CoRe-WAM: Condition-Aligned Distillation and Recurrent Action Refinement for Efficient World-Action Models

Anonymous authors
Paper under double-blind review

Abstract

World-Action Models (WAMs) integrate world prediction with action generation, but iterative sampling increases inference latency during closed-loop control. Reducing this latency is therefore important for timely action updates in response to new observations. In this paper, we focus on the sequential formulation, which first predicts future videos and then generates actions conditioned on them, rather than jointly sampling both modalities. Distribution Matching Distillation (DMD) offers a way to accelerate both stages by training few-step generators with distribution-level supervision. However, a direct branch-wise DMD adaptation that retains ground-truth video conditioning for action matching introduces a mismatch with the student-generated futures used at inference. Moreover, video distribution matching alone provides no explicit action-level feedback for updating the predicted futures. To address these issues, we propose CoRe-WAM, a framework that combines Condition-Aligned Distribution Matching Distillation (CA-DMD) with Recurrent Action Refinement (RAR). CA-DMD shares the same student-generated future across the action sampler and both the teacher and fake action score models. A vector–Jacobian product (VJP) further propagates the action distribution-matching gradient through the action sampler to the video generator, incorporating downstream action feedback alongside video distribution matching. During inference, RAR constructs a proposal from the preceding chunk's terminal action and refines the noise-perturbed proposal under the current video prediction. Experiments on RoboTwin 2.0, LIBERO, and LIBERO-Plus demonstrate competitive manipulation performance with significantly reduced inference latency. On RoboTwin 2.0, our method reduces the mean inference latency per recurrent chunk from 5443.70 ms for the original LingBot-VA teacher to 281.38 ms on an NVIDIA PRO 6000 GPU, yielding a $19.35\times$ speedup.

1 Introduction

World-Action Models (WAMs) integrate predictive world modeling with action generation, providing representations of future scene evolution to guide robot control (Yuan et al., 2026; Chen et al., 2026a). Diffusion and flow models are widely adopted for video and action generation in WAMs, but their iterative sampling requires a large number of function evaluations (NFEs) (Li et al., 2026a; Bi et al., 2026; Ye et al., 2026). This sampling cost significantly increases inference latency during closed-loop control. We focus on the sequential formulation in which a policy first predicts a future video from the interaction context and then generates an action chunk conditioned on that prediction. Since the action sampler depends on the video prediction, both sampling stages must be completed sequentially before each action chunk is executed. Therefore, reducing the NFEs while preserving manipulation performance is important for timely action updates in response to new observations.

Step distillation addresses this sampling cost by training a student to approximate multi-step generation with fewer sampling steps. Distribution matching distillation (DMD) provides distribution-level supervision by matching the student distribution to a target distribution (Yin et al., 2024b;a). For a sequential WAM, a direct adaptation applies separate video and action distribution matching (DM) losses under the original training conditions, which introduces two issues. First, the video DM loss matches student-generated futures, whereas the action DM loss is evaluated under ground-truth future

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.

frames. During inference, the action sampler instead receives student-generated futures. Consequently, matching actions under ground-truth futures does not directly constrain their distribution under the predicted futures encountered during execution. Second, the video DM objective measures discrepancies between student and teacher video distributions, but does not directly evaluate the actions generated from the predicted futures. Video updates based on this objective alone therefore lack explicit feedback on how the predicted futures should change to reduce the mismatch between student and teacher action distributions.

To address these limitations, we propose **CoRe-WAM**, a framework that combines Condition-Aligned Distribution Matching Distillation (**CA-DMD**) with Recurrent Action Refinement (**RAR**) for efficient sequential WAM inference. To resolve the conditioning mismatch, CA-DMD unifies the video condition during action distillation by feeding the identical student-generated future to the action sampler, the teacher score model, and the fake score model. Under this shared video condition, the action DM loss matches the student action distribution to the teacher action distribution. Furthermore, to make video generation sensitive to downstream control requirements, we propagate the action DM gradient through the action sampler back to the predicted video via a Vector–Jacobian Product (VJP), effectively guiding video updates with direct action-level feedback.

While CA-DMD enables few-step generation within individual chunks, closed-loop manipulation inherently requires temporal coherence across chunk boundaries. We introduce RAR, which repeats the preceding chunk's terminal action latent to form a proposal, perturbs the proposal with Gaussian noise, and refines the noisy proposal under the current video prediction. Our approach reuses recent action information as a starting point while allowing updated observations to guide the next action sequence.

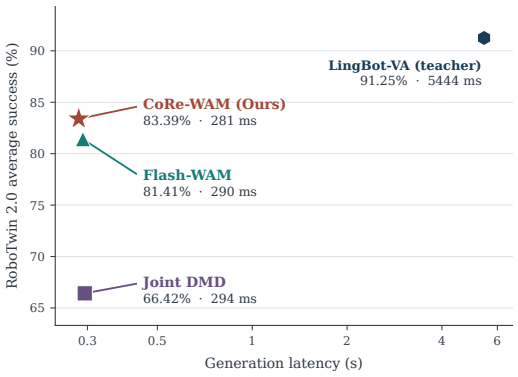

Figure 1: Success and latency on RoboTwin 2.0. Latency is measured locally and averaged over recurrent chunks on an NVIDIA PRO 6000 GPU.

Taking these strategies together, CoRe-WAM requires only one video and one action function evaluation per recurrent chunk, with a single additional action evaluation at episode initialization. We evaluate CoRe-WAM on three widely used datasets, *i.e.*, RoboTwin 2.0, LIBERO, and LIBERO-Plus. The results demonstrate competitive manipulation performance at the reduced sampling budget. On RoboTwin 2.0, our framework reduces the mean inference latency per recurrent chunk from 5443.70 ms for the original LingBot-VA teacher to 281.38 ms on an NVIDIA PRO 6000 GPU, yielding a $19.35\times$ speedup. Figure 1 summarizes the success–latency comparison. At comparable recurrent-chunk latency, our approach achieves a higher average success rate on RoboTwin 2.0 than Joint DMD and the consistency-distillation baseline Flash-WAM (Akbari et al., 2026).

Our contributions are summarized as follows:

- We propose CoRe-WAM for efficient sequential WAM inference. We evaluate it on RoboTwin 2.0, LIBERO, and LIBERO-Plus. The results demonstrate improved manipulation success over Joint DMD under a low sampling budget and substantially reduced inference latency relative to the original teacher.

- We develop CA-DMD to match student and teacher action distributions under student-generated futures, aligning video conditions between distillation and inference. A VJP propagates action DM feedback through the action sampler to guide video updates, which applies explicit action-level feedback alongside video distribution matching.

- We introduce RAR, a training-free warm-start strategy that reuses recent actions for one-evaluation refinement under the current video prediction.

2 Background and Related Work

2.1 World-Action Models

Video prediction connects visual dynamics with robot control through video planning, interactive simulation, and video–action pretraining (Du et al., 2023; Yang et al., 2024; Wu et al., 2024; Cheang et al., 2024). World-action models such as LingBot-VA, Motus, and DreamZero integrate future-video and action generation, allowing predictive representations learned from video to inform manipulation (Li et al., 2026a; Bi et al., 2026; Ye et al., 2026). Related advances in world modeling explore scalable pretraining and interactive simulation (Li et al., 2026b; Wu et al., 2026b; Huang et al., 2026; Rui et al., 2026; Wang et al., 2026a). To reduce the cost of predictive control, architectural approaches remove explicit future generation, use compact predictive features, or decouple imagination from control (Yuan et al., 2026; Chen et al., 2026a; Zhang et al., 2026; Wang et al., 2026b). Step distillation offers a complementary approach that retains the generative pipeline. Flash-WAM follows this approach through modality-specific consistency functions for the different video and action noise regimes (Akbari et al., 2026). Within the same sequential formulation, we instead study distribution matching for the coupled samplers.

In the sequential WAM considered here, both generative branches are trained with linear flow matching (Lipman et al., 2023). Let x_0 denote a clean sample and κ its conditioning input. At noise level $\sigma \in [0, 1]$, the interpolated state is $x_\sigma = (1-\sigma)x_0 + \sigma\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$. Given sample–condition pairs from the training distribution p_{data}, the velocity model u_θ is trained with

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{(x_0,\kappa)\sim p_{\text{data}},\,\sigma,\,\epsilon}\big\|u_\theta(x_\sigma, \sigma, \kappa) - (\epsilon - x_0)\big\|_2^2. \tag{1}$$

The corresponding clean estimate is $\hat{x}_0^\theta = x_\sigma - \sigma u_\theta(x_\sigma, \sigma, \kappa)$. Under this convention, $\sigma = 1$ corresponds to Gaussian noise and $\sigma = 0$ to clean data. Video and action variables below denote the corresponding model latents.

Let c denote the available observation history, action history, and language instruction, and let v_0 and a_0 denote a clean future video and a clean action sequence. A sequential WAM first predicts a future video conditioned on c, then generates an action sequence conditioned on both that future and c, giving the factorization

$$q_\theta(v_0, a_0 \mid c) = q_\theta(v_0 \mid c)q_\theta(a_0 \mid v_0, c). \tag{2}$$

The video and action samplers, denoted by V_θ and A_θ, generate $\tilde{v}_0 = V_\theta(\epsilon_v, c)$ and $\tilde{a}_0 = A_\theta(\epsilon_a; \tilde{v}_0, c)$ from independent Gaussian initializations ϵ_v and ϵ_a.

2.2 Distribution Matching Distillation

Progressive distillation compresses teacher sampling steps, while consistency models learn mappings that agree along a sampling trajectory (Salimans & Ho, 2022; Song et al., 2023; Luo et al., 2023a; Kim et al., 2024). Diff-Instruct and DMD formulate distribution-level objectives for few-step generation (Luo et al., 2023b; Yin et al., 2024b). We adopt the DMD formulation to describe the video and action objectives.

A student sampler G_θ generates $\tilde{x}_0 = G_\theta(\epsilon, \kappa)$ with distribution $q_\theta(x_0 \mid \kappa)$, while a frozen teacher represents the target distribution $p_T(x_0 \mid \kappa)$. Perturbing the sample gives $\tilde{x}_\sigma = (1-\sigma)\tilde{x}_0 + \sigma\epsilon'$, where ϵ' is an independent standard Gaussian draw. DMD seeks to reduce the reverse Kullback–Leibler (KL) divergence between the resulting noised distributions $q_{\theta,\sigma}$ and $p_{T,\sigma}$ over a range of noise levels. The teacher score estimates $s_T(x, \sigma, \kappa) \approx \nabla_x \log p_{T,\sigma}(x \mid \kappa)$, while an online fake score model estimates $s_F(x, \sigma, \kappa) \approx \nabla_x \log q_{\theta,\sigma}(x \mid \kappa)$. Together, these estimates provide the sample-space DM gradient (Yin et al., 2024b):

$$g_{\text{DM}} = w_\sigma \,\text{sg}[s_F(\tilde{x}_\sigma, \sigma, \kappa) - s_T(\tilde{x}_\sigma, \sigma, \kappa)]\,,$$

where $w_\sigma \geq 0$ absorbs the noise-level weighting and the perturbation Jacobian factor $1 - \sigma$, and sg denotes stop-gradient. The corresponding DM loss is implemented as a stopped-target surrogate (Liu et al., 2026a):

$$\mathcal{L}_{\text{DM}} = \tfrac{1}{2}\mathbb{E}\big\|\tilde{x}_0 - \text{sg}[\tilde{x}_0 - g_{\text{DM}}]\big\|_2^2.$$

Backpropagation through the student sampler gives

$$\nabla_\theta \mathcal{L}_{\mathrm{DM}} = \mathbb{E}_{\kappa,\epsilon,\sigma,\epsilon'}\left[\left(\frac{\partial G_\theta(\epsilon,\kappa)}{\partial \theta}\right)^\top g_{\mathrm{DM}}\right]. \tag{3}$$

The target is recomputed for each update and remains constant during differentiation.

Building on this formulation, DMD2 removes the paired regression requirement, stabilizes score estimation through more frequent fake-model updates, and incorporates adversarial supervision (Yin et al., 2024a). Subsequent variants address diversity loss and unreliable distillation gradients (Wu et al., 2026a; Chen et al., 2026b; Lu et al., 2025; Bai et al., 2026; Wu et al., 2026c). Continuous-Time Distribution Matching extends supervision to randomized continuous training timesteps and velocity-extrapolated latents to reduce integration errors in few-step sampling (Liu et al., 2026b). In robot control, OneDP transfers the conditional action distribution of a multi-step diffusion policy to a one-step generator (Wang et al., 2025). These developments motivate distribution matching for efficient generation, while the sequential WAM formulation additionally requires specifying the generated video conditions under which actions are matched and how the action objective contributes to video distillation.

3 METHOD

We propose CoRe-WAM, a framework that combines CA-DMD for distilling the video and action samplers with RAR for recurrent execution. During training, condition alignment evaluates action distribution matching under student-generated futures to supervise the action sampler. A vector–Jacobian product (VJP) propagates the action DM gradient through the action sampler to the generated video. We combine this action-guided gradient with the video DM gradient to train the video sampler. During execution, RAR initializes each subsequent chunk from the preceding terminal action and refines it under the current video prediction. Figure 2 presents the training and inference procedures.

3.1 CONDITION-ALIGNED DISTRIBUTION MATCHING DISTILLATION

Condition-aligned action matching. For sequential WAMs, a direct branch-wise DMD adaptation evaluates the video DM loss on student-generated futures and the action DM loss under the corresponding ground-truth future frames in the training data. During inference, the action sampler instead receives student-generated futures, creating a conditioning mismatch.

Let $p_T(v_0, a_0 \mid c)$ denote the joint teacher distribution. The factorization in Eq. (2) gives

$$\begin{aligned} &D_{\mathrm{KL}}(q_\theta(v_0, a_0 \mid c) \,\|\, p_T(v_0, a_0 \mid c)) \\ &\quad = D_{\mathrm{KL}}(q_\theta(v_0 \mid c) \,\|\, p_T(v_0 \mid c)) + \mathbb{E}_{v_0 \sim q_\theta(\cdot|c)}[D_{\mathrm{KL}}(q_\theta(a_0 \mid v_0, c) \,\|\, p_T(a_0 \mid v_0, c))]. \end{aligned} \tag{4}$$

The first term matches the video marginals, while the second matches conditional action distributions averaged over student-generated futures. Therefore, we evaluate action matching under a sampled student future $\tilde{v}_0 = V_\theta(\epsilon_v, c)$, shared by the action sampler and both action score models. The resulting action is denoted as $\tilde{a}_0 = A_\theta(\epsilon_a; \tilde{v}_0, c)$.

Let g_v and g_a denote the video and action DM gradients with respect to the generated samples. We compute g_v from a perturbed student future and g_a from the perturbed action $\tilde{a}_\sigma = (1-\sigma)\tilde{a}_0 + \sigma\epsilon'$:

$$g_a = w_\sigma \,\mathrm{sg}[s^a_F(\tilde{a}_\sigma, \sigma; \tilde{v}_0, c) - s^a_T(\tilde{a}_\sigma, \sigma; \tilde{v}_0, c)]. \tag{5}$$

The fake action score model is trained on student actions paired with their generated video conditions, while observation histories remain those provided by the training data.

Action-guided video update. Condition alignment brings action matching to student-generated futures. To make video distillation account for the actions generated from these futures, we propagate the action DM gradient back to the video condition. Let $\ell_a = \frac{1}{2}\|\tilde{a}_0 - \mathrm{sg}[\tilde{a}_0 - g_a]\|_2^2$ denote the per-sample action DM surrogate. Its gradient with respect to the video condition is computed through the action sampler by a VJP:

$$g^a_v = \nabla_{\tilde{v}_0} \ell_a = \left(\frac{\partial A_\theta(\epsilon_a; \tilde{v}_0, c)}{\partial \tilde{v}_0}\right)^\top g_a. \tag{6}$$

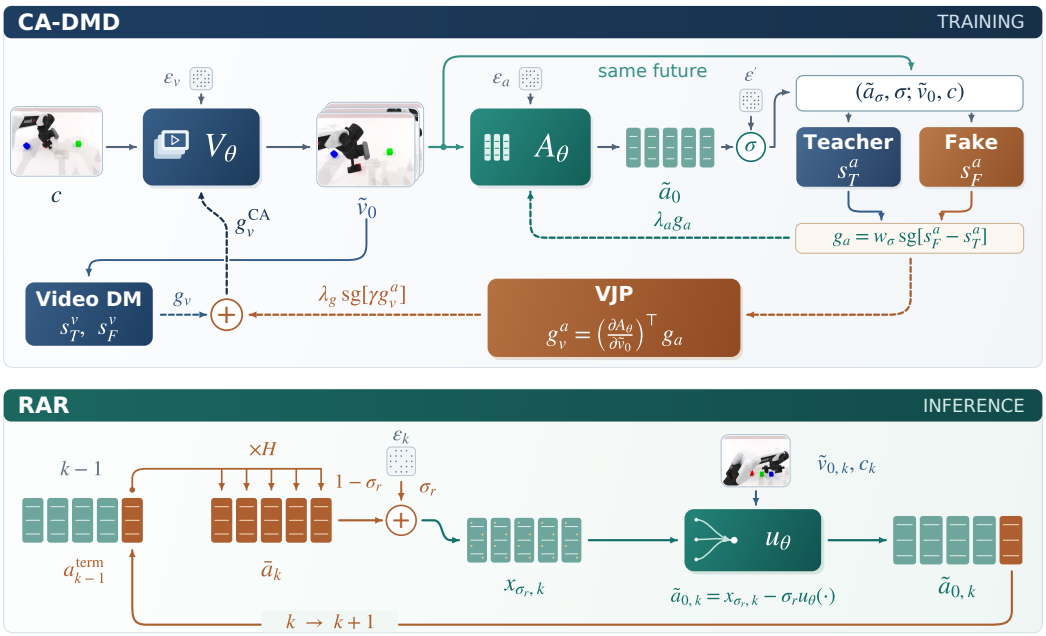

Figure 2: Overview of CoRe-WAM. Top: CA-DMD aligns action matching by conditioning the action sampler and both the frozen teacher and online fake action score models on the same student-generated future. For the video update, a VJP propagates the action DM gradient through the action sampler to the generated future, providing action feedback alongside the video DM gradient. Bottom: RAR repeats the preceding terminal action latent to construct a proposal, perturbs it with noise, and refines it under the current video prediction. The refined terminal action latent is stored to construct the next proposal.

This action-guided video gradient captures how changes in the generated future affect the local action DM surrogate, which complements video distribution matching with downstream action feedback.

We compute the direct action update and the action-guided video update separately so that their strengths can be controlled independently. The magnitude of g_v^a also depends on the sensitivity of the action sampler to its video condition. We therefore calibrate this gradient relative to the video DM gradient before combining them:

$$g_v^{\mathrm{CA}} = g_v + \lambda_g \,\mathrm{sg}[\gamma g_v^a]. \tag{7}$$

For each sample, γ estimates the ratio of the root mean square (RMS) of a reference video DM gradient to that of g_v^a. The weight $\lambda_g \geq 0$ controls the action-guided contribution without rescaling the direct action term.

The combined video gradient g_v^{CA} defines the video term of the student objective. We retain the direct action DM loss driven by g_a under the aligned video condition, together with the paired action flow-matching loss $\mathcal{L}_{\mathrm{FM}}^a$ from the original WAM. With weights $\lambda_a, \lambda_{\mathrm{FM}} \geq 0$, the student objective is

$$\mathcal{L}_{\mathrm{student}} = \tfrac{1}{2}\mathbb{E}\big\|\tilde{v}_0 - \mathrm{sg}[\tilde{v}_0 - g_v^{\mathrm{CA}}]\big\|_2^2 + \tfrac{\lambda_a}{2}\mathbb{E}\big\|\tilde{a}_0 - \mathrm{sg}[\tilde{a}_0 - g_a]\big\|_2^2 + \lambda_{\mathrm{FM}}\mathcal{L}_{\mathrm{FM}}^a. \tag{8}$$

In the direct action term, $\tilde{a}_0 = A_\theta(\epsilon_a; \mathrm{sg}[\tilde{v}_0], c)$ detaches the video condition. This avoids adding the video-feedback path again, since it is incorporated separately through g_v^{CA}. The computed DM gradients, g_v^a, and γ are treated as constants during the student update, avoiding second-order differentiation through the VJP. Appendix C characterizes the implemented update through a local DM surrogate.

3.2 Recurrent Action Refinement

While CA-DMD supports few-step video and action generation within each chunk, closed-loop manipulation also involves temporal dependencies across chunk boundaries. We therefore introduce

RAR to reuse recent action information when initializing the next chunk. Specifically, RAR constructs a proposal by repeating the terminal action of the preceding chunk, perturbs it with noise, and refines it under the current context and video prediction. This provides an initialization informed by recent actions while allowing new observations to guide the generated action sequence. The refinement requires one action evaluation per recurrent chunk and no additional training.

For each subsequent chunk k, we construct the proposal by repeating the preceding chunk's final action latent, a_{k-1}^{term}, with $\bar{a}_{k,h} = a_{k-1}^{\text{term}}$ for $h = 1, \ldots, H$, where H is the latent sequence length. We then perturb the proposal at noise level $\sigma_r \in (0, 1]$:

$$x_{\sigma_r,k} = (1 - \sigma_r)\bar{a}_k + \sigma_r \epsilon_k, \qquad \epsilon_k \sim \mathcal{N}(0, I). \tag{9}$$

Conditioned on the current context c_k and the generated future $\tilde{v}_{0,k}$, the action sampler refines this noisy proposal into a clean action estimate with one evaluation:

$$\begin{aligned} \tilde{a}_{0,k} &= x_{\sigma_r,k} - \sigma_r u_\theta(x_{\sigma_r,k}, \sigma_r; \tilde{v}_{0,k}, c_k) \\ &= \bar{a}_k + \sigma_r[\epsilon_k - \bar{a}_k - u_\theta(x_{\sigma_r,k}, \sigma_r; \tilde{v}_{0,k}, c_k)] \,. \end{aligned} \tag{10}$$

The preceding action influences the initialization, while the current observation and future determine how it is revised. Repeating the terminal action latent therefore does not constrain the output to a constant action sequence. Thus, RAR reuses recent action information across chunks while incorporating updated visual conditions through a single action evaluation.

4 Experiments

In this section, we conduct experiments to evaluate manipulation success, robustness to perturbations, component contributions, and inference latency. The full method requires one video and one action evaluation (1v1a) per recurrent chunk, with one additional action evaluation at episode initialization. We set the refinement noise level to $\sigma_r = 0.5$ in the quantitative evaluations.

4.1 Experimental Setup

Benchmarks. RoboTwin 2.0 contains 50 bimanual manipulation tasks with Clean and Randomized evaluations (Chen et al., 2026c). We report success rates on both splits using 100 evaluation trials per task. LIBERO evaluates language-conditioned manipulation across the Spatial, Object, Goal, and Long suites (Liu et al., 2023), and we report the success rate for each suite and their average. LIBERO-Plus extends LIBERO with more challenging perturbations in camera viewpoint, robot initial state, language, lighting, background, sensor noise, and object layout (Fei et al., 2026). Training relies on the original LIBERO demonstrations without incorporating augmented LIBERO-Plus training data. We evaluate the Long-suite policy on 2,519 cases across these perturbations and report the average success rate.

Models and comparisons. We build on LingBot-VA, retaining its shared video–action backbone and action representation. The student and fake score model are initialized from the frozen teacher. Local baselines are reduced-NFE teacher sampling and vanilla joint video–action DMD (Joint DMD). We also include latent consistency model (LCM) and Flash-WAM success rates (Luo et al., 2023a; Akbari et al., 2026), with Joint LCM on RoboTwin 2.0 and video-only LCM on LIBERO. NFE denotes video/action evaluations per recurrent chunk. Published results reported in Flash-WAM (Akbari et al., 2026) are marked by †, with averages retained as reported.

4.2 Main Results

RoboTwin 2.0. On RoboTwin 2.0, our method achieves 87.80% success on Clean and 78.98% on Randomized, averaging 83.39% as shown in Table 1. Compared with the reduced-NFE teacher, which achieves 54.50% average success, our method improves performance by 28.89 percentage points. It also outperforms Joint DMD by 17.52 and 16.42 percentage points on Clean and Randomized, respectively. We additionally include published Joint LCM and Flash-WAM results as references for consistency-based step distillation (Luo et al., 2023a; Akbari et al., 2026), whereas CA-DMD follows a distribution-matching formulation.

Table 1: RoboTwin 2.0 success rates (%). Bold marks the best reduced-NFE result.

Method	NFE	Clean	Rand.	Avg.
$\pi_0^\dagger$	–	65.92	58.40	62.20
$\pi_{0.5}^\dagger$	–	82.74	76.76	79.80
X-VLA†	–	72.90	72.80	72.80
Motus†	–	88.66	87.02	87.80
LingBot-VA†	25/50	91.64	90.86	91.25
LingBot-VA (reduced NFE)	1/1	56.52	52.48	54.50
LingBot-VA + Joint LCM†	1/1	39.68	32.96	36.32
LingBot-VA + Flash-WAM†	1/1	82.56	**80.26**	81.41
LingBot-VA + Joint DMD	1/1	70.28	62.56	66.42
Ours	1/1	**87.80**	78.98	**83.39**

LIBERO. Across the four LIBERO suites, our method achieves an average success rate of 97.75%, compared with 95.85% for Joint DMD and 91.60% for the reduced-NFE teacher (Table 2). Relative to Joint DMD, our method improves Spatial, Goal, and Long by 3.80, 1.80, and 2.20 percentage points, respectively, while Object is 0.20 points lower. The gains across Spatial, Goal, and Long support the effectiveness of the complete method under different task requirements, including spatial relationships and longer action sequences. On Object, both methods already achieve approximately 99% success, leaving limited room for further improvement. Our method remains within 0.85 percentage points of the original teacher average, indicating a favorable balance between manipulation performance and sampling cost.

Table 2: LIBERO success rates (%). Avg. reports the four-suite mean. Bold marks the best reduced-NFE result.

Method	NFE	Spatial	Object	Goal	Long	Avg.
$\pi_0^\dagger$	–	96.80	98.80	95.80	85.20	94.10
X-VLA†	–	98.20	98.60	97.80	97.60	98.10
LingBot-VA†	20/50	98.50	99.80	98.00	98.30	98.60
LingBot-VA (reduced NFE)	1/1	90.80	97.40	87.00	91.20	91.60
LingBot-VA + Video-only LCM†	1/1	95.00	91.50	95.00	95.40	94.20
LingBot-VA + Flash-WAM†	1/1	96.00	92.60	96.00	95.80	95.10
LingBot-VA + Joint DMD	1/1	93.60	**99.00**	96.20	94.60	95.85
Ours	1/1	**97.40**	98.80	**98.00**	**96.80**	**97.75**

Table 3: LIBERO-Plus Long success rates (%). Overall pools all 2,519 cases. Bold marks the best reduced-NFE result.

Method	Camera	State	Lang.	Light.	Bkgd.	Noise	Layout	Overall
LingBot-VA (teacher)	46.30	89.82	92.69	81.39	69.90	56.35	86.22	73.40
LingBot-VA (reduced NFE)	33.41	79.39	82.51	71.53	62.63	40.76	81.09	62.76
Joint DMD	36.04	83.21	85.38	75.55	60.55	39.20	81.73	64.23
Ours	**48.93**	**90.08**	**95.56**	**79.93**	**73.36**	**54.57**	**87.18**	**74.35**

LIBERO-Plus. For LIBERO-Plus, we focus on the challenging Long suite to evaluate robustness in tasks requiring extended action sequences, where perturbations may affect multiple stages of execution. Our method achieves 74.35% overall success (Table 3), exceeding the original teacher at 73.40% by 0.95 percentage points. Compared with Joint DMD and the reduced-NFE teacher, the overall gains are 10.12 and 11.59 percentage points, respectively. Our method outperforms Joint DMD across all seven perturbation categories and the original teacher in five categories, with lighting and sensor noise being the exceptions. These results support the effectiveness of the complete method when long-horizon manipulation is combined with distribution shifts, showing improved robustness at the reduced sampling budget in the evaluated setting.

4.3 Inference Efficiency

On RoboTwin 2.0, we measure the per chunk inference latency of video and action generation using an NVIDIA PRO 6000 GPU under the same timing scope for all four methods. Table 4 reports mean latency for recurrent chunks, excluding episode initialization. Our method reduces total generation latency from 5443.70 to 281.38 ms, yielding a $19.35\times$ speedup over the original teacher. Both branches contribute to this reduction, with video and action latencies of 143.81 and 137.58 ms, respectively. The VJP is evaluated only during training and adds no inference computation, while RAR retains one action evaluation for each subsequent chunk.

Table 4: Mean generation latency per recurrent chunk on an NVIDIA PRO 6000 GPU. Speedup is relative to the original LingBot-VA teacher.

Method	Video (ms)	Action (ms)	Total (ms)	Speedup
LingBot-VA teacher	1862.56	3581.15	5443.70	$1.00\times$
Joint DMD	151.67	142.38	294.04	$18.51\times$
Flash-WAM	147.75	142.39	290.14	$18.76\times$
Ours	**143.81**	**137.58**	**281.38**	$\mathbf{19.35}\times$

4.4 Component Analysis

In this analysis, we examine condition alignment, the action-guided video update, and RAR through a cumulative ablation on LIBERO-Plus Long (Table 5).

Starting from Joint DMD at 64.23%, conditioning action generation and both action score models on the same student-generated future increases success to 67.49% ($+3.26$ percentage points). This improvement supports aligning action distillation with the video conditions encountered at inference. With condition alignment retained, enabling VJP further increases success to 72.89% ($+5.40$ points), supporting the incorporation of the action DM loss into video distillation alongside video distribution matching.

Table 5: Cumulative component ablation on LIBERO-Plus Long (success rates, %).

Variant	Student future	VJP	RAR	Success (%)
Joint DMD	–	–	–	64.23
+ Condition alignment	✓	–	–	67.49
+ VJP	✓	✓	–	72.89
+ RAR (Ours)	✓	✓	✓	**74.35**

Adding RAR to the same trained checkpoint increases success from 72.89% to 74.35% (+1.46 percentage points). Both variants use two action evaluations for the first chunk and one for each subsequent chunk, while they differ only in whether subsequent chunks use Gaussian initialization or RAR. This additional gain supports recurrent action refinement as a complement to condition-aligned distillation, bringing the total improvement over Joint DMD to 10.12 percentage points.

Effect of video conditioning. To assess the reliability of teacher action predictions under student-generated futures, we compare the same frozen 50-step teacher under ground-truth and student-generated video conditions (Figure 3(a)). Across 50 randomly selected RoboTwin 2.0 RGB block-ranking snippets, we report mean per-snippet XYZ root mean square errors (RMSEs), which measure positional deviations from the paired demonstrations across all three coordinates. The mean RMSE decreases from 19.983 mm under ground-truth futures to 18.839 mm under student-generated futures, a 5.7% reduction. These results indicate that the teacher retains comparable average action accuracy under the evaluated generated conditions.

We further examine whether action supervision improves student predictions by guiding changes to the video condition. With the student parameters and action noise fixed, we adjust the video along $-g_v^a$ and regenerate actions. Across 50 randomly selected 32-action snippets, the mean XYZ RMSE decreases from 17.17 to 12.20 mm (28.9%; Figure 3(b)), indicating the positive effect of action-guided video feedback. The two diagnostics probe different aspects of conditioning: teacher supervision under generated futures and the local response of student actions to changes in those futures. Their trajectory-level measurements complement the closed-loop success results, but do not by themselves establish task completion or robustness beyond the evaluated snippets.

4.5 Qualitative Results

In this section, we examine selected closed-loop rollouts on RoboTwin 2.0. Figure 4 compares Joint DMD and our method on RGB block ranking, bimanual block handover, and opening a drawer to place an object inside. The green circles highlight our completed target configurations: the red–green–blue arrangement, the red block held by the receiving gripper, and the object inside the opened drawer. The red circles highlight the corresponding failures of Joint DMD: incomplete block ordering, an unsuccessful handover, and a closed drawer in the final observation. Together, these selected sequences illustrate differences in object placement and coordination across successive manipulation stages, providing qualitative context for the aggregate success rates reported in our quantitative evaluation. The intermediate frames provide context for these endpoints by showing the preceding object motions and gripper interactions as each selected rollout progresses.

Additional qualitative results on camera-perturbed LIBERO-Plus are provided in Appendix D.

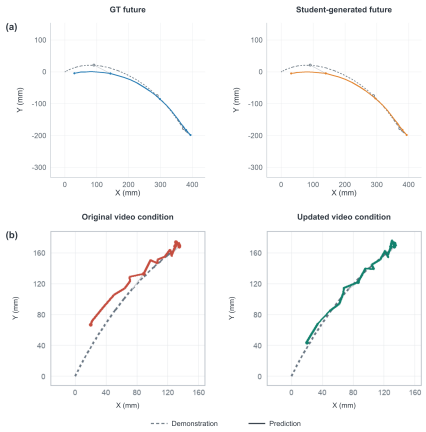

Figure 3: Mean XY trajectories (50 snippets, mm). (a) Teacher: ground-truth vs. student futures. (b) Student: before/after the $-g_v^a$ update. Dashed gray/solid: demonstrations/predictions.

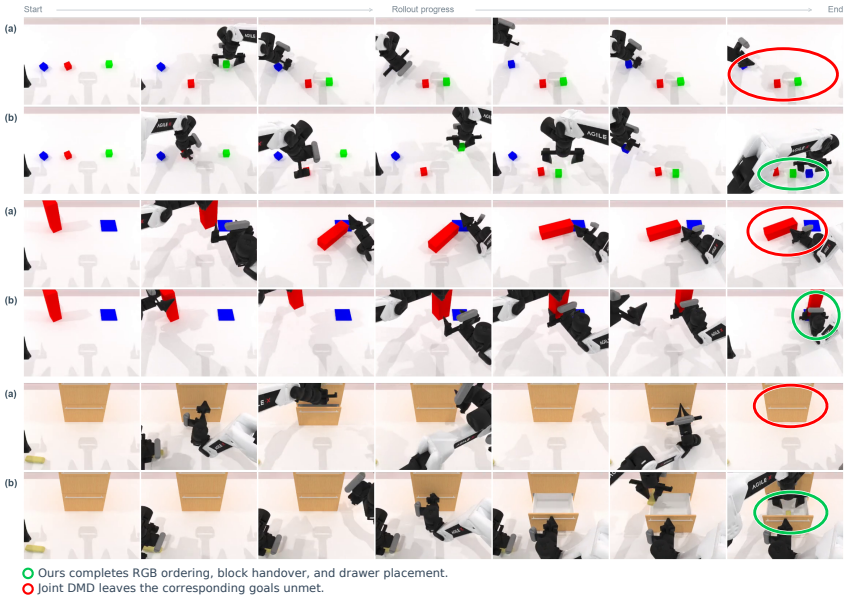

Figure 4: Qualitative comparison on RoboTwin 2.0. From top to bottom, the task groups show RGB block ranking, bimanual block handover, and opening a drawer to place an object inside. Within each group, (a) shows Joint DMD and (b) shows Ours. Red and green circles highlight unmet and completed target configurations, respectively.

5 Conclusion

We propose CoRe-WAM for efficient inference in sequential world-action models. CA-DMD aligns action distillation with student-generated futures and incorporates downstream action feedback into video distillation. RAR complements this training by refining a proposal derived from recent actions under the current video prediction. Across three benchmarks, CoRe-WAM improves upon Joint DMD at 1v1a per recurrent chunk. On RoboTwin 2.0, it achieves a $19.35\times$ speedup in mean recurrent-chunk generation latency over the original teacher.

AI USE STATEMENT

In this work, we used generative AI tools for grammar checking and language polishing. We reviewed all AI-assisted edits to ensure that they preserved the intended meaning and technical content. We take full responsibility for the final content of this work, including all text, claims, and artifacts.

REFERENCES

Arman Akbari, Ci Zhang, Arash Akbari, Lin Zhao, Yixiao Chen, Weiwei Chen, Xuan Zhang, Geng Yuan, and Yanzhi Wang. Flash-WAM: Modality-aware distillation for world action models. *arXiv preprint arXiv:2606.05254*, 2026. URL https://arxiv.org/abs/2606.05254.

Lichen Bai, Zikai Zhou, Shitong Shao, Wenliang Zhong, Shuo Yang, Shuo Chen, Bojun Cheng, and Zeke Xie. Optimizing few-step generation with adaptive matching distillation. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.07345.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A unified latent action world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35101–35113, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Bi_Motus_A_Unified_Latent_Action_World_Model_CVPR_2026_paper.html.

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, Hanbo Zhang, and Minzhao Zhu. GR-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024. URL https://arxiv.org/abs/2410.06158.

Jialei Chen, Kai Wang, Kang Chen, Shuaihang Chen, Feng Gao, Wenhao Tang, Zhiyuan Li, Weilin Liu, Zhuyu Yao, Boxun Li, Yuanbo Xu, and Chao Yu. LaWAM: Latent world action models for efficient dynamics-aware robot policies. In *Conference on Robot Learning*, 2026a. URL https://arxiv.org/abs/2606.15768.

Siyi Chen, Shaowei Liu, Yixuan Jia, Zian Wang, Huan Ling, Qing Qu, and Jun Gao. Data-forcing distillation: Restoring diversity and fidelity in few-step video generation. *arXiv preprint arXiv:2606.18478*, 2026b. URL https://arxiv.org/abs/2606.18478.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan-ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. In *International Conference on Machine Learning*, 2026c. URL https://arxiv.org/abs/2506.18088.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10–11):1684–1704, 2025. doi: 10.1177/02783649241273668. URL https://doi.org/10.1177/02783649241273668.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1d5b9233ad716a43be5c0d3023cb82d0-Abstract-Conference.html.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: A progressive robustness benchmark for visual-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 38574–38583, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Fei_LIBERO-Plus_A_Progressive_Robustness_Benchmark_for_Visual-Language-Action_Models_CVPR_2026_paper.html.

Junchao Huang, Guian Fang, Shengju Qian, Xianghao Kong, Zhuoran Zhao, Wei Huang, Yihua Du, Zixin Zhang, Justin Cui, Yuchao Gu, Yukang Chen, Xinting Hu, Tianyu He, Shaoshuai Shi, Zhuotao Tian, Xin Wang, Mike Zheng Shou, and Li Jiang. SolarWM: Open data and scalable training for long-horizon video world models. *arXiv preprint arXiv:2609.02886*, 2026. URL https://arxiv.org/abs/2609.02886.

Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ODE trajectory of diffusion. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/c204d12afa0175285e5aac65188808b4-Abstract-Conference.html.

Jeongsol Kim, Bryan Sangwoo Kim, and Jong Chul Ye. FlowDPS: Flow-driven posterior sampling for inverse problems. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12328–12337, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Kim_FlowDPS__Flow-Driven_Posterior_Sampling_for_Inverse_Problems_ICCV_2025_paper.html.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Luyao Zhang, Mingrui Yu, Zelin Gao, Nan Xue, Boyu Zhou, Xing Zhu, Mingyu Ding, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. In *Proceedings of Robotics: Science and Systems*, 2026a. doi: 10.15607/RSS.2026.XXII.016. URL https://www.roboticsproceedings.org/rss22/p016.html.

Shalfun Li, Victor Yao, Charles Yang, Truth Qu, Regis Cheng, Ryan Yu, Howard Lu, Newton Von, Vincent Chen, Yohann Tang, Maeve Zhang, Ellie Ma, Gody Li, Starrick Liu, Sage Yang, Lorien Shu, J. W. Gao, Ethan Chen, Colin Ye, Yu Sun, Elise Mon, PS Zhang, Neo Li, Lily Li, James Wang, Ping Yang, Chris Pan, Lucy Liang, Hang Su, Roy Gan, Hao Wang, and Qian Wang. WALL-WM: Carving world action modeling at the event joints. *arXiv preprint arXiv:2606.01955*, 2026b. URL https://arxiv.org/abs/2606.01955.

Shaolong Li, Lichao Sun, and Yongchao Chen. One-step flow policy: Self-distillation for fast visuomotor policies. In *European Conference on Computer Vision*, 2026c. URL https://arxiv.org/abs/2603.12480.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.02747.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8c3c666820ea055a77726d66fc7d447f-Abstract-Datasets_and_Benchmarks.html.

Dongyang Liu, Peng Gao, David Liu, Ruoyi Du, Zhen Li, Qilong Wu, Xin Jin, Sihan Cao, Shifeng Zhang, Steven Hoi, and Hongsheng Li. Decoupled DMD: CFG augmentation as the spear, distribution matching as the shield. In *International Conference on Learning Representations*, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/e425b75bac5742a008d643826428787c-Abstract-Conference.html.

Tao Liu, Hao Yan, Mengting Chen, Taihang Hu, Zhengrong Yue, Zihao Pan, Jinsong Lan, Xiaoyong Zhu, Ming-Ming Cheng, Bo Zheng, and Yaxing Wang. Continuous-time distribution matching for few-step diffusion distillation. *arXiv preprint arXiv:2605.06376*, 2026b. URL https://arxiv.org/abs/2605.06376.

Yanzuo Lu, Yuxi Ren, Xin Xia, Shanchuan Lin, Xing Wang, Xuefeng Xiao, Andy J. Ma, Xiaohua Xie, and Jian-Huang Lai. Adversarial distribution matching for diffusion distillation towards efficient image and video synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16818–16829, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Lu_

Adversarial_Distribution_Matching_for_Diffusion_Distillation_Towards_Efficient_Image_and_ICCV_2025_paper.html.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023a. URL https://arxiv.org/abs/2310.04378.

Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-Instruct: A universal approach for transferring knowledge from pre-trained diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, 2023b. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/f115f619b62833aadc5acb058975b0e6-Abstract-Conference.html.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2108.01073.

Shaohao Rui, Xiaofeng Mao, Zhanyu Zhang, Peijia Lin, Yansong Zhu, Yibo Zhang, Haibin Wan, Zhangrui Zhao, and Weijie Ma. BiWM: Advancing open-source interactive video world models with bidirectional autoregression. *arXiv preprint arXiv:2606.10135*, 2026. URL https://arxiv.org/abs/2606.10135.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2202.00512.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252, 2023. URL https://proceedings.mlr.press/v202/song23a.html.

Jiaqi Wang, Zhuo Zhang, Haining Guan, Tingguang Zhou, Haowen Cui, ChuanYe Wang, Zhongyang Zhu, Yulong Zheng, Xuefeng Chen, Zhen Yang, Tianchen Deng, Feiyang Tan, Hangning Zhou, Bo Dai, Lixia Shen, Xiwu Chen, Xiyang Wang, and Jiajun Zhu. BehaviorWorldGen: Closing the loop between action models and world simulators via controllable behavior-aware structured world generation. *arXiv preprint arXiv:2608.22187*, 2026a. URL https://arxiv.org/abs/2608.22187.

Linhan Wang, Zijian An, Mingyuan Zhang, Chen Dai, Yi Xu, Can Cui, Zichong Yang, Yinlin Chen, Lifeng Zhou, and Chang-Tien Lu. GlanceWAM: Sparse test-time imagination for world-action models. *arXiv preprint arXiv:2608.23927*, 2026b. URL https://arxiv.org/abs/2608.23927.

Zhendong Wang, Max Li, Ajay Mandlekar, Zhenjia Xu, Jiaojiao Fan, Yashraj Narang, Linxi Fan, Yuke Zhu, Yogesh Balaji, Mingyuan Zhou, Ming-Yu Liu, and Yu Zeng. One-step diffusion policy: Fast visuomotor policies via diffusion distillation. In *International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 63399–63416, 2025. URL https://proceedings.mlr.press/v267/wang25ba.html.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/2c37c5bcef24b9541550261dcd63261b-Abstract-Conference.html.

Tianhe Wu, Ruibin Li, Lei Zhang, and Kede Ma. Diversity-preserved distribution matching distillation for fast visual synthesis. In *International Conference on Machine Learning*, 2026a. URL https://arxiv.org/abs/2602.03139.

Xionghao Wu, Yijun Yang, Shiyang Zhou, Haoze Sun, Jianhui Liu, Songsong Yu, Jiyao Zhang, Wenbo Li, Bo Wang, Guoqing Ma, Lin Song, Renjie Liao, Shenghe Zheng, Wei Tang, Xiaojuan Qi, Yanwei

Li, Yuan Zhang, Zhuotao Tian, Haoyang Huang, and Nan Duan. ZimaBlue: Evolving generalizable world action models through scalable video pre-training. *arXiv preprint arXiv:2609.00188*, 2026b. URL https://arxiv.org/abs/2609.00188.

Yuheng Wu, Xiangbo Gao, Tianhao Chen, Xinghao Chen, Qing Yin, Zhengzhong Tu, and Dongman Lee. Delta Forcing: Trust region steering for interactive autoregressive video generation. *arXiv preprint arXiv:2605.14382*, 2026c. URL https://arxiv.org/abs/2605.14382.

Sherry Yang, Yilun Du, Kamyar 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, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi "Jim" Fan, and Joel Jang. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026. URL https://arxiv.org/abs/2602.15922.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. In *Advances in Neural Information Processing Systems*, volume 37, 2024a. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/54dcf25318f9de5a7a01f0a4125c541e-Abstract-Conference.html.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6613–6623, 2024b. URL https://openaccess.thecvf.com/content/CVPR2024/html/Yin_One-step_Diffusion_with_Distribution_Matching_Distillation_CVPR_2024_paper.html.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-WAM: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026. URL https://arxiv.org/abs/2603.16666.

Yuyang Zhang, Wenyao Zhang, Zekun Qi, He Zhang, Haitao Lin, Jingbo Zhang, Yao Mu, Xiaokang Yang, Wenjun Zeng, and Xin Jin. ImageWAM: Do world action models really need video generation, or just image editing? In *Conference on Robot Learning*, 2026. URL https://arxiv.org/abs/2606.19531.

Appendix

CoRe-WAM: Condition-Aligned Distillation and Recurrent Action Refinement for Efficient World-Action Models

A Discussions and Limitations

In this section, we discuss the rationale behind the main design choices, relate them to our empirical findings, and identify a direction for improving recurrent refinement. Since the action sampler receives student-generated futures at inference, CA-DMD evaluates action distribution matching under these same conditions. This choice is consistent with LingBot-VA, which conditions action generation on predicted video representations and incorporates noisy video conditioning during training (Li et al., 2026a). To complement this conditional supervision, we retain video distribution matching and paired action flow matching to supervise video generation and preserve action learning from demonstrations. The paired diagnostic in Figure 3(a) further shows comparable mean teacher action errors under ground-truth and student-generated futures, supporting the practical relevance of teacher predictions under the evaluated generated conditions.

Beyond aligning the conditioning inputs, CA-DMD explicitly incorporates the action DM loss into video distillation through a VJP. Separating the direct action update from video feedback allows their strengths to be controlled independently. Since the feedback magnitude also depends on the sensitivity of the action sampler to its video input, RMS calibration balances it relative to the video DM gradient. Empirically, the controlled VJP ablation supports this action-guided update, improving success by 5.40 percentage points with the remaining training and inference settings unchanged (Table 5). To further characterize this update, Appendix C provides its local objective interpretation.

The distribution-level objective also accommodates multiple valid action sequences without prescribing a fixed teacher trajectory for each noise input (Chi et al., 2025; Yin et al., 2024b; Wang et al., 2025). Complementing this within-chunk learning, RAR incorporates recent actions into the initialization of subsequent chunks and refines them under the current video condition. The refinement noise level controls how much of the preceding action proposal is retained at initialization. Our quantitative evaluations adopt a fixed value of σ_r, whereas adapting it to the current observation and action history would allow this balance to vary with the execution context. This provides a natural direction for further improving recurrent refinement.

B More Details of Recurrent Action Refinement

B.1 Recurrent generation and evaluation budget

At the beginning of each episode, the recurrent state is cleared. The first future video is generated with one evaluation, and the first action chunk is initialized from Gaussian noise and generated with two action evaluations. Its terminal action latent is stored for the next prediction. For each subsequent chunk, RAR repeats the stored latent across the action horizon, perturbs the proposal, and performs one refinement under the current observation and generated future, following Eq. (10). The terminal latent of the refined chunk then replaces the stored state. This procedure follows the denoising principle explored in image editing and action generation (Meng et al., 2022; Li et al., 2026c), without additional training.

For an episode of T chunks, the total video and action evaluation counts are T and $T+1$, respectively. The average budgets are therefore 1 video and $1+1/T$ action evaluations per chunk, or $2+1/T$ in total. The 1v1a notation and recurrent-chunk latency refer to chunks after initialization. The quantitative evaluations set $\sigma_r = 0.5$ at the action scheduler's physical noise level, shared by proposal perturbation and refinement.

Table 6 compares refinement noise levels on RoboTwin 2.0 Clean tasks. The main setting, $\sigma_r = 0.5$, performs best among the three tested values. Lower noise retains more of the preceding action proposal, whereas higher noise permits greater revision under the current condition.

Table 6: RAR noise-level sensitivity on RoboTwin 2.0 Clean (success rates, %).

Noise level σ_r	0.25	0.50	0.75
Success (%)	71.12	**87.80**	84.48

At $\sigma_r = 1$, the proposal coefficient vanishes and Eq. (10) becomes the one-evaluation Gaussian-start update, which defines a Gaussian control for recurrent chunks.

B.2 Denoising interpretation

This section explains the denoising interpretation of the RAR refinement in Eq. (10). Specifically, the update $x_\sigma - \sigma u$ gives a conditional mean estimate of the clean action under an ideal flow-matching model.

For a fixed condition κ and noise level σ, let $x_\sigma = (1-\sigma)a_0 + \sigma\epsilon$, where $a_0 \sim p(a_0 \mid \kappa)$ and $\epsilon \sim \mathcal{N}(0, I)$ are independent. The population minimizer of the squared-error flow-matching objective is $u^*(x_\sigma, \sigma, \kappa) = \mathbb{E}[\epsilon - a_0 \mid x_\sigma, \kappa]$. Consequently,

$$\hat{a}_0^*(x_\sigma, \sigma, \kappa) = x_\sigma - \sigma u^*(x_\sigma, \sigma, \kappa) = \mathbb{E}[a_0 \mid x_\sigma, \kappa]. \tag{11}$$

Thus, the ideal update estimates the clean action from its noisy observation under the given condition (Lipman et al., 2023; Kim et al., 2025).

RAR uses this same update form with the distilled velocity u_θ, the perturbed historical proposal $x_{\sigma_r,k}$, and the current condition $\kappa_k = (\tilde{v}_{0,k}, c_k)$. However, the distilled velocity need not equal u^*, and the historical proposal need not follow the current clean action distribution. The identity therefore provides an interpretation of the refinement operation, rather than a guarantee that RAR computes the exact conditional mean or preserves temporal continuity.

C Gradient Derivation and Local Objective

We explain how the action DM gradient contributes to video learning and how the resulting calibrated directions are implemented through the student objective. We first separate the direct action path from the path through video generation, which explains the gradient detachment used in training. We then construct fixed regression targets and verify that the resulting local objective produces the intended student update.

C.1 Action and Video Gradient Paths

We start from the action DM surrogate to derive the feedback received by the generated video. Before independent weighting and RMS calibration, let $\tilde{v}_0 = V_\theta(\epsilon_v, c)$ and $\tilde{a}_0 = A_\theta(\epsilon_a; \tilde{v}_0, c)$, with the video condition differentiable. At the current student iterate, the action DM gradient g_a and the stopped target in

$$\ell_a = \tfrac{1}{2}\left\|\tilde{a}_0 - \mathrm{sg}[\tilde{a}_0 - g_a]\right\|_2^2$$

are held fixed during differentiation. The residual gives the action gradient, and the chain rule propagates it to the video input:

$$\nabla_{\tilde{a}_0}\ell_a = g_a, \qquad \nabla_{\tilde{v}_0}\ell_a = \left(\frac{\partial A_\theta(\epsilon_a; \tilde{v}_0, c)}{\partial \tilde{v}_0}\right)^\top g_a = g_v^a.$$

This recovers the VJP in Eq. (6). With parameters, action noise, and the stopped target fixed, $-g_v^a$ is a local descent direction for this surrogate in the video condition whenever $g_v^a \neq 0$.

We next trace the action gradient to the model parameters to distinguish direct action learning from feedback through video generation. Substituting $\tilde{v}_0 = V_\theta(\epsilon_v, c)$ gives

$$\nabla_\theta \ell_a = \left(\frac{\partial A_\theta(\epsilon_a; \mathrm{sg}[\tilde{v}_0], c)}{\partial \theta}\right)^\top g_a + \left(\frac{\partial V_\theta(\epsilon_v, c)}{\partial \theta}\right)^\top g_v^a. \tag{12}$$

The first term differentiates the action sampler with its video input fixed. The second term follows the dependence of the generated video on the parameters. This decomposition remains valid when the samplers share parameters, since the two terms correspond to distinct computational paths.

We use this decomposition to explain the detached video condition in the direct action loss. CA-DMD incorporates the video-feedback path through the calibrated direction g_v^{CA}. Keeping the video input differentiable in the direct action term would introduce an additional contribution along that path. We therefore detach this input so that λ_a weights the direct action path, while λ_g controls the separately calibrated video feedback. The following section verifies how these choices are realized by the student objective.

C.2 Local Objective for the Calibrated Update

We now construct a local objective whose gradient at the current iterate matches that of Eq. (8), making the two gradient paths and their independent weighting explicit. The construction turns each computed direction into a fixed regression target by subtracting it from the current output. At iteration n, let $\bar{v}_n = V_{\theta_n}(\epsilon_v, c)$ and $\bar{a}_n = A_{\theta_n}(\epsilon_a; \bar{v}_n, c)$. Compute g_v, g_a, g_v^a, and γ at this iterate and define

$$b_{v,n} = \bar{v}_n - g_v - \lambda_g \gamma g_v^a, \qquad b_{a,n} = \bar{a}_n - g_a. \tag{13}$$

These targets are held fixed during the student update. At θ_n, the video residual is therefore $g_v + \lambda_g \gamma g_v^a$, and the action residual is g_a.

We form a local objective from these targets, keeping the direct action condition fixed as required by the gradient-path decomposition:

$$S_n(\theta) = \tfrac{1}{2}\mathbb{E}\big\|V_\theta(\epsilon_v, c) - b_{v,n}\big\|_2^2 + \tfrac{\lambda_a}{2}\mathbb{E}\big\|A_\theta(\epsilon_a; \bar{v}_n, c) - b_{a,n}\big\|_2^2 + \lambda_{\mathrm{FM}}\mathcal{L}_{\mathrm{FM}}^a(\theta). \tag{14}$$

The first term applies the combined video direction, the second applies the independently weighted direct action direction, and the third retains the paired action flow-matching loss.

We differentiate this objective to check that these terms reproduce the implemented student gradient. Let J_v denote the parameter Jacobian of the video sampler and J_a the parameter Jacobian of the action sampler with its video input fixed, both evaluated at θ_n. Assuming finite moments and valid differentiation under expectations, the residuals defined above give

$$\nabla_\theta S_n(\theta_n) = \mathbb{E}\big[J_v^\top(g_v + \lambda_g \gamma g_v^a) + \lambda_a J_a^\top g_a\big] + \lambda_{\mathrm{FM}}\nabla_\theta \mathcal{L}_{\mathrm{FM}}^a(\theta_n). \tag{15}$$

This is the student gradient specified by Eqs. (7)–(8). The identity uses the computed directions as constants and therefore also holds when the score models are approximate. If the direct action term and the video-feedback term use independent action-noise draws, each direction and fixed target is paired with its corresponding sample.

We finally clarify how this local construction is applied during training. The directions and targets are recomputed at each iteration and remain constant during the corresponding student differentiation. In particular, stopping gradients through the computed VJP and RMS scale avoids second-order differentiation. Thus, S_n characterizes the implemented gradient at the current iterate. This identity characterizes the current-iterate surrogate update, rather than the gradient of the joint KL in Eq. (4).

D Implementation Details and Additional Results

D.1 Training and implementation

Training alternates between updates of the fake score models and the student, while keeping the teacher fixed. Table 7 summarizes the optimization settings, followed by implementation details of the action DM contributions and their gradient handling. Warmup lengths count the updates of each optimizer separately.

To compute the direct action DM loss and the action-guided video update, we use independent action-noise draws under the same generated future. The action DM gradient is recomputed for each draw and paired with the corresponding action sample. Squared errors are summed over latent dimensions, with reduction factors incorporated into the DM gradients before constructing the stopped targets.

Table 7: Training configuration for CA-DMD.

Setting	Implementation
Optimizer	AdamW for student and fake score models, $\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$, weight decay 0
Learning rates	Student: 5×10^{-7}, fake: 10^{-6}
Learning-rate schedule	100 optimizer updates of warmup for each model, then constant
Batch size	One GPU, micro-batch 1, accumulation 8, effective batch 8 per optimizer update
Training duration	2,000 outer steps, 2,000 fake and 400 student updates
Update schedule	Student:fake $= 1 : 5$, fake first at every outer step, student every fifth outer step
Loss coefficients	$\lambda_v = 1$, $\lambda_a = 0.1$, $\lambda_{\text{FM}} = 1$, $\lambda_g = 0.1$
RMS calibration	Per-sample ratio, denominator floor 10^{-12}, cap 100 on γ

For the action-guided video update, we compute g_v^a through a VJP without forming the full Jacobian. We then calibrate its magnitude relative to the video DM gradient g_v. The per-sample scale in Eq. (7) is

$$\gamma = \min\left\{ \frac{\text{RMS}(g_v)}{\max\{\text{RMS}(g_v^a), 10^{-12}\}}, 100 \right\}, \qquad \text{RMS}(z) = \frac{\|z\|_2}{\sqrt{d_z}},$$

where d_z is the number of scalar latent elements in z. Both gradients are evaluated at the same generated future. The coefficient λ_g scales the calibrated feedback in video-latent space before backpropagation through the video sampler, while λ_a separately weights the direct action DM loss.

During the student update, the computed DM directions, the VJP output, and the calibration scale are treated as constants. In particular, detaching the VJP output and γ avoids second-order differentiation, while gradients through the video and action sampler outputs remain available for student learning.

D.2 Additional results

In this section, we provide more qualitative results on LIBERO-Plus.

Figure 5 shows LIBERO-Plus Spatial task 854 under a perturbed camera viewpoint. Our method places the target bowl on the plate at simulator step 159, with the completed placement highlighted by the green circle. In contrast, the red circles show the bowl and plate remaining separate in the displayed Joint DMD rollout, which does not complete the placement within 800 steps. These selected rollouts complement the aggregate success rates.

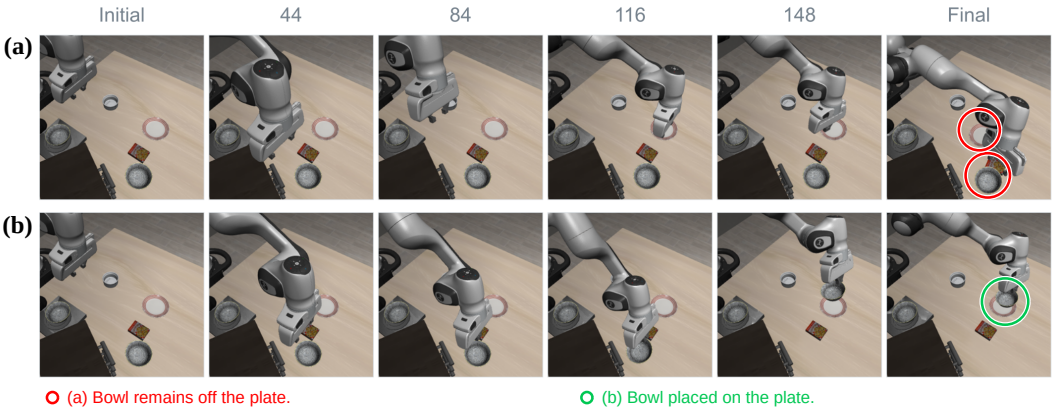

Figure 5: Camera-perturbed LIBERO-Plus Spatial task 854. Rows show (a) Joint DMD and (b) Ours. Columns show the initial observation, simulator steps 44, 84, 116, and 148, and the final recorded observation. Our method completes the bowl placement. Red and green circles highlight unmet and completed target configurations, respectively.

Figure 6 presents camera-perturbed LIBERO-Plus Spatial task 835 from a distant viewpoint. Our method completes the bowl placement, with the green circle highlighting the bowl on the target plate.

For the reduced-NFE teacher and Joint DMD, the red circles show the bowl remaining separate from the plate in the final observations. This example and task 854 supplement the quantitative evaluation on the Long suite.

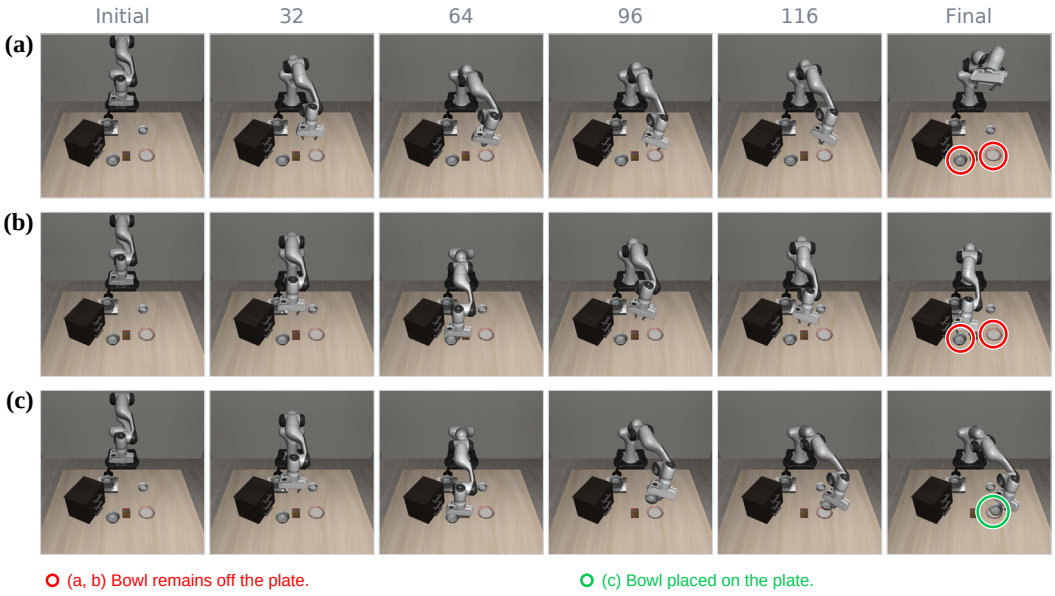

Figure 6: Camera-perturbed LIBERO-Plus Spatial task 835. Rows show (a) the reduced-NFE teacher, (b) Joint DMD, and (c) Ours. Columns show the initial observation, simulator steps 32, 64, 96, and 116, and the final recorded observation. Our method completes the manipulation at step 120, while the displayed baselines fail within 800 steps. Red and green circles highlight unmet and completed target configurations, respectively.