathway for robot action adaptation, we introduce separate projections for the kinematic and residual components. Given a training clip $o_{t:t+H}$, the frozen RGB encoder extracts $z^{\mathrm{rgb}}_{\mathrm{kin},i}$ and $z^{\mathrm{rgb}}_{\mathrm{res},i}$ from each transition

(o_i, o_{i+1}), where $i \in \{t, \dots, t+H-1\}$. We use separate MLP projections P_b with branch-presence masks $m_b \in \{0, 1\}$, leaving latent standardization implicit in the notation (Appendix A.4):

$$\begin{aligned} \tilde{z}_{b,i} &= m_b z^{\mathrm{rgb}}_{b,i}, \qquad b \in \{\mathrm{kin}, \mathrm{res}\}, \\ h_i &= P_{\mathrm{kin}}(\tilde{z}_{\mathrm{kin},i}; m_{\mathrm{kin}}) + P_{\mathrm{res}}(\tilde{z}_{\mathrm{res},i}; m_{\mathrm{res}}), \end{aligned} \tag{8}$$

where a dropped branch has a zero latent and a zero presence mask. The world model $p_\theta(o_{t+1:t+H} \mid o_t, h_{t:t+H-1})$ learns from video-derived latents through full-backbone fine-tuning after projector warmup. Robot-action latents reuse the kinematic projection.

Branch dropout and world-model training. Residual latents use future observations unavailable at deployment. To retain complementary information while preparing for missing conditions, $\mathcal{R}$ samples one pair of branch masks per clip:

$$(m_{\mathrm{kin}}, m_{\mathrm{res}}) = \begin{cases} (1,1), & \text{full (keep)}, \\ (0,1) \text{ or } (1,0), & \text{drop_kin or drop_res (partial)}, \\ (0,0), & \text{drop_both (null)}. \end{cases} \tag{9}$$

These four whole-branch modes share masks across all H transitions; their step-dependent probabilities are given in Appendix B.4. Partial means dropping one entire branch, not individual coordinates; null drops both branches using fixed zeros in standardized latent space. The supervision target remains the complete future clip $o_{t+1:t+H}$ in every mode. The drop_res mode trains prediction from o_t and kinematic latents, matching action-conditioned inference.

Let ρ collect the learnable conditioning parameters. The world-model generation objective is

$$\mathcal{L}_{\mathrm{wm}}(\theta, \rho) = \mathbb{E}_{\mathcal{D}_{\mathrm{video}}, q^{\mathrm{rgb}}_\phi, \mathcal{R}, \xi}[\ell_{\mathrm{gen}}(\theta; o_{t+1:t+H}, o_t, h_{t:t+H-1}, \xi)], \tag{10}$$

where $\mathcal{D}_{\mathrm{video}}$ contains training clips, ℓ_{gen} is the generation loss, and ξ collects its stochastic variables. The RGB encoder remains frozen.

Robot action adaptation and inference. To condition the world model on robot controls, we train a lightweight MLP adapter $A_\psi : \mathcal{A} \to \mathbb{R}^{d_{\mathrm{kin}}}$, where $\mathcal{A}$ denotes the robot action space. For each transition (o_i, a_i, o_{i+1}), the frozen RGB encoder provides the kinematic posterior mean $\mu^{\mathrm{rgb}}_{\mathrm{kin},i}$ as the regression target:

$$\mathcal{L}_{\mathrm{adapt}}(\psi) = \mathbb{E}_{\mathcal{D}_{\mathrm{robot}}}\left[d_{\mathrm{adapt}}\left(A_\psi(a_i), \mathrm{sg}\left[\mu^{\mathrm{rgb}}_{\mathrm{kin},i} \right] \right) \right], \tag{11}$$

where $\mathcal{D}_{\mathrm{robot}}$ denotes the robot training data and d_{adapt} is the latent regression criterion. Restricting regression to the kinematic component focuses adaptation on the motion representation learned through skeleton supervision. Only ψ is optimized, with the RGB encoder and (θ, ρ) held fixed.

We then fine-tune the full world-model backbone using predicted kinematic latents and a curriculum residual–mask pair $(\tilde{z}^{(s)}_{\mathrm{res},i}, m^{(s)}_{\mathrm{res}})$ (Appendix B.5):

$$\hat{h}^{\mathrm{act}}_i = P_{\mathrm{kin}}(A_\psi(a_i); 1) + P_{\mathrm{res}}\left(\tilde{z}^{(s)}_{\mathrm{res},i}; m^{(s)}_{\mathrm{res}} \right). \tag{12}$$

The action-conditioned generation objective is

$$\mathcal{L}_{\mathrm{wm}} = \mathbb{E}_{\mathcal{D}_{\mathrm{robot}}, \xi}\left[\ell_{\mathrm{gen}}\left(\theta; o_{t+1:t+H}, o_t, \hat{h}^{\mathrm{act}}_{t:t+H-1}, \xi \right) \right]. \tag{13}$$

At inference, $\tilde{z}_{\mathrm{res},i} = 0$ and $m_{\mathrm{res}} = 0$, with no future teacher residual; the current observation and actions specify the rollout:

$$\hat{o}_{t+1:t+H} \sim p_\theta\left(\cdot \mid o_t, \hat{h}^{\mathrm{act}}_{t:t+H-1} \right). \tag{14}$$

4 Experiment

We evaluate the effect of the kinematic–residual decomposition on action-conditioned world modeling, and examine motion transfer and rollout evaluation as downstream applications. Specifically, we investigate the following questions:

- **RQ1 (World modeling):** Does adapting robot actions to a dedicated kinematic component improve motion accuracy while preserving visual fidelity? (Section 4.2)
- **RQ2 (Motion representation):** Do skeleton-grounded kinematic latents support more accurate linear decoding of robot motion than unstructured latent actions? (Section 4.3)
- **RQ3 (Motion transfer):** Can a kinematic representation learned from robot interactions transfer human motion to generated robot rollouts in a zero-shot setting? (Section 4.4)
- **RQ4 (Policy evaluation):** How closely do open-loop world-model rollouts reproduce policy rankings and success rates measured in real robot executions? (Section 4.5)

4.1 Experimental Setup

Data. DROID (Khazatsky et al., 2024) pairs visual observations with robot actions, allowing us to assess whether predicted dynamics follow robot controls. We train the latent action model and world model on DROID, incorporating AgiBot (Bu et al., 2025) to broaden the range of robot interactions observed during training. Evaluation uses 400 DROID clips, each containing one conditioning frame and 12 future frames at 480×640 resolution. More details are provided in Appendix A.

Evaluation Metrics. To quantify trajectory, direction, and magnitude agreement, we report 2D end-effector average displacement error (ADE, px), direction macro-F1 ($\mathrm{F1}_{\mathrm{dir}}$), normalized magnitude MAE (L_1^{mag}), and magnitude Pearson correlation (r_{mag}). PSNR, SSIM, and LPIPS (Zhang et al., 2018) assess pixel, structural, and perceptual fidelity, respectively, with tLPIPS reported as a temporal diagnostic. More details are provided in Appendix C.1.

Implementation Details. We implement the latent action model using Transformer VAEs with $d_{\mathrm{kin}} = d_{\mathrm{res}} = 32$ and use Cosmos-predictor-2B as world model backbone. We train with AdamW, using stage-specific learning rates and global batch sizes. SkeLAM training uses NVIDIA A100 GPUs and requires approximately 2 days for one complete run. Complete architectural specifications, loss weights, optimization schedules, and inference settings are provided in Appendix B.

Baselines. We compare SkeLAM with latent-action world models DreamDojo (Gao et al., 2026) and CD-LAM (Wei et al., 2026). Alternative conditioning strategies include direct robot actions in IRASim (Zhu et al., 2025), Ctrl-World (Guo et al., 2026), Cosmos-Predict2.5 (Ali et al., 2025), Cosmos3 (Agarwal et al., 2026), and GE-Sim (Liao et al., 2025), and geometric guidance in EnerVerse-AC (Jiang et al., 2025) and OSCAR (Wu & Gao, 2026). TesserAct (Zhen et al., 2025), and GE-Base (Liao et al., 2025) serve as text-conditioned references for visual fidelity, with detailed baseline configurations provided in Appendix C.

4.2 Action-Conditioned World Modeling

Quantitative results. To assess how accurately SkeLAM simulates the consequences of robot actions, we evaluate motion accuracy and visual fidelity on DROID. Table 1 shows that SkeLAM ranks first on seven of eight metrics among the reported action-conditioned models, including all four visual fidelity metrics. Compared with DreamDojo and CD-LAM, SkeLAM reduces ADE by 19.5% and 23.7%, respectively, alongside LPIPS reductions of 11.1% and 17.4%. These improvements indicate closer agreement with both the target robot trajectories and visual observations. Ctrl-World and OSCAR achieve higher direction F1, while SkeLAM attains lower trajectory error and stronger visual fidelity. In particular, compared with Ctrl-World, SkeLAM reduces ADE from 42.96 to 41.15 px and LPIPS from 0.4029 to 0.1310. The combined improvements in trajectory accuracy and visual fidelity support the generation of robot motion together with the associated scene dynamics.

Qualitative results. To evaluate whether generated rollouts reproduce robot motion while preserving scene structure, we compare predictions with ground truth in Figure 3. The overlaid trajectories reveal deviations from the reference motion in DreamDojo and OSCAR, accompanied by visible arm deformation in later frames. Ctrl-World produces a shorter trajectory and local gripper distortions, while Cosmos-Predict2.5 alters the apparent table scale and bowl position. SkeLAM more closely follows the curved reference trajectory while retaining the tabletop layout, although local gripper errors remain. This example highlights the importance of assessing motion correspondence alongside scene consistency. More qualitative results are provided in Appendix. D.5.

Table 1: **World modeling results.** Control denotes the inference-time conditioning interface. Text-conditioned models serve as visual references. **Bold** and underlined values indicate the best and second-best action-conditioned results. † denotes large-scale interaction pretraining.

		Visual fidelity				Motion accuracy			
Method	Control	PSNR↑	SSIM↑	LPIPS↓	MSE↓	ADE↓	$F1_{dir}$ ↑	L_1^{mag} ↓	r_{mag} ↑
Text-conditioned references									
TesserAct	Text	15.08	0.7400	0.2561	0.0349	83.76	0.2950	0.0797	0.0823
GE-Base	Text	12.51	0.4530	0.5864	0.0596	55.30	0.3006	0.0956	0.0504
Action- and geometry-conditioned models									
IRASim	Action emb.	11.55	0.4137	0.5993	0.0738	67.11	0.2616	0.0843	0.1193
Ctrl-World	Action emb.	16.34	0.5910	0.4029	0.0255	42.96	**0.5888**	0.0511	0.5154
Cosmos-Predict2.5	Action emb.	11.51	0.4209	0.6018	0.0758	60.76	0.2375	0.0839	0.1049
Cosmos3	Action emb.	16.98	0.7630	0.2186	0.0270	67.23	0.3278	0.0725	0.2518
EnerVerse-AC	Action map	14.13	0.5157	0.4139	0.0415	59.63	0.2433	0.0982	-0.0050
GE-Sim	Action map	12.53	0.4550	0.4950	0.0593	55.57	0.2799	0.0941	0.0091
OSCAR	Skeleton	12.57	0.4471	0.4788	0.0587	43.74	0.5810	0.0530	0.5029
Models with learned latent actions									
DreamDojo†	a	18.91	0.8057	0.1473	0.0168	51.11	0.4989	0.0667	0.3335
CD-LAM†	$a \to z$	18.34	0.7910	0.1586	0.0181	53.90	0.4903	0.0690	0.3179
SkeLAM (ours)	$a \to z_{kin}$	**19.35**	**0.8340**	**0.1310**	**0.0151**	**41.15**	0.5465	**0.0510**	**0.5277**

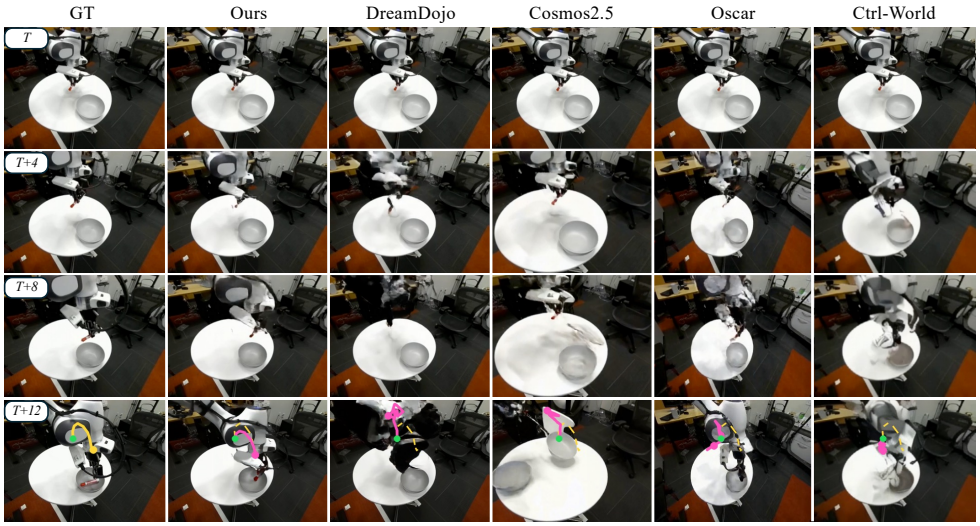


Figure 3: **Qualitative comparison of action-conditioned world modeling.** Ground-truth and generated observations are compared at selected time steps, highlighting differences in end-effector motion, manipulated-object geometry, and scene appearance.

Ablation studies. To assess how each component supports action-conditioned generation, we compare variants in Table 2. **(i) Skeleton supervision** grounds the RGB kinematic latent in robot motion; removing it increases ADE from 41.15 to 64.59 and reduces magnitude correlation from 0.5277 to 0.2009, supporting geometric supervision for learning motion representations suitable for action conditioning. **(ii) Cross-modal swap** couples skeleton supervision to RGB generation beyond latent alignment; removing it increases ADE to 66.74 and reduces magnitude correlation to 0.1897, while LPIPS rises from 0.1310 to 0.1436, supporting reconstruction-level RGB–skeleton alignment for downstream action conditioning. **(iii) Residual latent** retains complementary scene dynamics; zeroing it from world-model pretraining maintains similar visual fidelity (0.1305 vs. 0.1310 LPIPS) while ADE increases to 43.37 and magnitude correlation decreases to 0.4737, suggesting that complementary visual information benefits motion conditioning. **(iv) Action adaptation** supervises the action-to-kinematic mapping; omitting adapter regression increases ADE to 65.84, supporting explicit supervision of the mapping from robot actions to the learned kinematic representation. Further details and adaptation-budget analysis are provided in Appendix D.

Table 2: **Component ablations.** Each variant removes one component from the full model under the same training budget. **Bold** and underlined values indicate the best and second-best results.

	Visual fidelity				Motion accuracy			
Variant	PSNR↑	SSIM↑	LPIPS↓	MSE↓	ADE↓	$\mathrm{F1}_{\mathrm{dir}}$ ↑	L_1^{mag} ↓	r_{mag} ↑
Full SkeLAM	19.35	0.8340	0.1310	0.0151	**41.15**	**0.5465**	**0.0510**	**0.5277**
w/o skeleton supervision	18.18	0.8206	0.1538	0.0188	64.59	0.3268	0.0723	0.2009
w/o cross-modal swap	19.30	0.8330	0.1436	0.0192	66.74	0.3132	0.0752	0.1897
w/o residual latent	**19.38**	**0.8348**	**0.1305**	**0.0150**	43.37	0.5263	0.0547	0.4737
w/o action adaption	18.14	0.8196	0.1545	0.0189	65.84	0.3259	0.0739	0.1603

4.3 Motion Representation Analysis

Motion decodability. To examine how robot motion is encoded, we train linear probes on frozen latent representations. As shown in Table 3, the kinematic component achieves an EEF MAE of 5.99×10^{-3} and a magnitude macro-F1 of 0.7467, improving on both latent-action baselines. Within SkeLAM, it also outperforms the residual component on all three metrics. Concatenating the residual increases EEF R^2 from 0.3200 to 0.3277 and magnitude macro-F1 from 0.7467 to 0.7482. The kinematic component thus approaches the probing performance of the full representation using half the latent dimensions, supporting its specialization for displacement and magnitude encoding.

Cross-modal compatibility. To assess cross-modal motion conditioning, we intervene on the latent inputs to the RGB decoder. Figure 4 shows that substituting paired and mismatched skeleton kinematic means increases reconstruction MAE by 0.53% and 11.67%, respectively. SSIM exhibits the same ordering. Zeroing the kinematic component increases MAE by 5.78%, compared with 0.61% for the residual. The greater degradation under mismatched substitution than kinematic removal indicates sensitivity to motion correspondence. These results support compatible RGB and skeleton conditioning and a functional role for the kinematic component in reconstruction.

Table 3: **Motion probing.** Linear probes predict EEF displacement (MAE, R^2) and motion magnitude (macro-F1) from frozen representations. Latents are 32D except the 64D concatenation. †: privileged skeleton input.

Representation	MAE↓	R^2_{EEF} ↑	$\mathrm{F1}_{\mathrm{mag}}$ ↑
DreamDojo	0.01620	0.0078	0.7183
CD-LAM	0.01615	0.0079	0.7006
SkeLAM z_{kin}	0.00599	0.3200	0.7467
SkeLAM z_{res}	0.00678	0.1879	0.5652
$[z_{\mathrm{kin}}; z_{\mathrm{res}}]$	0.00597	0.3277	0.7482
Skeleton teacher†	0.00552	0.4312	0.7233

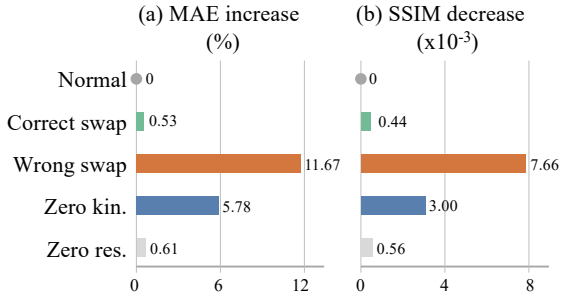


Figure 4: **Latent action interventions.** Reconstruction MAE increase and SSIM decrease relative to Normal. Correct and wrong swaps use paired and mismatched skeleton kinematic means.

4.4 Zero-Shot Human-to-Robot Motion Transfer

To examine whether the learned motion representation transfers across embodiments, we use human demonstration videos from UniSkill (Kim et al., 2025) as motion prompts. The RGB encoder extracts latent actions from these videos to directly condition SkeLAM, with a robot observation specifying the initial scene. All model parameters remain fixed during transfer. We compare with the pretrained checkpoint of DreamDojo (Gao et al., 2026), which supports latent-action conditioning. As shown in Figure 5, SkeLAM generates distinct end-effector motions corresponding to the human demonstrations: a downward reach for the towel-pushing prompt and an upward movement for the tissue-pulling prompt. DreamDojo exhibits limited end-effector displacement in both examples. These results provide qualitative evidence that human-derived latent actions can guide robot motion despite differences in embodiment and scene appearance.

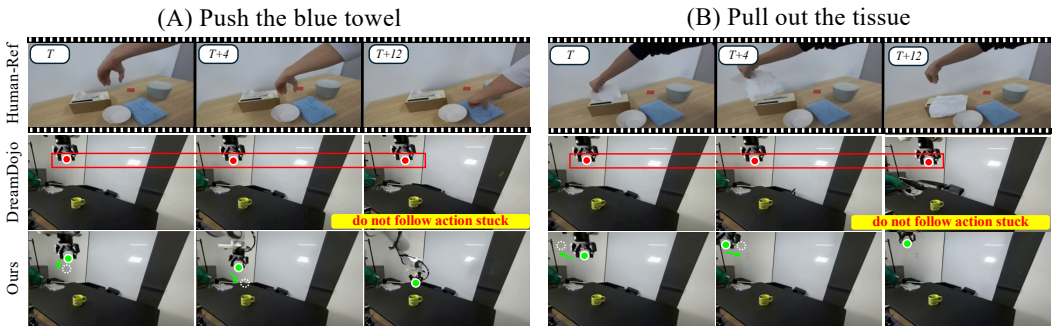


Figure 5: **Zero-shot human-to-robot motion transfer.** Human videos from UniSkill (Kim et al., 2025) provide latent conditions for robot rollout generation. Markers indicate end-effector positions.

Table 4: **Open-loop policy evaluation on RoboArena.** Seven policies are evaluated across 15 sessions (105 episodes). Episode-level metrics include balanced accuracy (BA, %), success recall (%), and Matthews correlation coefficient (MCC). Policy-level metrics include Spearman rank correlation (ρ), Pearson correlation (r), and success-rate MAE (Δ_{SR}).

	Episode-level agreement			Policy-level agreement		
Model	BA↑	Recall↑	MCC↑	$\rho\uparrow$	$r\uparrow$	$\Delta_{\text{SR}}\downarrow$
Cosmos-Predict2.5	48.44	0.00	-0.0525	-0.0756	-0.1936	7.62
SkeLAM	63.02	33.33	0.2483	0.5347	0.6091	4.76

4.5 World Model for Robot Policy Evaluation

To assess whether generated rollouts preserve task outcomes and relative policy performance, we evaluate seven policies across 15 RoboArena sessions (105 episodes). Both models receive the same initial observations and recorded actions. Human annotators label task success in generated rollouts for comparison with real execution outcomes. Table 4 reports 33.33% success recall and 63.02% balanced accuracy for SkeLAM, compared with 0.00% and 48.44% for Cosmos-Predict2.5. At the policy level, success-rate MAE decreases from 7.62 to 4.76 percentage points, with positive Spearman (ρ=0.5347) and Pearson (r=0.6091) correlations for SkeLAM. Together, these metrics indicate improved agreement in both absolute policy success rates and relative policy ordering. Figure 6 compares predicted and real success rates for individual policies, with proximity to the diagonal indicating closer agreement. The results support agreement in both individual task outcomes and relative policy performance under open-loop action replay. Nevertheless, the modest success recall indicates that reproducing successful task outcomes remains challenging. Further evaluation details are provided in Appendix F.

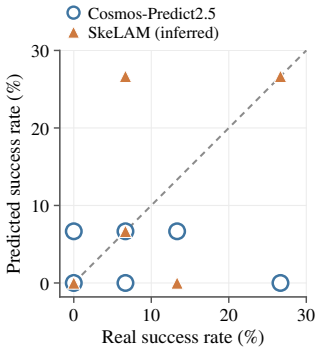


Figure 6: **Policy-level agreement.** The dashed line indicates perfect agreement.

5 Conclusion

We presented SkeLAM, a skeleton-guided latent action model for controllable robot world modeling. The kinematic component provides a motion-grounded target for robot action adaptation, while the residual retains complementary visual dynamics. Cross-modal swap reconstruction and branch dropout support reuse of the learned kinematic pathway for action-conditioned generation. Experiments on DROID show improved motion accuracy and visual fidelity, with representation analyses supporting kinematic specialization. Qualitative results demonstrate zero-shot human-to-robot motion transfer, while policy evaluation yields success-rate estimates consistent with real robot executions. These results support structuring latent actions around robot motion while retaining complementary information for generation. Future work will extend the current recorded-action replay protocol to closed-loop policy evaluation. A key question is how prediction errors affect subsequent policy decisions and task outcomes over longer interaction horizons.

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.

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.

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 *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4603–4623. PMLR, 2024. URL https://proceedings.mlr.press/v235/bruce24a.html.

Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Xindong He, Xu Huang, et al. AgiBot World Colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2025. URL https://arxiv.org/abs/2503.06669.

Xizhou Bu, Jiexi Lyu, Fulei Sun, Ruichen Yang, Zhiqiang Ma, and Wei Li. LAOF: Robust Latent Action Learning with Optical Flow Constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27334–27344, 2026. URL https://arxiv.org/abs/2511.16407.

Shenyuan Gao, Siyuan Zhou, Yilun Du, Jun Zhang, and Chuang Gan. AdaWorld: Learning adaptable world models with latent actions. In *International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2503.18938.

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, Qianli Ma, Seungjun Nah, Loic Magne, Jiannan Xiang, Yuqi Xie, Ruijie Zheng, Dantong Niu, You Liang Tan, K. R. Zentner, George Kurian, Suneel Indupuru, Pooya Jannaty, Jinwei Gu, Jun Zhang, Jitendra Malik, Pieter Abbeel, Ming-Yu Liu, Yuke Zhu, Joel Jang, and Linxi "Jim" Fan. DreamDojo: A generalist robot world model from large-scale human videos. *arXiv preprint arXiv:2602.06949*, 2026. URL https://arxiv.org/abs/2602.06949.

Quentin Garrido, Tushar Nagarajan, Basile Terver, Nicolas Ballas, Yann LeCun, and Michael Rabbat. Learning Latent Action World Models In The Wild. *arXiv preprint arXiv:2601.05230*, 2026. URL https://arxiv.org/abs/2601.05230.

Manish Kumar Govind, Dominick Reilly, Pu Wang, and Srijan Das. UniLACT: Depth-Aware RGB Latent Action Learning for Vision-Language-Action Models. *arXiv preprint arXiv:2602.20231*, 2026. URL https://arxiv.org/abs/2602.20231.

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. URL https://arxiv.org/abs/2608.06332.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-World: A Controllable Generative World Model for Robot Manipulation. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.10125.

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

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, Vitor Guizilini, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Muhammad Zubair Irshad, Donovon Jackson, Charlotte Le, Yunshuang Li, Kevin Lin, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. DROID: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024. URL https://arxiv.org/abs/2403.12945.

Hanjung Kim, Jaehyun Kang, Hyolim Kang, Meedeum Cho, Seon Joo Kim, and Youngwoon Lee. Uniskill: Imitating human videos via cross-embodiment skill representations. In *9th Annual Conference on Robot Learning*, 2025.

Anthony Liang, Pavel Czempin, Matthew M. Hong, Yutai Zhou, Jingzhen Wang, Erdem Biyik, and Stephen Tu. CLAM: Continuous latent action models for robot learning from unlabeled demonstrations. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2026. URL https://arxiv.org/abs/2505.04999.

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

Jiahao Liu, Zhongpu Xia, Shuai Tian, Huangrui Li, Yuhang Zheng, Ning Ma, Xin Fu, Xiaotian Liu, Jing Li, Yixian Li, ShangQing Zhou, Zebin Xing, Linbo Wang, Chaoyue Li, Haoran Li, and Dongbin Zhao. WALA Learning Executable Latent Actions from Action-Labeled Demonstrations and Action-Free Videos. *arXiv preprint arXiv:2607.11397*, 2026a. URL https://arxiv.org/abs/2607.11397.

Yichen Liu, Puzhen Yuan, Xiang Zhu, Yanjiang Guo, and Jianyu Chen. GALA: Geometry-Aware Latent Action Modeling for Vision-Language-Action Model Pretraining across Embodiments. *arXiv preprint arXiv:2609.21948*, 2026b. URL https://arxiv.org/abs/2609.21948.

Alexander Nikulin, Ilya Zisman, Denis Tarasov, Nikita Lyubaykin, Andrei Polubarov, Igor Kiselev, and Vladislav Kurenkov. Latent action learning requires supervision in the presence of distractors. In *International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2502.00379.

Zhenhao Shen, Jiaqi Liang, Jasper Lu, Feng Jiang, Yuran Wang, Chuanbo Wei, Jiayi Liu, Jianchun Yang, Qize Yu, Jiadi You, Ce Hao, Guanqi He, Chen Xie, and Ruihai Wu. LD4WAM: Learning Latent Dynamics from Human Videos for World Action Models. *arXiv preprint arXiv:2608.22403*, 2026. URL https://arxiv.org/abs/2608.22403.

Lirui Wang, Kevin Zhao, Chaoqi Liu, and Xinlei Chen. Learning Real-World Action-Video Dynamics with Heterogeneous Masked Autoregression. *arXiv preprint arXiv:2502.04296*, 2025a. URL https://arxiv.org/abs/2502.04296.

Yuang Wang, Chao Wen, Haoyu Guo, Sida Peng, Minghan Qin, Hujun Bao, Xiaowei Zhou, and Ruizhen Hu. Precise action-to-video generation through visual action prompts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025b. URL https://arxiv.org/abs/2508.13104.

Yufan Wei, Kun Zhou, Lingjun Mao, Zijun Zhang, Ziming Xu, Ziqiao Xi, Shuang Liang, Ruobing Han, Yuchen Yan, Xinyue Wang, Fan Feng, and Biwei Huang. Causally debiased latent action model for embodied action conditioned world models. *arXiv preprint arXiv:2607.09185*, 2026.

Jialong Wu, Shaofeng Yin, Ningya Feng, Xu He, Dong Li, Jianye Hao, and Mingsheng Long. iVideoGPT: Interactive VideoGPTs are scalable world models. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2405.15223.

Zhuoyuan Wu and Jun Gao. OSCAR: Omni-Embodiment Action-Conditioned World Model for Robotics. *arXiv preprint arXiv:2606.04463*, 2026. URL https://arxiv.org/abs/2606.04463.

Yifan Xie, Hekun Tian, Jinkun Liu, YuAn Wang, Qiao Sun, and Wenbo Ding. GeoLAM: Learning Geometry-Grounded Latent Actions from Unlabeled Human Videos. *arXiv preprint arXiv:2609.17099*, 2026. URL https://arxiv.org/abs/2609.17099.

Mutian Xu, Tianbao Zhang, Tianqi Liu, Zhaoxi Chen, Xiaoguang Han, and Ziwei Liu. Kinema4D: Kinematic 4D World Modeling for Spatiotemporal Embodied Simulation. *arXiv preprint arXiv:2603.16669*, 2026. URL https://arxiv.org/abs/2603.16669.

Jiange Yang, Yansong Shi, Haoyi Zhu, Mingyu Liu, Kaijing Ma, Yating Wang, Gangshan Wu, Tong He, and Limin Wang. CoMo: Learning Continuous Latent Motion from Internet Videos for Scalable Robot Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026a. URL https://arxiv.org/abs/2505.17006.

Jiarui Yang, Jiale Zhange, Jiawei Li, Hang Guo, Wen Huang, Jinpeng Wang, Peidong Liu, and Shu-Tao Xia. LAWM-3D: Learning 3D-Aware Latent Actions from Human Videos for Generalizable Robot World Models. *arXiv preprint arXiv:2608.05706*, 2026b. URL https://arxiv.org/abs/2608.05706.

Sherry Yang, Yilun Du, Seyed Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/c4d66eae503694424123b93ac0fbaf17-Abstract-Conference.html.

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 *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/45d74e190008c7bff2845ffc8e3facd3-Abstract-Conference.html.

Chuheng Zhang, Tim Pearce, Pushi Zhang, Kaixin Wang, Xiaoyu Chen, Wei Shen, Li Zhao, and Jiang Bian. What do latent action models actually learn? In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2506.15691.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Haoyu Zhen, Qiao Sun, Hongxin Zhang, Junyan Li, Siyuan Zhou, Yilun Du, and Chuang Gan. Tesseract: learning 4d embodied world models. *arXiv preprint arXiv:2504.20995*, 2025.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. IRASim: A fine-grained world model for robot manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025. URL https://arxiv.org/abs/2406.14540.

APPENDIX CONTENTS

A Data Preparation and Temporal Alignment **14**

A.1 Training data and evaluation cohort . . . 14

A.2 RGB and skeleton preprocessing . . . 14

A.3 Video clips and action alignment . . . 15

A.4 Deterministic latent extraction . . . 15

B Architecture and Training Details **16**

B.1 Model initialization and training stages . . . 16

B.2 Latent action architecture . . . 16

B.3 Latent-action losses and optimization . . . 17

B.4 World-model conditioning and full fine-tuning . . . 18

B.5 Robot-action adaptation . . . 18

C Evaluation Protocols and Baseline References **20**

C.1 Action-conditioned video prediction . . . 20

C.2 Baseline conditioning interfaces . . . 21

C.3 Linear motion probes . . . 22

C.4 Cross-modal latent interventions . . . 22

D Additional Experimental Analysis **23**

D.1 Effect of world-model adaptation budget . . . 23

D.2 Extended DROID results . . . 23

D.3 Extended evaluation on AgiBot World . . . 23

D.4 Component ablation settings and analysis . . . 25

D.5 Additional qualitative comparisons . . . 25

E Zero-Shot Human-to-Robot Transfer Protocol **25**

F Open-Loop Robot Policy Evaluation **27**

F.1 Evaluation unit and outcome labels . . . 27

F.2 Episode-level agreement . . . 27

F.3 Policy-level agreement . . . 27

G AI Use Statement **27**

Reading guide. Data and preprocessing appear in Appendix A, architectures and training in Appendix B, evaluation protocols in Appendix C, and additional analyses in Appendix D. Appendices E and F describe human-to-robot transfer and policy evaluation. Appendices G states the usage of AI assistant.

A Data Preparation and Temporal Alignment

A.1 Training data and evaluation cohort

We use DROID and AgiBot World (Bu et al., 2025) in all three training stages: latent-action learning, world-model pretraining, and robot-action adaptation. Table 5 summarizes the stage-wise training inputs and DROID training clip counts. The DROID evaluation cohort contains 400 clips, each with one observed RGB frame, 12 future frames, and aligned controls for action-conditioned models. The main comparison in Table 1 reports the 400-clip DROID subset. Training-step budgets and checkpoint selection are reported in Appendix B.1.

Table 5: **Training data usage across stages.** A clip is a source data unit; Stage 1 samples frame pairs from clips, whereas Stage 2/3 use temporally continuous prediction windows. Each of Stage 2A, 2B, 3A, and 3B uses 45,358 clips. All counts in this table refer to training data used per substage; the evaluation cohorts are reported separately.

Training stage	Inputs and supervision	DROID train clips
Stage 1A/B/C	Paired RGB frames and corresponding rendered skeleton observations; reconstruction, grounding, alignment, and swap supervision	60,000 clips
Stage 2A/B	Continuous 13-frame RGB windows and cached RGB-teacher kinematic/residual latents; first-frame-conditioned video prediction	45,358 clips
Stage 3A/B	Aligned 12-step robot-action blocks and teacher latents for adapter fitting; RGB video targets for generation training	45,358 clips

RGB observations and skeleton renders are matched by episode, camera, and frame index. The latent-action model samples within-clip frame pairs at multiple temporal gaps. World-model windows are constructed from the original videos, preserving temporal continuity within each target. Skeleton observations supervise Stage 1 and are not required for robot-action-conditioned generation. Episode identity is retained in the manifests so that overlapping windows from one episode remain in the same partition. Action and latent normalization statistics are computed on the training partition and reused during evaluation.

Evaluation cohort selection. For DROID, we deterministically select 100 trajectories from each motion quartile, with additional stratification by camera identifier. Each selected trajectory contributes one clip. Each of the 400 DROID clips is generated once using the inference protocol in Appendix B.5.

A.2 RGB and skeleton preprocessing

The RGB and skeleton inputs to the latent action model are letterboxed to $H \times W = 240 \times 320$ pixels in Stage 1. For an original frame of size $H_o \times W_o$, we compute

$$s = \min\left(\frac{240}{H_o}, \frac{320}{W_o}\right), \qquad H' = \mathrm{round}(H_o s), \quad W' = \mathrm{round}(W_o s).$$

The frame is resized to $H' \times W'$ using OpenCV's `INTER_AREA` interpolation and centered on a zero-filled $240 \times 320 \times 3$ canvas. The top and left padding widths are $\lfloor (240 - H')/2 \rfloor$ and $\lfloor (320 - W')/2 \rfloor$, respectively; any extra pixel from an odd padding width is placed at the bottom or right. Pixel intensities are then normalized to $[0, 1]$ by dividing by 255.

We use the RGB videos and pre-rendered skeleton observations provided by OSCAR (Wu & Gao, 2026) for the corresponding DROID and AgiBot World subsets. Robot kinematics, camera projection, and skeleton rasterization follow the supplied OSCAR data; we refer to its rendering protocol for details. After letterboxing, we derive a binary foreground mask from the target skeleton frame s_{t+1} by selecting pixels whose RGB channel sum exceeds 15 on the 8-bit scale, before intensity normalization. We dilate this mask once using a 5×5 elliptical structuring element (radius 2). The mask is used only for target skeleton reconstruction losses (Appendix B.3) and is not concatenated with the model inputs.

For Stage 2/3, the world model receives a 4:3 center crop resized to $H \times W = 480 \times 640$ pixels. The frozen latent action encoder retains its Stage 1 input resolution when extracting cached latents. Both preprocessing paths use the same original frame indices; the world-model target is read from the original RGB video. All resolutions in this appendix are reported as height $\times$ width.

A.3 Video clips and action alignment

Each 13-frame DROID clip is sampled from 60-FPS video with a stride of four source frames, yielding an effective frame rate of 15 FPS and a 0.8-second span:

$$f_k = f_0 + 4k, \qquad k = 0, \ldots, 12. \tag{15}$$

The first frame provides the visual condition and the remaining 12 frames are prediction targets. The window-construction setting advances the start index by 24 source frames, corresponding to a 0.4-second hop. The manifest records the original indices and timestamps rather than relying only on a nominal video frame rate.

The final conditioning signal is a 12-step executed end-effector state-delta proxy, computed from recorded robot states rather than joint-space control commands. Each of the 12 transitions is associated with one latent target and one state-derived action proxy:

$$a_k \longleftrightarrow (o_k, o_{k+1}) \longleftrightarrow z^{\mathrm{rgb}}_{\mathrm{kin},k}, \qquad k = 0, \ldots, 11. \tag{16}$$

Each per-arm proxy has seven coordinates,

$$a_k = [\Delta p_k; \Delta\omega_k; g_{k+1}] \in \mathbb{R}^7, \tag{17}$$

where $\Delta p_k = p_{k+1} - p_k \in \mathbb{R}^3$ is the translation delta in the robot base frame, $\Delta\omega_k \in \mathbb{R}^3$ is the rotation-vector representation of the relative $\mathrm{SO}(3)$ rotation, and g_{k+1} is the destination gripper openness. The deltas are computed between the states aligned with consecutive sampled frames; they are not accumulated relative to the beginning of the clip. This seven-dimensional interface is used for the DROID action adapter. An action-validity indicator records whether an entire transition is valid and is broadcast over the action or latent coordinates when losses are averaged.

A.4 Deterministic latent extraction

We freeze the final Stage 1 RGB encoder and evaluate it in deterministic mode. For each adjacent pair in a world-model clip, we extract a 64-dimensional RGB latent and split it into two 32-dimensional components:

$$z^{\mathrm{rgb}}_k = [z^{\mathrm{rgb}}_{\mathrm{kin},k}; z^{\mathrm{rgb}}_{\mathrm{res},k}]. \tag{18}$$

Thus, a clip produces two arrays of shape 12×32. Consistent with the world-model conditioning notation in Section 3.2, we use z for the exported latent components. For deterministic extraction, their values are the corresponding posterior means; Stage 1 reconstruction training uses reparameterized latent samples. The skeleton encoder is not called to provide world-model conditions.

For a latent component z, conditioning uses the elementwise standardized representation

$$\bar{z} = (z - m_z) \oslash \max(s_z, \epsilon), \tag{19}$$

where m_z and s_z are estimated from training data, $\oslash$ denotes elementwise division, and the floor prevents division by a vanishing standard deviation. The same kinematic statistics are used when training the action adapter and when applying its predictions to the world model. Section 3.2 and Appendix B.5 suppress this standardization in their notation, using z for conditioning latents and μ for adapter targets; the bar in Eq. (19) makes the preprocessing operation explicit. The offline cache associates each latent sequence with its episode, camera, frame indices, timestamps, preprocessing configuration, and Stage 1 checkpoint identifier. No gradient from world-model training propagates into this cache or the latent action model.

The Stage 3 action cache pairs these teacher latents with the aligned 12-transition state-derived action blocks described in Appendix A.3. Both caches are constructed offline; cache construction is a data-preparation operation, not an additional optimization stage.

B Architecture and Training Details

B.1 Model initialization and training stages

SkeLAM comprises structured latent-action learning (Stage 1A/B/C), video world-model training (Stage 2A/B), and robot-action adaptation (Stage 3A/B). Phase A/B/C are alternative names for Stage 1A/B/C only. Figure 7 summarizes the seven substages, selected checkpoints, and model and data dependencies.

The RGB and skeleton Transformer-VAE branches are trained from random initialization. The world-model backbone is initialized from official Cosmos-Predict2.5 2B weights (Ali et al., 2025), independently of the pretrained DreamDojo baseline. The video tokenizer, text encoder, and final Stage 1 teacher remain frozen throughout Stage 2/3.

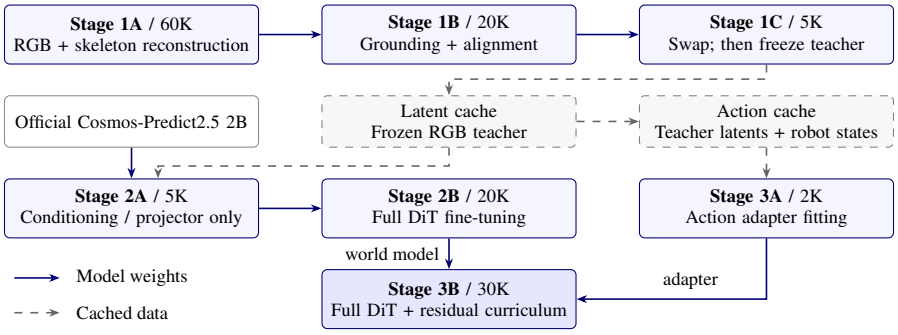


Figure 7: **Training workflow.** Solid arrows transfer model weights; dashed arrows indicate cached data. Stage 3B starts from the Stage 2B world model and Stage 3A adapter and updates all DiT backbone weights. Cache construction does not add optimization stages.

B.2 Latent action architecture

The RGB branch parameterizes a 64-dimensional Gaussian posterior, split into 32-dimensional kinematic and residual components. The skeleton branch parameterizes a 32-dimensional kinematic posterior. Both branches use the Transformer configuration in Table 6. Grounding conditions the frozen skeleton decoder on the RGB kinematic posterior mean. Latent alignment uses the paired RGB and skeleton kinematic posterior means, while cross-modal swapping uses the frozen skeleton posterior mean. In a swap reconstruction, the residual is the *same sampled residual* used in the corresponding ordinary RGB reconstruction.

Table 6: **Latent action model architecture.** These are the Transformer settings of the 32+32-dimensional checkpoint used to export world-model conditions.

Setting	Value
Input resolution (height × width)	240×320
Patch size	16×16
Transformer width	1,024
Encoder / decoder blocks	16 / 16
Attention heads	16
Temporal attention	Causal
Transformer dropout	0
RGB posterior dimensions	$d_{\text{kin}} = 32$, $d_{\text{res}} = 32$
Skeleton posterior dimension	32
Stage 2/3 exported representation	RGB posterior mean

B.3 LATENT-ACTION LOSSES AND OPTIMIZATION

Initialization and RGB reconstruction. Phase A initializes the two branches by reconstructing the next RGB frame and next skeleton observation from their respective current observations and transition latents. The RGB variational reconstruction objective is Eq. (6), with $\beta = 2 \times 10^{-4}$. The RGB reconstruction term is a mean-squared pixel error. The kinematic and residual KL terms are counted once; no additional KL penalty is added to a swapped reconstruction.

Skeleton reconstruction and grounding. The criterion ℓ_{ske} in Eq. (4) combines foreground color reconstruction, mask supervision, and background suppression; we specify these components below. Let s be a target colored skeleton image, m its foreground mask, and $\hat{s}$ the reconstructed image. Let $\hat{m}$ denote the foreground prediction used by the skeleton loss. With P pixels and C_s color channels, the foreground and background terms can be written as

$$\ell_{\text{fg}}(\hat{s}, s, m) = \frac{\sum_{p,c} m_p |\hat{s}_{pc} - s_{pc}|}{C_s \sum_p m_p + \epsilon}, \tag{20}$$

$$\ell_{\text{bg}}(\hat{s}, m) = \frac{\sum_{p,c} (1 - m_p) |\hat{s}_{pc}|}{C_s \sum_p (1 - m_p) + \epsilon}. \tag{21}$$

The mask term combines binary cross-entropy and soft Dice loss,

$$\ell_{\text{mask}}(\hat{m}, m) = \text{BCE}(\hat{m}, m) + 1 - \frac{2 \sum_p \hat{m}_p m_p + \epsilon}{\sum_p \hat{m}_p + \sum_p m_p + \epsilon}. \tag{22}$$

The skeleton criterion used for grounding is

$$\ell_{\text{ske}} = \ell_{\text{fg}} + 0.5 \ell_{\text{mask}} + 0.05 \ell_{\text{bg}}. \tag{23}$$

In Eq. (4), $\ell_{\text{ske}}(\hat{s}, s)$ denotes the mask-aware criterion above, with the target mask derived as in Appendix A.2 and used only in the loss. Grounding conditions the frozen decoder on the RGB kinematic posterior mean $\mu^{\text{rgb}}_{\text{kin},t}$. Latent alignment compares the paired RGB and skeleton kinematic posterior means. Gradients through the frozen decoder update the RGB encoder; the skeleton encoder output is detached.

Alignment and regularization. Alignment applies the cosine-plus-MSE criterion in Eq. (4) to ℓ_2-normalized RGB and skeleton kinematic means. The skeleton mean is the fixed target. For a batch representation $Z \in \mathbb{R}^{B \times d}$, define

$$V_\gamma(Z) = \frac{1}{d} \sum_{j=1}^{d} \max(0, \gamma - \text{Std}_b(Z_{bj})), \qquad \gamma = 0.5. \tag{24}$$

Writing $\text{Cov}(Z_{\text{kin}}, Z_{\text{res}})$ for the cross-covariance of the centered components, the regularizer is

$$\mathcal{L}_{\text{var}} = \frac{V_{0.5}(Z_{\text{kin}}) + V_{0.5}(Z_{\text{res}})}{2} + \frac{0.1}{d_{\text{kin}} d_{\text{res}}} \left\| \text{Cov}(Z_{\text{kin}}, Z_{\text{res}}) \right\|_F^2. \tag{25}$$

This term discourages constant latent coordinates and linear redundancy between the components. It does not impose statistical independence. The frozen skeleton representation is excluded from the regularizer.

Phase B/C training objectives. The skeleton branch is frozen throughout Phase B/C. Phase B uses

$$\mathcal{L}_{\text{B}} = \mathcal{L}_{\text{rgb}} + \mathcal{L}_{\text{ground}} + 0.5 \mathcal{L}_{\text{align}} + 0.1 \mathcal{L}_{\text{var}}. \tag{26}$$

Phase C adds $\lambda_{\text{swap}} \mathcal{L}_{\text{swap}}$, where $\mathcal{L}_{\text{swap}}$ is the pixel MAE in Eq. (5). The weight is 0.5 for the first 500 updates and 1.0 for the remaining 4,500 updates. With probability 0.2, a Phase C update instead uses only $\mathcal{L}_{\text{rgb}}$. This RGB-only update retains the KL terms already included in $\mathcal{L}_{\text{rgb}}$ and omits the additional grounding, alignment, swap, and variance terms.

Phase B/C optimization. Table 7 lists the optimization settings and selected checkpoints. At each phase transition, model weights are loaded without restoring the previous optimizer, scheduler, or random-number-generator state. Phase C uses bfloat16 precision.

Table 7: **Phase B/C optimization settings.** Update counts are local to each phase (K = 1,000); the selected checkpoints define the final training sequence. Merged entries apply to both phases.

Setting	Phase B	Phase C
Initialization	Phase A, 60K	Phase B, 20K
Configured update limit	40K	10K
Selected checkpoint	20K	5K
Learning rate	5×10^{-5}	2×10^{-5}
Optimizer	AdamW	
Learning-rate schedule	Constant	
Warmup steps	0	
Minimum learning-rate ratio	1.0	
Weight decay	0	
AdamW (β_1, β_2)	$(0.9, 0.999)$	
AdamW ϵ	10^{-8}	
Gradient-norm clipping	0.3	
Hardware	8 × NVIDIA A100 (80 GB)	
Per-GPU batch size	4 frame pairs	
Global batch size	32 frame pairs	

B.4 World-model conditioning and full fine-tuning

Branch availability and zero conditions. Cached latents are standardized using training statistics (Eq. (19)), then masked before projection. One mode from Table 8 is sampled per clip and shared across its 12 transitions. The condition carrier concatenates two masked 32-dimensional latents and their binary presence masks, giving 66 values. A dropped branch uses a fixed zero vector and a zero mask; the supervised video target remains unchanged. The schedule uses the global Stage 2 update index, with Stage 2B starting at offset 5,000 and using the second probability column throughout.

Table 8: **Stage 2 branch-dropout schedule.** One mode is sampled per clip; the last two columns give probabilities at the specified global Stage 2 steps.

Mode	$(m_{\text{kin}}, m_{\text{res}})$	Figure 1(B)	Steps 0–999	Steps $\geq$ 1,000
`full`	$(1, 1)$	keep	0.80	0.50
`drop_both`	$(0, 0)$	null	0.10	0.10
`drop_kin`	$(0, 1)$	partial	0.05	0.05
`drop_res`	$(1, 0)$	partial	0.05	0.35

Optimization. Stage 2A performs 5,000 projector-only updates from the official Cosmos backbone. Stage 2B continues for 20,000 updates of all DiT backbone weights and the conditioning modules. Table 9 summarizes the full-fine-tuning settings.

B.5 Robot-action adaptation

Adapter architecture and targets. The DROID adapter is a shared per-transition MLP with widths $7 \to 256 \to 256 \to 32$ and GELU activations between linear layers, mapping $[B, 12, 7]$ actions to $[B, 12, 32]$ kinematic latents. It receives only the state-derived action proxies in Appendix A.3, standardized using training-set statistics. The targets are standardized RGB-derived kinematic posterior means (Appendix A.4), preserving the coordinate system used during world-model pretraining.

Following Section 3.2, we leave normalization implicit: a_i denotes the standardized adapter input, and $\mu^{\text{rgb}}_{\text{kin},i}$ denotes its standardized posterior-mean target. Writing $v_{ni} \in \{0, 1\}$ for the validity of

Table 9: **Stage 2B latent-conditioned world-model training.** The selected checkpoint is at 20K updates after the 5K Stage 2A warmup. The 5K/10K/30K adaptation experiment concerns Stage 3B.

Setting	Value
Backbone	Cosmos-Predict2.5 2B; full DiT fine-tuning
Conditioning modules	Trainable, inherited from Stage 2A
Learning rates	Conditioning: 5×10^{-5}; full DiT: 2.5×10^{-5}
Objective	Video generation; Eq. (10)
Optimizer	AdamW, $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, weight decay 10^{-3}
Schedule	LambdaLinear; 100-step warmup, $f_{\text{start}} = 10^{-6}$, $f_{\max} = 0.99$, $f_{\min} = 0.4$, cycle length 400,000
Gradient clipping	Global norm 0.1
Batch / precision	Global batch 32; bfloat16
Parallelism / training seed	Eight-way FSDP / 0
EMA / checkpoint interval	Enabled / every 1,000 updates
Frozen modules	Latent action teacher, video tokenizer, text encoder

transition i in clip n, the regression criterion in Eq. (11) is

$$\mathcal{L}_{\text{adapt}}(\psi) = \frac{\sum_{n,i} v_{ni} \| A_\psi(a_{ni}) - \text{sg}[\mu^{\text{rgb}}_{\text{kin},ni}] \|_2^2}{32 \sum_{n,i} v_{ni} + \epsilon}. \tag{27}$$

Only the adapter is updated during Stage 3A; the latent action model, normalization statistics, and pretrained world model remain fixed. The learning-rate cycle is configured for 8,000 updates, while the selected adapter checkpoint is at 2,000 updates. This same 2K adapter initializes the Stage 3B adaptation-budget comparisons.

Action-conditioned generation training. Table 10 specifies Stage 3B, using the conditioning and generation objective in Eqs. (12) and (13). Kinematic conditions are predicted from robot actions throughout training.

Table 10: **Stage 3B action-conditioned generation training.** Update counts are local to Stage 3B; the 2K Stage 3A adapter-fitting updates are excluded.

Setting	Value
Initialization	Stage 2B world model at 20K updates and Stage 3A adapter at 2K updates
Training / evaluation	30,000 reference updates; extended budget evaluation at 5K–50K
Video condition	One observed RGB frame; 12 aligned robot-action transitions
Kinematic condition	$A_\psi(a_i) \in \mathbb{R}^{32}$, with $m_{\text{kin}} = 1$
Residual condition	Paired RGB-derived residual or fixed zero, according to Table 11
Generation supervision	The 12 future RGB frames; Eq. (13)
Trainable modules	Full DiT backbone, action adapter A_ψ, and kinematic projection P_{kin}
Frozen components	Residual projection P_{res}, latent action teacher, video tokenizer, text encoder, and normalization statistics
Learning rates	DiT: 2.5×10^{-5}; adapter: 3×10^{-5}; kinematic projection: 10^{-5}
Optimizer	AdamW, $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, weight decay 10^{-3}
Gradient clipping	Global norm 0.1
Batch / precision	Global batch 32; bfloat16
Parallelism	Eight-way FSDP
EMA / checkpoint interval	Enabled / every 1,000 updates
Evaluation residual	$\tilde{z}_{\text{res},i} = 0$, $m_{\text{res}} = 0$ at every budget

Residual curriculum. At local adaptation update s, the paired standardized residual is retained with probability $q_{\text{res}}(s)$ from Table 11 and presence mask one; otherwise the residual and mask are zero. One choice is shared across all 12 transitions of a clip.

Table 11: **Stage 3B residual curriculum.** The probability controls access to the paired residual during training only; kinematic conditions always come from the action adapter.

Local adaptation updates	Residual retention probability $q_{\mathrm{res}}(s)$
$0 \le s < 2{,}000$	0.50
$2{,}000 \le s < 8{,}000$	Linear decay from 0.50 to 0
$8{,}000 \le s < 30{,}000$	0

Action-conditioned inference. At every evaluated checkpoint, generation uses the current RGB frame and adapter-predicted kinematic conditions, with the residual input and presence mask fixed to zero as in Eq. (14). Table 12 gives the DROID inference settings.

Table 12: **DROID inference configuration at 30K updates.** The verified DROID run completed all 400 generations, with no failed or missing outputs.

Setting	Value
Observed RGB frames	1
Prediction horizon	12 frames
Robot-action sequence	12 transitions
Output resolution ($H \times W$)	480×640
Sampler	`FlowUniPCMultistepScheduler`; rectified flow
Solver	Order 2; solver type `bh2`
Sampling schedule	35 steps; shift 5; non-Karras
Guidance	$g = 0.0$ (conditional prediction)
Negative prompt	Disabled (`is_negative_prompt=False`)
Text condition	Empty string (`task_text=""`), encoded online
Checkpoint / weights	Stage 3B step 30,000; regular weights
EMA weights	Available in the checkpoint; not used for these results
Generations per clip	1
DROID noise seeds	20260823–20261222

Guidance is implemented as $f_{\mathrm{cond}} + g(f_{\mathrm{cond}} - f_{\mathrm{uncond}})$, so $g = 0$ retains the conditional prediction without extrapolation. For zero-based manifest position p, the single generation uses noise seed $20260823 + p$. Clips are distributed over eight ranks without padding or duplication.

C Evaluation Protocols and Baseline References

C.1 Action-conditioned video prediction

The DROID evaluation uses 400 clips at $H \times W = 480 \times 640$. Each canonical RGB clip contains $T = 13$ frames in $[0, 1]^{3\times T\times H\times W}$, including the initial conditioning frame and 12 future frames. Video metrics use all 13 frames, whereas motion errors use the 12 future displacements relative to the initial frame. Scores are first computed per clip or per valid trajectory, as specified below, and then aggregated over the DROID evaluation cohort. The cohort contains 400 single-arm trajectories.

RGB reconstruction and PSNR. For clip n, RGB MSE and MAE average squared and absolute reconstruction errors jointly over all channels, frames, and spatial locations. Writing $e_n = \mathrm{mean}_{c,t,h,w}[(\hat{I}_n - I_n)^2]$, the reported PSNR is

$$\mathrm{PSNR} = \frac{1}{N_v} \sum_{n=1}^{N_v} 10 \log_{10} \frac{1}{e_n}, \tag{28}$$

where N_v is the number of valid clips included in video evaluation. PSNR is computed from the complete clip before averaging over clips and is reported in dB. RGB MSE and MAE have units of squared normalized intensity and normalized intensity, respectively.

SSIM and LPIPS. SSIM is computed framewise with `structural_similarity` from `skimage.metrics`, using `data_range=1.0`, `channel_axis=-1`, and the default 7×7 uniform window. LPIPS (Zhang et al., 2018) uses the TorchMetrics AlexNet implementation with `normalize=True` on canonical RGB frames in $[0, 1]$. Both metrics are averaged over the 13 frames of each clip and then over valid clips.

Temporal perceptual change. Using the same AlexNet LPIPS distance D_{LPIPS}, we define

$$\mathrm{tLPIPS}_n = \frac{1}{12} \sum_{t=1}^{12} \left| D_{\mathrm{LPIPS}}(\hat{I}_{n,t}, \hat{I}_{n,t-1}) - D_{\mathrm{LPIPS}}(I_{n,t}, I_{n,t-1}) \right|. \tag{29}$$

The reported value is the mean over valid clips. This metric measures agreement in adjacent-frame perceptual change, rather than end-effector motion, and is distinct from applying LPIPS to frame-difference images.

Trajectory validity and coverage. A trajectory is valid only when the reference positions are valid and inside the image in every evaluated frame, prediction tracking covers every evaluated frame, and neither trajectory contains NaN or infinite coordinates. Motion scores are aggregated over valid arm trajectories; invalid trajectories are not assigned zero error or replaced with other cases. They remain in the fixed DROID coverage denominator of 400. We report valid/total trajectory counts and validity rates alongside motion scores, and report valid-video counts separately from generated-video counts. Because validity can differ across methods, aggregate motion scores can reflect different subsets of the same evaluation cohort.

End-effector displacement error. Let $p_{nt}, \hat{p}_{nt} \in \mathbb{R}^2$ be the reference and predicted end-effector positions for valid arm trajectory n. We compare displacements relative to the respective initial positions, $\Delta p_{nt} = p_{nt} - p_{n0}$ and $\Delta \hat{p}_{nt} = \hat{p}_{nt} - \hat{p}_{n0}$. With N_m valid trajectories, the average displacement error is

$$\mathrm{ADE} = \frac{1}{12 N_m} \sum_{n=1}^{N_m} \sum_{t=1}^{12} \|\Delta \hat{p}_{nt} - \Delta p_{nt}\|_2. \tag{30}$$

ADE is measured in pixels at the canonical resolution; the initial frame sets the origin and is excluded from the error average.

Direction agreement. Direction is determined from the initial-to-final displacement $\Delta p_{n,12}$. If $|\Delta x| \geq |\Delta y|$, the horizontal sign determines left or right; otherwise the vertical sign determines up or down, with increasing image y corresponding to down. Displacements smaller than 0.005 times the image diagonal, or 4 pixels at 480×640, have no direction label. Direction macro-F1 averages the four per-class F1 scores over valid trajectories for which both the reference and prediction have nonstationary direction labels. Consequently, the direction-evaluation subset can be smaller than the set of valid trajectories used for ADE.

Magnitude agreement. Motion magnitude is the initial-to-final displacement length normalized by the image diagonal: $m_n = \|\Delta p_{n,12}\|_2 / 800$, with $\hat{m}_n$ defined analogously. Magnitude MAE is $N_m^{-1} \sum_n |\hat{m}_n - m_n|$ and is dimensionless, rather than an error in pixels. We report Pearson correlation r_{mag} between predicted and reference magnitudes and Spearman correlation ρ_{mag} between their ranks across valid trajectories. An undefined correlation, including insufficient valid trajectories or zero variance, is unavailable rather than zero.

C.2 Baseline conditioning interfaces

Table 13 summarizes the model families and the conditioning interfaces used in our comparison. The method summaries describe the published approaches; the control column identifies the evaluated interface. Text-conditioned models provide visual-fidelity references without receiving the numerical robot-action sequence. The pretrained DreamDojo baseline is independent of the Cosmos initialization used for SkeLAM (Appendix B.1).

Table 13: **Baseline methods and conditioning interfaces.** Each method is summarized by its main modeling approach and control representation. TesserAct refers to the evaluated RGB-LoRA variant.

Model	Control interface	Method summary
TesserAct (Zhen et al., 2025)	Text	Joint RGB, depth, and normal video prediction for 4D scene modeling.
GE-Base (Liao et al., 2025)	Text	Instruction-conditioned video diffusion for robotic interactions.
IRASim (Zhu et al., 2025)	Action embedding	Video diffusion conditioned on an initial image and an action trajectory.
Ctrl-World (Guo et al., 2026)	Action embedding	Frame-level action conditioning with pose-based memory retrieval for consistent rollouts.
Cosmos-Predict2.5 (Ali et al., 2025)	Action embedding	Flow-based video foundation model adapted for action-conditioned world prediction.
Cosmos3 (Agarwal et al., 2026)	Action embedding	Omnimodal world model with joint video and action modeling.
EnerVerse-AC (Jiang et al., 2025)	Action map	Multi-level action conditioning and ray-map encoding for multi-view prediction.
GE-Sim (Liao et al., 2025)	Action map	Extends GE-Base with projected pose maps and temporal motion conditions.
OSCAR (Wu & Gao, 2026)	Skeleton	Uses rendered 2D kinematic skeletons as a shared control representation across embodiments.
DreamDojo (Gao et al., 2026)	Robot-action interface	Learns continuous latent actions from human videos and adapts the world model to robot controls.
CD-LAM (Wei et al., 2026)	Action-to-latent mapping	Debiases latent actions through embodiment-focused reconstruction, contrastive learning, and calibration.
SkeLAM	Action-to-kinematic mapping	Decomposes motion and residual dynamics using skeleton supervision and cross-modal swap reconstruction.

C.3 Linear motion probes

The probes in Table 3 operate on frozen representations. A linear predictor maps a representation z to an end-effector displacement target:

$$\widehat{\Delta p} = Wz + b. \tag{31}$$

We evaluate displacement with MAE and R^2, and motion-magnitude classification with macro-F1. The compared representations include the DreamDojo and CD-LAM latents, the kinematic and residual components from SkeLAM, the concatenation of these two components, and the skeleton-teacher latent. Individual components have 32 coordinates and the concatenation has 64. The skeleton-teacher row uses privileged skeleton input and serves as a reference for the supervision signal. The latent encoders are not updated by the probe.

These probes measure linearly accessible motion information; together with the interventions, they support motion specialization without establishing complete disentanglement.

C.4 Cross-modal latent interventions

The interventions in Figure 4 use deterministic posterior means throughout. Let $k = z^{\mathrm{rgb}}_{\mathrm{kin}}$, $r = z^{\mathrm{rgb}}_{\mathrm{res}}$, and $s = z^{\mathrm{ske}}_{\mathrm{kin}}$. The current RGB frame, target frame, and RGB decoder are held fixed, while the latent input to the decoder is changed as follows:

$$\begin{aligned} &\text{Normal:} && [k; r], \\ &\text{Correct swap:} && [s; r], \\ &\text{Wrong swap:} && [s'; r], \\ &\text{Zero kin.:} && [0; r], \\ &\text{Zero res.:} && [k; 0]. \end{aligned} \tag{32}$$

Here s' is obtained from an unrelated transition in another episode. The ordinary and swapped reconstructions therefore differ only in the kinematic input. For intervention j, the figure reports relative MAE increase and absolute SSIM decrease from the normal reconstruction:

$$\Delta^{(j)}_{\mathrm{MAE}} = 100\frac{\overline{e}_j - \overline{e}_0}{\overline{e}_0}, \qquad \Delta^{(j)}_{\mathrm{SSIM}} = \overline{s}_0 - \overline{s}_j. \tag{33}$$

MAE increases are percentages; the displayed SSIM decreases are scaled by 10^3. The zero-residual intervention acts on the latent-action RGB decoder. It differs from world-model branch dropout, which zeros a standardized latent and sets its presence mask to zero before projection.

D Additional Experimental Analysis

D.1 Effect of world-model adaptation budget

We compare action-conditioned world-model adaptation after 5K, 10K, 15K, and 30K updates. The budget counts Stage 3B updates and excludes latent-action pretraining, world-model pretraining, and the 2K Stage 3A adapter-fitting updates. The final 30K checkpoint corresponds to the reference training budget in Appendix B.5. Inference uses robot-action-derived kinematic conditions, a fixed zero residual, and $m_{\mathrm{res}} = 0$.

Table 14: **DROID: adaptation-budget sweep.** Each row uses 400 videos; motion scores use the listed valid arm trajectories.

Updates	Valid/total	ADE↓	SSIM↑	LPIPS↓	tLPIPS↓	$\mathrm{F1}_{\mathrm{dir}}$ ↑	r_{mag} ↑
5K	325/400	45.25	0.8299	0.1374	0.0241	0.5257	0.4930
10K	326/400	45.37	0.8326	0.1337	0.0232	0.5248	0.4986
15K	327/400	42.48	0.8345	0.1314	0.0228	0.5439	0.5408
30K	326/400	41.15	0.8340	0.1310	0.0222	0.5465	0.5277

Table 15: **AgiBot World: adaptation-budget sweep.** Each row uses 400 videos; motion scores use the listed valid arm trajectories.

Updates	Valid/total	ADE↓	SSIM↑	LPIPS↓	tLPIPS↓	$\mathrm{F1}_{\mathrm{dir}}$ ↑	r_{mag} ↑
5K	756/800	20.79	0.8452	0.1073	0.0185	0.2968	0.1742
10K	754/800	21.29	0.8477	0.1056	0.0182	0.2888	0.1332
15K	755/800	21.14	0.8471	0.1057	0.0182	0.2752	0.1639
30K	755/800	21.10	0.8479	0.1049	0.0181	0.2824	0.1188

On DROID, increasing the budget from 5K to 30K reduces ADE from 45.25 to 41.15 pixels and LPIPS from 0.1374 to 0.1310. LPIPS decreases across the four reported budgets, whereas ADE increases slightly at 10K before improving at 15K and 30K. Magnitude correlation peaks at 15K among these checkpoints, so the motion metrics do not improve uniformly with additional updates.

D.2 Extended DROID results

Tables 16 and 17 separate video fidelity from motion scores and evaluation coverage. TesserAct denotes the RGB-LoRA evaluation variant.

SkeLAM improves LPIPS and tLPIPS over DreamDojo and CD-LAM. OSCAR has lower ADE (40.36 pixels) and tLPIPS (0.0174) than SkeLAM, but substantially higher LPIPS (0.4790 versus 0.1310).

D.3 Extended evaluation on AgiBot World

We additionally evaluate on 400 held-out AgiBot World clips. Tables 18 and 19 report video fidelity and motion scores, respectively.

Table 16: **DROID: RGB and perceptual evaluation.** All rows have 400 generated videos; N_v is the valid-video count. MSE and MAE are computed in normalized RGB space; PSNR is in dB.

Model	N_v	MSE↓	MAE↓	PSNR↑	SSIM↑	LPIPS↓	tLPIPS↓
TesserAct	400	0.0349	0.0928	15.08	0.7400	0.2561	0.0993
GE-Base	400	0.0596	0.1587	12.51	0.4530	0.5864	0.0549
IRASim	400	0.0738	0.1917	11.55	0.4137	0.5993	0.0912
Ctrl-World	400	0.0255	0.0859	16.34	0.5910	0.4029	0.0220
Cosmos-Predict2.5	400	0.0758	0.1873	11.51	0.4209	0.6018	0.0596
Cosmos3	400	0.0270	0.0753	16.98	0.7630	0.2186	0.0570
EnerVerse-AC	400	0.0415	0.1208	14.13	0.5157	0.4139	0.0584
GE-Sim	400	0.0593	0.1578	12.53	0.4550	0.4950	0.0534
OSCAR	400	0.0587	0.1531	12.57	0.4471	0.4788	0.0174
DreamDojo	400	0.0168	0.0532	18.91	0.8057	0.1473	0.0366
CD-LAM	400	0.0181	0.0571	18.34	0.7910	0.1586	0.0352
SkeLAM	400	0.0151	0.0471	19.35	0.8340	0.1310	0.0222

Table 17: **DROID: motion evaluation and coverage.** ADE is in pixels; magnitude MAE is normalized by the 800-pixel image diagonal. Motion metrics are evaluated on valid trajectories; direction F1 additionally excludes stationary directions.

Model	Valid/total	ADE↓	$\mathrm{F1}_{\mathrm{dir}}$ ↑	L_1^{mag} ↓	r_{mag} ↑	ρ_{mag} ↑	Valid (%)
TesserAct	315/400	83.76	0.2950	0.0797	0.0823	0.0431	78.75
GE-Base	330/400	55.30	0.3006	0.0956	0.0504	0.0340	82.50
IRASim	328/400	67.11	0.2616	0.0843	0.1193	0.1318	82.00
Ctrl-World	326/400	42.96	0.5888	0.0511	0.5154	0.4980	81.50
Cosmos-Predict2.5	326/400	60.76	0.2375	0.0839	0.1049	0.0351	81.50
Cosmos3	328/400	67.23	0.3278	0.0725	0.2518	0.2080	82.00
EnerVerse-AC	327/400	59.63	0.2433	0.0982	-0.0050	-0.0448	81.75
GE-Sim	328/400	55.57	0.2799	0.0941	0.0091	0.0346	82.00
OSCAR	325/400	43.74	0.5810	0.0530	0.5029	0.5030	81.25
DreamDojo	325/400	51.11	0.4989	0.0667	0.3335	0.3032	81.25
CD-LAM	325/400	53.90	0.4903	0.0690	0.3179	0.2496	81.25
SkeLAM	326/400	41.15	0.5465	0.0510	0.5277	0.5283	81.50

Table 18: **AgiBot World: RGB and perceptual evaluation.** All rows have 400 generated videos; N_v is the valid-video count. MSE and MAE are computed in normalized RGB space; PSNR is in dB.

Model	N_v	MSE↓	MAE↓	PSNR↑	SSIM↑	LPIPS↓	tLPIPS↓
TesserAct	376	0.0223	0.0684	17.19	0.7781	0.2043	0.0673
GE-Base	400	0.0077	0.0392	22.15	0.7732	0.3004	0.0268
IRASim	400	0.1889	0.3568	7.25	0.1489	0.8636	0.5998
Ctrl-World	376	0.0364	0.1146	14.69	0.5471	0.5313	0.0243
Cosmos-Predict2.5	400	0.0576	0.1627	12.67	0.4610	0.5950	0.0408
Cosmos3	400	0.0151	0.0523	19.74	0.7982	0.1740	0.0315
EnerVerse-AC	376	0.0355	0.1113	14.81	0.5425	0.4484	0.0185
GE-Sim	376	0.0073	0.0383	22.66	0.8459	0.1506	0.0260
OSCAR	376	0.0058	0.0329	23.44	0.8399	0.0966	0.0126
DreamDojo	400	0.0067	0.0341	22.79	0.8330	0.0889	0.0134
CD-LAM	400	0.0092	0.0421	20.97	0.7994	0.1172	0.0113
SkeLAM	400	0.0088	0.0344	21.79	0.8479	0.1049	0.0181

Table 19: **AgiBot World: motion evaluation and coverage.** ADE is in pixels; magnitude MAE is normalized by the 800-pixel image diagonal. The two arms contribute 800 trajectories in total. Motion metrics are evaluated on valid trajectories; direction F1 additionally excludes stationary directions.

Model	Valid/total	ADE↓	$\mathrm{F1}_{\mathrm{dir}}$ ↑	L_1^{mag} ↓	r_{mag} ↑	ρ_{mag} ↑	Valid (%)
TesserAct	684/800	45.74	0.2348	0.0850	-0.0489	0.0548	85.50
GE-Base	754/800	16.71	0.4830	0.0336	0.1224	0.5104	94.25
IRASim	750/800	78.81	0.2628	0.1182	-0.0335	0.0027	93.75
Ctrl-World	707/800	18.52	0.2652	0.0382	0.1127	0.0317	88.38
Cosmos-Predict2.5	754/800	26.61	0.2802	0.0481	0.0491	0.0816	94.25
Cosmos3	752/800	28.61	0.2853	0.0477	0.0972	0.4452	94.00
EnerVerse-AC	709/800	18.27	0.2276	0.0376	0.0442	0.1576	88.63
GE-Sim	707/800	17.22	0.2529	0.0359	0.0331	0.3021	88.38
OSCAR	708/800	17.32	0.5557	0.0309	0.1946	0.6831	88.50
DreamDojo	754/800	19.55	0.5720	0.0338	0.2056	0.6299	94.25
CD-LAM	750/800	22.73	0.5273	0.0394	0.1583	0.5983	93.75
SkeLAM	755/800	21.10	0.2824	0.0366	0.1188	0.5397	94.38

D.4 Component ablation settings and analysis

Table 20 specifies the training changes for the four variants evaluated in Table 2. These experiments modify training, whereas the interventions in Appendix C.4 change decoder inputs at fixed model weights.

Table 20: **Training changes in the component ablations.** Variant names follow Table 2.

Variant	Training change
Without skeleton supervision	Omit skeleton grounding, RGB–skeleton alignment, and cross-modal swap; learn the latent representation from RGB reconstruction.
Without cross-modal swap	Set $\lambda_{\mathrm{swap}} = 0$ while retaining skeleton grounding and latent alignment.
Without residual latent	Use zero residual conditions from Stage 2 world-model pretraining onward, including robot-action adaptation; retain the action-derived kinematic input.
Without action adaptation	Skip supervised adapter regression and randomly initialize the adapter before action-conditioned generation training; train the adapter and world model jointly.

The zero-residual and no-adapter-regression variants use the same robot-action interface and inference settings as the full model (Appendix B.5). Thus, the residual ablation tests its contribution during training, with zero residual used at inference for both models. All four removals increase motion error, while changes in SSIM and LPIPS do not consistently reflect this degradation. In particular, removing residual training conditions slightly improves LPIPS while worsening ADE and magnitude correlation, supporting their contribution to motion conditioning.

D.5 Additional qualitative comparisons

Figures 8 and 9 extend Figure 3 with two additional DROID examples. Each grid compares the recorded observations with SkeLAM, DreamDojo, Cosmos-Predict2.5, OSCAR, and Ctrl-World at four time steps. The examples illustrate differences in end-effector displacement and the preservation of arm geometry and scene appearance over the rollout. They complement the quantitative results in Tables 16 and 17.

E Zero-Shot Human-to-Robot Transfer Protocol

The transfer experiment in Section 4.4 uses human demonstration videos from UniSkill (Kim et al., 2025) as motion prompts and an observed robot frame as the initial scene. The frozen RGB latent-

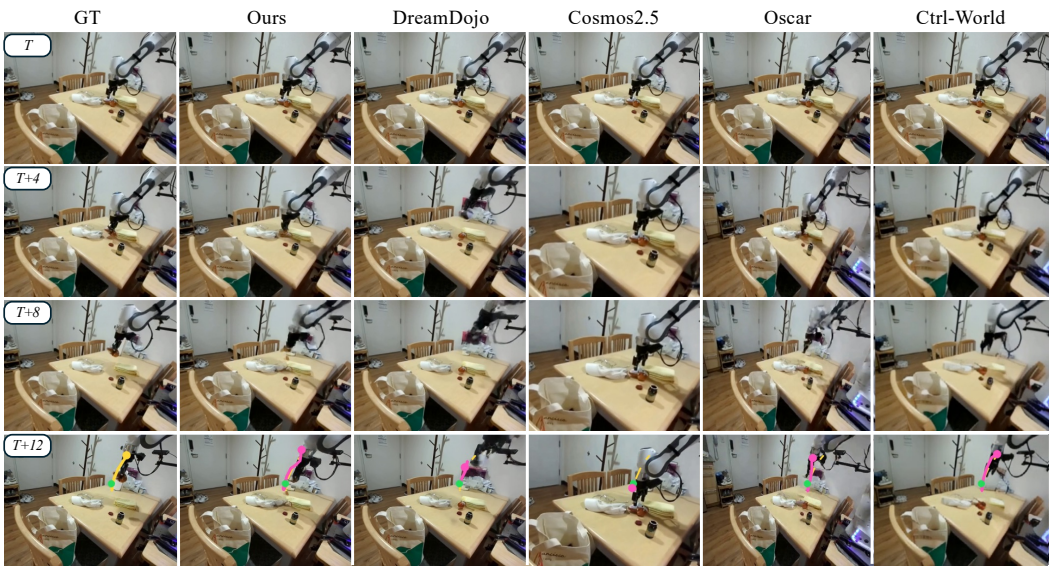


Figure 8: **Additional qualitative comparison in a tabletop scene.** Columns compare ground truth, SkeLAM, DreamDojo, Cosmos-Predict2.5, OSCAR, and Ctrl-World; rows show T, $T+4$, $T+8$, and $T+12$. Colored markers and arrows highlight end-effector motion in the final row.

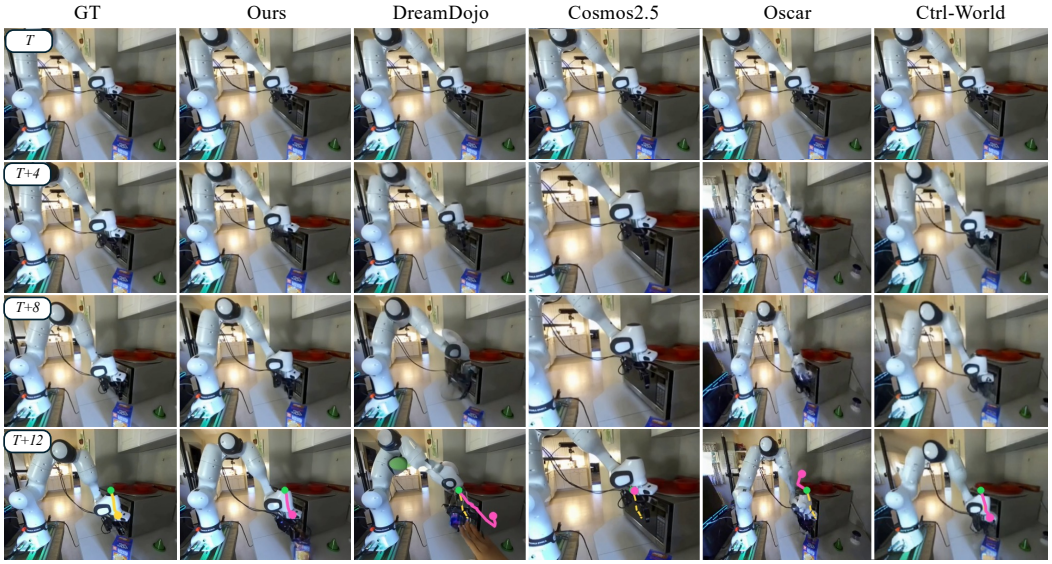


Figure 9: **Additional qualitative comparison in a kitchen scene.** The models and time steps follow the same arrangement as Figure 8. Colored markers and arrows highlight end-effector motion in the final row.

action encoder extracts transition representations from the human video. These latents directly provide motion conditions to the world model; the robot-action MLP is bypassed in this experiment. All parameters remain fixed. The human video is therefore an inference-time motion prompt, rather than an additional fine-tuning dataset for this evaluation.

The initial robot observation provides the scene appearance and embodiment, while the temporal latent sequence provides motion information. The qualitative examples show a downward reach for the towel-pushing demonstration and an upward movement for the tissue-pulling demonstration. The comparison uses a DreamDojo checkpoint that supports latent-action conditioning. This baseline checkpoint is independent of the Cosmos initialization used for SkeLAM.

The comparison concerns correspondence of visible end-effector motion across embodiments. The illustrated examples do not measure successful completion of the human task in the robot scene or establish reliable transfer across arbitrary viewpoints and embodiments.

F Open-Loop Robot Policy Evaluation

F.1 Evaluation unit and outcome labels

The RoboArena experiment evaluates seven policies across 15 sessions, yielding 105 real episodes.

The evaluated policies are `pi0_fast_droid`, `paligemma_fast_droid`, `paligemma_fast_specialist_droid`, `paligemma_vq_droid`, `pi0_droid`, `paligemma_binning_droid`, and `paligemma_diffusion_droid`.

For each episode, the world model receives the same initial observation and recorded action sequence used for the comparison and generates a video rollout. Human annotators inspect each generated rollout and assign a binary label indicating whether the task goal is achieved. These labels are compared with the corresponding real-execution outcomes. Recorded actions are replayed without policy feedback from generated observations, so this experiment measures outcome agreement under *open-loop action replay*.

Let $y_{ps} \in \{0, 1\}$ be the real success label for policy p in session s, and $\hat{y}_{ps}$ the success label assigned by human annotators to its generated rollout. The same task-success criterion is used for both kinds of execution. The 105 real episodes contain nine successes and 96 failures.

F.2 Episode-level agreement

Episode-level metrics pool the 105 label pairs, with task success as the positive class. We report success recall, balanced accuracy (the mean of sensitivity and specificity), and the Matthews correlation coefficient (MCC). Recall and balanced accuracy are percentages, while MCC uses its standard $[-1, 1]$ scale.

F.3 Policy-level agreement

For each policy, the real-execution success rate and the success rate assessed by human annotators in generated rollouts are

$$r_p = \frac{1}{15} \sum_{s=1}^{15} y_{ps}, \qquad \hat{r}_p = \frac{1}{15} \sum_{s=1}^{15} \hat{y}_{ps}. \tag{34}$$

We compute Pearson correlation between the seven pairs $(r_p, \hat{r}_p)$ and Spearman correlation between their ranks, using average ranks for tied success rates. The success-rate MAE is

$$\Delta_{\mathrm{SR}} = \frac{100}{7} \sum_{p=1}^{7} |r_p - \hat{r}_p|, \tag{35}$$

and is reported in percentage points. Policy-level correlations use seven observations, whereas the episode-level metrics use 105 paired outcomes.

The limited number of policies and real successes also bounds the scope of the ranking evidence.

G AI Use Statement

We used generative AI tools for language polishing and readability improvements. All AI-assisted content was reviewed by authors.