DE-DIFFUSING WAMS: ACTION-SUFFICIENT DIRECT PREDICTION FOR ROBOT CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Cascade world-action models (WAMs) use predicted visual futures to condition robot actions, but inherit costly iterative inference from diffusion-pretrained experts. We ask what a single-step predicted future must preserve to support accurate single-step action prediction. In the evaluated WAM, resampling video noise has little effect on downstream actions, yet naively truncating both streams to one step substantially reduces task success. Video-space supervision only partially closes this gap, and a local sensitivity analysis shows that vision prediction errors have highly unequal consequences for action accuracy. These findings motivate **DirectWAM**, which converts pretrained video and action experts into single-step predictors without stochastic sampling through *action-sufficient distillation*, while retaining the explicit future-to-action cascade. After learning a one-step action model, we train the future predictor by decoding actions from its own predicted futures and backpropagating demonstration action error through the expert, with an optional future-latent reconstruction loss. The resulting objective evaluates the student-generated future through its downstream control consequences rather than relying on representation matching alone. DirectWAM achieves 97.1% average success across four LIBERO suites and 85.4% on RoboTwin 2.0, outperforming Flash-WAM at the same one-evaluation-per-stream budget. This advantage has also extended to three real-world tasks that we designed. Compared with the full iterative teacher, it reduces inference latency by $28.0\times$.

1 INTRODUCTION

Cascade world-action models (WAMs) leverage pretrained video models to predict visual futures, which then condition an inverse dynamics model (IDM) for action generation. Unlike standalone video generation, the predicted future is not the final output but an intermediate representation for robot control. This causal future-to-action relationship allows a robot policy to utilize world prior from video while retaining futures that can be decoded for inspection. However, diffusion-based video and action experts realize this computation through repeated denoising, introducing a substantial inference cost. We study how to retain the pretrained experts and explicit future-to-action interface while replacing their iterative inference procedure with direct prediction.

Recent acceleration methods either modify the future-to-action interface or compress its computation. Fast-WAM retains video co-training but removes explicit future prediction at deployment Yuan et al. (2026); Motus supports action-only inference alongside other joint and conditional modeling modes Bi et al. (2026). Among methods that retain the cascade, Flash-WAM Akbari et al. (2026) uses modality-aware consistency distillation to achieve one-step inference in both the video and action streams. DIDO Lyu et al. (2026) concentrates supervision on control-relevant visual regions to learn single-step future prediction, while retaining iterative action generation. These advances establish that cascade inference can be accelerated. However, a more basic question is largely overlooked in WAM acceleration: what must a single-step future preserve to support accurate downstream action prediction? As the future serves only as an intermediate representation, the reconstruction objective of video diffusion may not naturally align with the control objective of the downstream IDM.

We first disentangle the roles of stochastic sampling and iterative refinement, using LingBot-VA as our experimental instantiation. Varying only the video noise produces limited changes in downstream actions, and fixing it throughout an episode barely affects closed-loop success. Stochasticity

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.

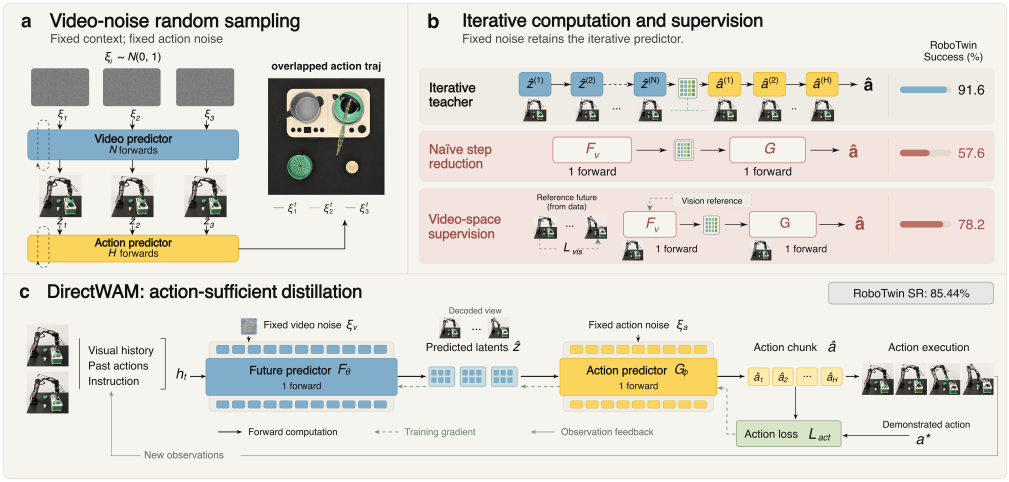


Figure 1: **Overview of DirectWAM.** (a) Video randomness produces similar downstream actions. (b) Naive step truncation and visual supervision induce a noticeable performance gap. (c) DirectWAM is a deterministic one-step cascade WAM trained through action-sufficient distillation.

can therefore be removed, yielding a deterministic future predictor that nonetheless still relies on the teacher's multi-step computation. Removing this iteration is non-trivial: naively reducing the steps degrades control, and learning a one-step predictor with video-space supervision recovers only part of the gap. Therefore, these results motivate learning the direct future predictor through the action model that consumes its predictions, rather than relying on video-space supervision alone.

To understand how the remaining prediction errors affect control, we selectively correct components of the future-representation error under a frozen action expert. Correcting only a small, locally action-sensitive component recovers a disproportionate share of the achievable offline action-error reduction. The complementary component still contributes, but the imbalance shows that representation-error magnitude is a poor proxy for downstream importance. This motivates *action sufficiency* as the learning principle for single-step prediction: the predicted future should be optimized according to the action decoded from it, rather than through representation matching alone.

We implement this principle in **DirectWAM** through *action-sufficient distillation*. Starting from diffusion-pretrained video and action experts, we first train a single-step action expert with the video expert fixed. We then train the single-step future predictor end-to-end through this action expert: actions are decoded from the student's own predicted futures, and the demonstration action error is backpropagated into the future predictor. An optional future-latent reconstruction loss anchors predictions to the demonstrated future. Because both objectives supervise final predictions rather than intermediate denoising states, the pretrained experts are converted into a direct cascade without the student having to reproduce the teacher's denoising trajectory.

In experiments, DirectWAM achieves 97.1% average success on LIBERO and 85.4% on RoboTwin 2.0, outperforming cascade baselines at the same compute budget and substantially narrowing the single-step control gap. These gains further carry over to three real-world manipulation tasks. On a single H100, inference latency decreases from 5042 ms for the full iterative teacher to 180 ms for DirectWAM, a 28.0× speedup. Together, these results demonstrate a practical route to retaining explicit future prediction while replacing iterative diffusion inference with direct computation trained for downstream control. Our contributions are threefold:

1. **Separating stochastic sampling from iterative refinement.** We characterize their distinct roles in the evaluated cascade WAM: video-noise resampling has limited downstream effects, whereas naively truncating both streams to one step causes control drops.
2. **Action sufficiency as a learning criterion.** We show that the magnitude of future-prediction error is a poor proxy for its effect on action accuracy, motivating a training objective that supervises the student-generated future through the actions it induces online.
3. **A deterministic single-step future-to-action cascade.** We introduce **DirectWAM**, a one-step de-diffused cascade WAM that preserves explicit future prediction while achieving strong control performance both in simulation and in the real world.

2 Related Work

World-action models and future-conditioned control. Video-based control uses predicted observations as an intermediate representation for action generation (Du et al., 2023). Recent World-Action Models (WAMs) more tightly integrate visual dynamics and action prediction: LingBot-VA through autoregressive video–action modeling Li et al. (2026), DreamZero with a video-pretrained backbone Ye et al. (2026), and Motus within a multi-expert architecture Bi et al. (2026). Building on the cascade WAM interface, we investigate how to learn direct future prediction according to its action consequences, rather than assuming that control requires fully refined video.

Distillation for efficient cascade WAMs. Consistency models and distribution matching distillation provide general routes to reducing diffusion sampling steps Song et al. (2023); Yin et al. (2024). A distinct architectural approach, Fast-WAM, retains video co-training but removes explicit future prediction at deployment Yuan et al. (2026). We instead focus on acceleration that preserves the future-to-action cascade. Flash-WAM addresses the different noise regimes of video and action streams by assigning modality-specific consistency functions, enabling single-step inference in both streams Akbari et al. (2026). DIDO combines distribution matching with interaction-centric representation guidance for one-step video prediction, using explicit interaction representations together with visual feature alignment to preserve manipulation-relevant information Lyu et al. (2026). These methods demonstrate that efficient cascade WAM prediction requires more than applying a generic sampling shortcut. DirectWAM pursues the same goal of retaining useful future prediction, but defines its learning criterion through the existing action model.

Control-oriented predictive representations. Predictive accuracy need not align with downstream decision quality, as studied through objective mismatch Lambert et al. (2020), value-aware model learning Farahmand et al. (2017), and value equivalence Grimm et al. (2020). Recent WAMs address related representation challenges through geometric or semantic supervision: MECo-WAM Zhang et al. (2026) transfers action-relevant 4D geometric priors from a frozen VGGT Wang et al. (2025), while Robust-WAM augments video-generation-based policies with future-semantic alignment Yan et al. (2026). GigaWorld-Policy-0.5 Team et al. (2026) strengthens action–dynamics coupling through action-conditioned future modeling while allowing action-only inference. Our work shares the broader goal of learning predictive information useful for control, but focuses on how a predicted future should be learned when it must condition a single-step action predictor. Specifically, DirectWAM converts diffusion-pretrained experts into a deterministic single-step cascade while retaining their future-representation interface. Rather than introducing a new semantic or geometric target space, it trains the future predictor through demonstration-action error decoded from its own predictions by a learned direct action model, with optional future-latent anchoring. This places the learning criterion at the interface between the student-generated future and its action consumer, while preserving explicit future conditioning with one evaluation per stream.

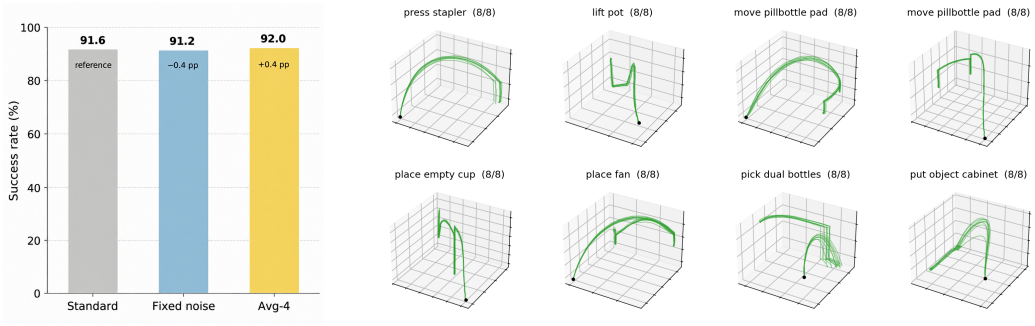
3 Rethinking Future Prediction for Cascade WAMs

In this section, we systematically study the effect of the future prediction in cascade WAMs on downstream robot control. The future is not the endpoint of the model: its effect on control is mediated by the action model and should therefore be viewed from a control perspective.

3.1 Does video-side stochasticity matter for control?

We begin by evaluating whether the action model is sensitive to the stochasticity of future prediction. Diffusion provides the video stream with the ability to sample diverse futures under the same context, but it is unclear whether this variation is propagated into different actions. We therefore ask: *does resampling the imagined future produce meaningfully different control decisions?*

To isolate the effect of stochastic future sampling, we fixed the action-side noise across samples, while varying only the video-diffusion noise. Thus, any variation in the generated actions is induced by changing the sampled future, rather than by resampling the action model itself.



(a) (b)

Figure 2: **Closed-loop effects of video-side stochasticity.** **(a)** RoboTwin success rates of the teacher, fixed video noise, and averaging four future predictions. **(b)** Eight rollouts per scene under different video-noise seeds. Successful trajectories remain closely aligned.

Video-side sampling variation is strongly attenuated by the downstream action model. We first ask how much variation induced by changing only the video-diffusion noise survives after the action mapping. For two samples from the same context, let D_V denote their distance in future-representation space and D_A the corresponding distance between their induced actions. We evaluate variation propagation through three complementary statistics: (i) the Pearson correlation between D_V and D_A across same-context sample pairs, (ii) the mean scale ratio D_A/D_V, and (iii) the fraction of total variance attributable to within-context sampling variation, computed separately in video and action space. Experiment setting details are provided in Appendix A.1.

Table 1: Three metrics measuring the effect of video variation on downstream actions. Same-context video-sampling variation is substantially attenuated by the action model.

Benchmark	Contexts $\times$ samples	$\rho(D_V, D_A)$	Mean D_A/D_V	V $\to$ A var ratio
RoboTwin	237×32	0.69	0.025	$6.6\% \to 0.030\%$
LIBERO	64×16	0.80	0.138	$1.6\% \to 0.25\%$

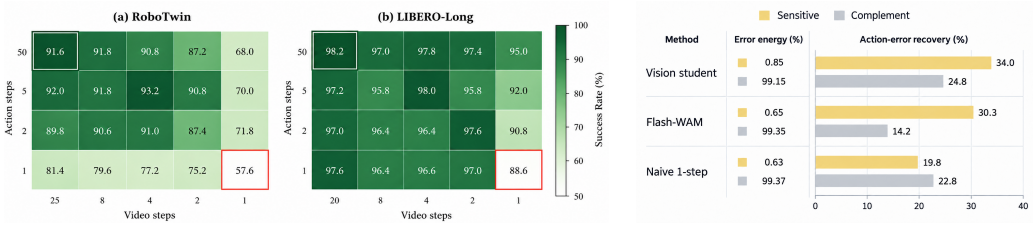
Table 1 shows that video sampling variation is positively correlated with variation on the action side, but its magnitude is substantially attenuated. On RoboTwin, the mean action-space distance is only 2.5% of the corresponding video-space distance, and the relative contribution of sampling variance shrinks from 6.6% in video space to 0.030% in action space. The same trend holds on LIBERO, although with weaker attenuation. Additional clustering and multimodality analyses supporting this conclusion are provided in Appendix A.2 and A.4.

All the above tests support a consistent conclusion: measurable video-sampling variation rarely expands into clearly separated downstream control decisions.

Different video samples rarely induce distinct closed-loop behaviors. Beyond offline metrics, we extend experiments to the closed loop tasks to test the effect of the future randomness in practice.

We intervene directly on video-side stochasticity with two complementary variants. *Fixed noise* reuses a single in-distribution video-noise realization throughout the episode, $\xi_{v,t} = \bar{\xi}_v$ for all t. *Avg-4* averages four independently sampled future representations, $\bar{z}_t = \mathrm{mean}(z_t^{(1)}, \ldots, z_t^{(4)})$, and passes the mean to the action model. The two interventions suppress different aspects of video-side stochasticity: *Fixed noise* suppresses temporal resampling across control chunks, whereas *Avg-4* suppresses sample-specific variation within each chunk by averaging multiple futures. As shown in Fig. 2a, neither intervention causes a systematic degradation in task success.

Limited Closed-Loop Trajectory Diversity. Success rates alone do not reveal whether video-noise sampling induces distinct execution strategies. We therefore randomly sample the teacher eight times from the same initial state for each of 30 contexts selected for the strongest evidence of future branching, varying only the video-diffusion noise. As shown in Fig. 2b, successful rollouts follow closely aligned paths, with local deviations but no clear separation into distinct strategies.



(a) (b)

Figure 3: **Inference-step reduction and action-sensitive error recovery.** **(a)** Success rates (%) across video/action step budgets. Performance degrades as the number of steps decreases. **(b)** The action-sensitive component holds only 0.63–0.85% of the discrepancy energy yet recovers 19.8–34.0% of the offline action-error improvement.

We compute the RMS distance $\sigma(t)$ of the rollouts' end-effector positions from their mean position. We then average this spread over the time steps shared by all rollouts and divide by the episode's motion range R, giving $D_{\text{traj}} = \text{mean}_t[\sigma(t)]/R$. The median normalized spread is 0.020, accounting for a median absolute spread of 0.86 cm compared to a median motion range is 43 cm. Normalization details and outcome-specific summaries appear in Appendix A.3. Video noise can alter outcomes in challenging contexts but induces limited closed-loop trajectory diversity.

3.2 What Should a One-Step Future Preserve?

One-step prediction requires more than truncation or video-space matching. The limited effect of video-noise sampling raises a separate question: how much iterative computation can be removed without retraining? Figure 3a sweeps the video and action inference budgets on RoboTwin and LIBERO-Long. Reducing both streams to two steps retains much of the teacher's performance: success changes from 91.6% to 87.4% on RoboTwin and from 98.2% to 97.6% on LIBERO-Long. The transition to one step per stream is markedly more costly, lowering success to 57.6% and 88.6%, respectively. The streams differ in their sensitivity: RoboTwin degrades when either stream is reduced to one step, whereas LIBERO-Long largely tolerates action-step reduction until the video stream also reaches one step. Thus, although some inference steps can be removed with little loss, the remaining refinement still contributes substantially to control. Limited reliance on stochastic sampling does not imply that a useful one-step predictor is already available in the pretrained model.

Video-space matching remains insufficient. We therefore ask whether the remaining gap can be closed by a learned one-step future predictor. To isolate this factor, all compared future predictors use the same model initialization, training data, optimization budget, and frozen single-step action expert; only the future-learning objective is changed. On RoboTwin, video-space supervision toward teacher-predicted or ground truth future representations achieves 76.0% and 78.2% success, respectively. While substantially stronger than naive one-step truncation, both remain well below the iterative teacher. Thus, improving representation-space fidelity alone does not fully recover the control performance of the original cascade. To understand this remaining gap, we next examine how future-representation errors translate into downstream action errors.

Representation-space fidelity is an incomplete proxy for control fidelity. We examine whether representation-space error is a reliable proxy for downstream action error. Using the local Jacobian of a frozen action model, we decompose the difference between a predicted future z^{B} and the full-teacher future z^{T} into a locally action-sensitive component and its orthogonal complement:

$$\Delta z = z^{\text{T}} - z^{\text{B}}, \qquad \Delta z_{\text{sens}} = P\Delta z, \qquad \Delta z_{\text{comp}} = (I - P)\Delta z. \tag{1}$$

Here, P is an orthogonal projection constructed from 32 sampled action-output directions. We add each component separately to z^{B} and measure the resulting reduction in action error against demonstrations. Recovery is normalized by the improvement obtained from replacing z^{B} with the full teacher prediction z^{T}. Appendix C.3 provides the projection and evaluation details.

Figure 3b shows that, across three variants, small components of the teacher-referenced representation discrepancy can have disproportionately large effects on the decoded action. Thus, represen-

tation distance alone does not indicate which prediction errors are most consequential for action prediction. A representation-matching objective therefore treats errors according to their size in representation space, rather than their effect on the downstream action model. This motivates using downstream action error as a direct learning signal for the one-step future predictor.

3.3 DIRECTWAM: ACTION-SUFFICIENT DISTILLATION

We introduce **action-sufficient distillation**, a model-conversion principle that defines the adequacy of a single-step predicted future of cascade WAMs by the action it enables, rather than by how faithfully it reproduces the teacher's iterative generative trajectory or full future visual representation. Building on this principle, **DirectWAM** converts the diffusion-pretrained video and action experts of a cascade WAM into a de-diffused deterministic single-step future-to-action pipeline; we instantiate it with LingBot-VA in our experiments.

Learning the single-step action model. We first train the action model G_ψ to predict action chunks in a single evaluation. During this stage, the video pathway is frozen and the action model is conditioned on the demonstrated future representation, following the teacher-forced training interface of LingBot-VA. Supervision is applied directly to the final action prediction rather than to intermediate denoising states. This yields a single-step action readout, which is frozen in the subsequent future-learning stage and used to map each student-predicted future to its downstream action.

Action-sufficient future learning. The analysis above suggests that representation matching alone does not indicate which future-prediction errors matter most for downstream control. We therefore change both the training interface and the learning criterion of the cascade. The future predictor is evaluated on its own prediction through the single-step action model that consumes it, and the resulting action error serves as the primary supervision. Given a conditioning context h_t at time t, the future predictor F_θ produces a future representation $F_\theta(h_t)$. The action model G_ψ then predicts an action chunk conditioned on this representation and h_t. Let $\mathcal{D}$ denote the demonstration dataset and $a^\star_{t:t+K-1}$ the target action chunk over a horizon of K steps. We minimize

$$\mathcal{L}_{\mathrm{ASD}}(\theta, \psi) = \mathbb{E}_{\mathcal{D}} \left[\ell_{\mathrm{act}} \left(G_\psi(F_\theta(h_t), h_t), a^\star_{t:t+K-1} \right) \right], \tag{2}$$

where ℓ_{act} is the mean squared error loss. The future predictor receives supervision through how the action model translates its predictions into actions. A predicted future satisfies this objective if it yields the demonstrated action, even when it differs from the reference future. This is the sense in which the future needs to be *action-sufficient*: its adequacy is determined by the action it enables.

This differs from the original LingBot-VA training in two respects. First, LingBot-VA trains the action model under teacher forcing, conditioning it on ground-truth visual transitions, so its action loss does not supervise the consequences of errors made by the video model itself. Second, future prediction is trained as a generative objective in its own right. In DirectWAM, the student-predicted future instead passes through the deployment-time action interface, and downstream action error becomes the primary criterion for learning that future, while representation reconstruction is optional.

Optional Future Anchoring and Loss Composition. We optionally add a future-reconstruction anchor, $\mathcal{L}_{\mathrm{anchor}} = \mathbb{E}_{\mathcal{D}}[\|F_\theta(h_t) - z^\star_{t+1:t+K}\|_2^2]$, where $z^\star_{t+1:t+K}$ is the demonstrated future latent chunk. The overall objective is

$$\mathcal{L}_{\mathrm{train}} = \mathcal{L}_{\mathrm{ASD}} + \lambda_v \mathcal{L}_{\mathrm{anchor}}. \tag{3}$$

Here, λ_v controls the anchor strength; $\lambda_v = 0$ gives purely action-supervised future learning. The reconstruction term acts only as an optional regularizer that preserves proximity to the demonstrated future; downstream action error remains the primary criterion for learning the future representation. Training details and loss curves are provided in Appendix B and C.1.

Direct Prediction WAM targeted at Robot Control In a nutshell, DirectWAM converts diffusion-pretrained video and action experts into a deterministic single-step cascade WAM, retaining the explicit future-to-action interface. The predicted future is learned primarily through the downstream action it induces under the principle of Action Sufficiency, with future reconstruction used only as an optional anchor. In this way, DirectWAM preserves the predictive prior of the pretrained WAM without retaining its stochastic iterative generative inference procedure.

Table 2: **Main simulation results for cascade WAMs.** Success rates (%) on RoboTwin 2.0 and LIBERO; higher is better. The full iterative LingBot teacher is italicized. Steps are video/action steps; Full denotes 20 video steps for LIBERO and 25 for RoboTwin, following the respective LingBot-VA evaluation configurations. RoboTwin results use 50 episodes per task across all 50 tasks. **Bold** denotes the better result between learned single-step methods (Flash-WAM and Direct-WAM). The † denotes LIBERO results quoted from Flash-WAM because no corresponding checkpoint was released; these entries are therefore not treated as a strictly matched reproduction.

Method	Steps (Video / Action)	RoboTwin 2.0			LIBERO				
		Clean	Rand.	Average.	Long	Goal	Object	Spatial	Average.
Lingbot Baseline	*Full / 50*	*91.2*	*90.2*	*90.7*	*98.2*	*93.0*	*97.2*	*98.0*	*96.6*
Naive Step Reduction	2 / 1	76.2	71.5	73.9	97.0	90.8	96.8	96.6	95.3
	1 / 2	67.8	68.0	67.9	90.8	82.8	94.0	94.2	90.5
	1 / 1	57.3	52.4	54.9	88.6	77.8	87.8	91.2	86.4
Flash-WAM	1 / 1	82.1	74.2	78.2	95.8†	96.0†	92.6†	96.0†	95.1†
DirectWAM	1 / 1	**87.0**	**83.7**	**85.4**	**97.2**	**96.6**	**96.2**	**98.5**	**97.1**

4 EXPERIMENTS

4.1 SIMULATION RESULTS

We conduct experiments for DirectWAM and other baselines and methods mainly on two simulation benchmarks: LIBERO Liu et al. (2023) and RoboTwin 2.0 Chen et al. (2025). We use 100 evaluation episodes per task for LIBERO. RoboTwin success rates are evaluated across all 50 tasks, using 50 episodes per task in Table 2 and 10 episodes per task in all other success-rate evaluations.

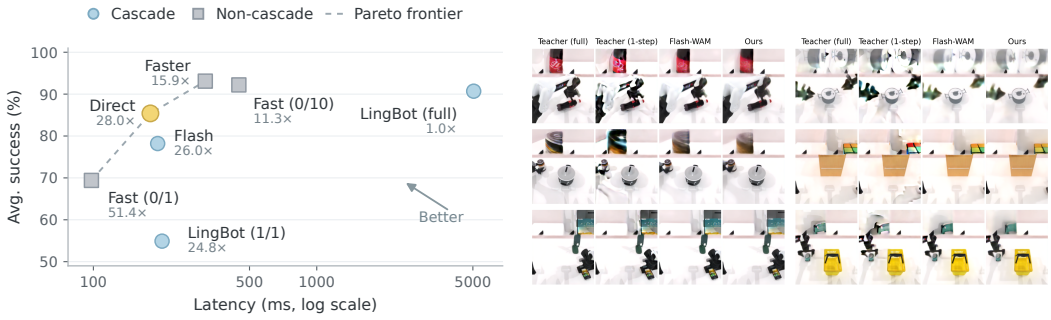
Learning strategy matters for single-step control performance. Table 2 reveals a consistent progression from naive truncation of cascade WAMs to learned future predictors. On RoboTwin, simply reducing the iterative teacher to $1/1$ evaluations causes success to fall to 54.9%. Retaining additional computation in either stream partially recovers performance, reaching 67.9% for $1/2$ and 73.9% for $2/1$, but both remain below learned single-step models. This shows that the remaining iterative computation cannot simply be truncated; the direct mapping itself must be learned.

Among learned $1/1$ models, the training objective further separates performance. DirectWAM reaches 85.4% average success on RoboTwin, compared with 78.2% for Flash-WAM. The gap increases from 4.9 percentage points on the Clean split (87.0% vs. 82.1%) to 9.5 points under randomized evaluation (83.7% vs. 74.2%). A similar numerical trend appears on LIBERO, although the Flash-WAM results there are taken from the original paper rather than from a matched checkpoint. Both methods compress the cascade to the same single-step deployment structure, but differ in how that mapping is learned: Flash-WAM distills the pretrained generative process, whereas DirectWAM evaluates the student-predicted future through the downstream action it induces. The resulting gap supports our central premise that, for direct WAM prediction, a future should be learned according to its usefulness to the action model rather than representation or generative fidelity alone.

DirectWAM remains on the efficiency–performance frontier. Figure 4 places DirectWAM in the broader efficiency–performance landscape on RoboTwin. Within cascade WAMs, DirectWAM substantially improves the trade-off under compressed inference, achieving 85.4% average success with a $28.0\times$ reduction in model-side latency relative to the full iterative teacher.

For broader context, we also include efficient WAMs with different future-conditioning and deployment designs, such as FastWAM and FasterWAMZhao et al. (2026). We omit DIDO from this comparison because no released checkpoint is available for a matched latency evaluation. Despite these architectural differences, DirectWAM remains on the empirical Pareto frontier among the evaluated configurations, occupying an intermediate operating point that balances performance and efficiency.

More broadly, this result suggests that explicit future prediction and iterative generative inference need not be coupled. Retaining a future-to-action interface does not inherently require the compu-



(a) (b)

Figure 4: **Efficiency and qualitative comparison on RoboTwin. (a)** Success–latency trade-off; labels show speedups over full LingBot-VA (5042 ms). **(b)** Six examples of predicted futures from the full teacher, the one-step teacher, Flash-WAM, and DirectWAM.

tational cost of multi-step diffusion. DirectWAM preserves this interface while replacing iterative inference with direct prediction, without moving off the observed efficiency–performance frontier.

Action-supervised futures remain visually decodable. Using downstream action error as the primary supervision raises a natural concern: could the intermediate future drift into an opaque code that is useful only to the action model and sacrifice the explainability of cascade WAMs? Figure 4 provides an empirical check on this possibility by comparing decoded futures from different cascade WAM variants on RoboTwin using predicted frames from different stages of three tasks. Despite being trained primarily through action supervision, DirectWAM produces recognizable future predictions that preserve the overall scene structure, without the obvious visual collapse that would be expected from an unconstrained latent code. Notably, DirectWAM exhibits fewer obvious ghosting artifacts and local color, texture, and shape distortions than the naively truncated one-step teacher. This observation is consistent with the intended roles of the two objectives. The action loss is free to reshape the future representation according to its downstream control consequences, while the optional reconstruction term limits drift from the demonstrated future.

Importantly, the stronger control performance of DirectWAM does not come with an obvious visual degradation relative to Flash-WAM. This suggests that improvement does not come simply from producing visually better futures. Rather, DirectWAM appears to improve how the predicted future supports the downstream action model while preserving a decodable and visually meaningful intermediate representation, retaining the explainability of cascade WAMs. We also provide a quantitative evaluation of predicted-future quality in Appendix C.4, showing that DirectWAM retains strong visual decodability despite being optimized for downstream control.

4.2 Real-World Experiment

Setting. Beyond simulation benchmarks, we evaluate whether DirectWAM preserves effective physical robot control. Experiments use a Galaxea A1 robotic arm equipped with a gripper, a fixed external camera (RealSense D435), and a wrist-mounted camera (RealSense D405). We consider three tabletop manipulation tasks: Grasp Banana, Pouring Water, and Place Teapot, covering object transfer, pouring, and relative-position placement. We use 50 demonstration trajectories per task for training and compare DirectWAM with the LingBot-VA teacher and Flash-WAM over 20 evaluation trials per task. Figure 5 presents representative execution sequences from both camera views.

Results. DirectWAM preserves teacher-level performance in the real world while outperforming the single-step Flash-WAM baseline (Tab. 3). All methods perform near saturation on Grasp Banana, whereas larger differences emerge on the more failure-prone tasks: DirectWAM achieves 11/20 successes on Pouring Water and 12/20 on Place Teapot, compared with 2/20 and 8/20 for Flash-WAM, respectively, while remaining comparable to the iterative teacher. Together with the simulation results, these findings support action-sufficient learning as an effective way to convert the iterative future-to-action cascade into direct prediction without sacrificing physical control.

Table 3: **Real-world task success rates.** Each method is evaluated over 20 trials per task. Results are reported as success rates (%) with successful/total trials. Bold indicates the best result per task.

Method	Grasp Banana	Pouring Water	Place Teapot
LingBot-VA	**95 (19/20)**	45 (9/20)	**60 (12/20)**
DirectWAM	**95 (19/20)**	**55 (11/20)**	**60 (12/20)**
Flash-WAM	90 (18/20)	10 (2/20)	40 (8/20)

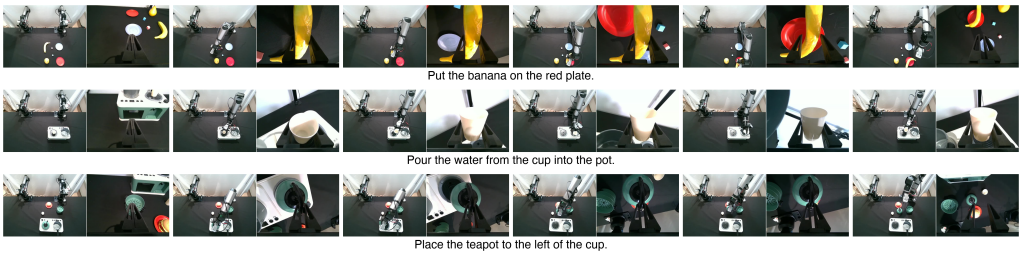


Figure 5: **Real-world task trajectories.** Six keyframes per task are ordered from left to right. Each keyframe pairs the external view (left) with the wrist view (right).

5 Discussion

Separating pretrained capability from inference procedure. DirectWAM highlights a distinction between inheriting a diffusion-pretrained model and retaining its generative inference procedure. A pretrained future-to-action cascade can be converted into deterministic direct prediction without removing the intermediate future. This does not imply that iterative computation is redundant: naive step reduction causes substantial degradation. Rather, our results suggest that the useful capability acquired through diffusion pretraining can be re-expressed as direct computation better suited to downstream control. De-diffusion is therefore a model-conversion strategy that retains the predictive prior while replacing its deployment-time generative process.

Action sufficiency is relative to the downstream mapping. Our results suggest that the quality of a predicted future should be judged as part of the future-to-action interface, rather than by reconstruction fidelity alone. Prediction errors of similar representation-space magnitude can have very different action consequences, and training through the student-generated future directly exposes these consequences to the predictor. This does not imply that visual fidelity is unimportant or that there exists a fixed action-relevant subspace. Instead, for single-step cascade conversion, action sufficiency is defined relative to the direct action mapping that consumes the predicted future: the representation need not preserve every aspect of the reference future, but it must preserve what that downstream mapping requires for control.

6 Conclusion

We study whether a cascade World-Action Model can retain explicit future prediction without retaining the stochastic, iterative diffusion process used to realize it. Our analysis shows that video-side sampling variation has limited downstream effect, while directly truncating iterative inference leaves a substantial control gap. We further find that representation-space prediction error is an incomplete proxy for control relevance, motivating action-sufficient future learning. Based on these observations, we introduce **DirectWAM**, which converts diffusion-pretrained video and action experts into deterministic single-step predictors and trains the future representation through the action induced by its own prediction. DirectWAM preserves the future-to-action cascade while eliminating stochastic sampling and iterative denoising at deployment, achieving strong control performance with substantially reduced inference cost. Overall, our results suggest that the predictive prior learned by diffusion and the generative procedure used at inference need not be inseparable: for robot control, future prediction can be re-learned as a direct, action-sufficient computation.

AI USE STATEMENT

In this work, we used generative AI tools to provide feedback on the presentation of our research methodology and evaluation protocols, check the consistency of reported statistics, and assist in reviewing the interpretation of results. We also used these tools to assist with drafting, rewriting, and language polishing, checking relevant literature and citations, and reviewing manuscript formatting. We did not use generative AI tools to develop or implement the proposed method, generate or process experimental data, conduct experiments, or produce mathematical proofs. The authors reviewed all AI-assisted content and suggestions, verified the final methodological descriptions and numerical claims against the underlying experiments and records, and checked cited claims against the original sources. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We describe the training objectives and learning procedure of DirectWAM in Section 3.3. Section 4 presents the simulation and real-world evaluation settings, including the benchmarks, robotic platform, tasks, and evaluation trial counts. Appendix B provides additional training details, including checkpoint initialization, hardware, optimizer settings, and stage-specific training configurations. Appendix A documents the sampling protocols, statistical measures, and control analyses used to investigate video-side stochasticity. Appendix C details the construction of the action-sensitive projection, the selective-correction procedures, and the associated action-error recovery metric.

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, 2026. URL https://arxiv.org/abs/2606.05254.

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 (CVPR)*, pp. 35101–35113, June 2026.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

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 A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 9156–9172. Curran Associates, Inc., 2023. doi: 10.52202/075280-0403. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1d5b9233ad716a43be5c0d3023cb82d0-Paper-Conference.pdf.

Amir-Massoud Farahmand, Andre Barreto, and Daniel Nikovski. Value-Aware Loss Function for Model-based Reinforcement Learning. In Aarti Singh and Jerry Zhu (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1486–1494. PMLR, 20–22 Apr 2017. URL https://proceedings.mlr.press/v54/farahmand17a.html.

Christopher Grimm, Andre Barreto, Satinder Singh, and David Silver. The value equivalence principle for model-based reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 5541–5552. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/3bb585ea00014b0e3ebe4c6dd165a358-Paper.pdf.

J. A. Hartigan and P. M. Hartigan. The dip test of unimodality. *The Annals of Statistics*, 13(1), March 1985. ISSN 0090-5364. doi: 10.1214/aos/1176346577. URL http://dx.doi.org/10.1214/aos/1176346577.

Nathan Lambert, Brandon Amos, Omry Yadan, and Roberto Calandra. Objective mismatch in model-based reinforcement learning. In Alexandre M. Bayen, Ali Jadbabaie, George Pappas, Pablo A. Parrilo, Benjamin Recht, Claire Tomlin, and Melanie Zeilinger (eds.), *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pp. 761–770. PMLR, 10–11 Jun 2020. URL https://proceedings.mlr.press/v120/lambert20a.html.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Jing Lyu, Shuanghao Bai, Runze Xiao, Zhenyu Liao, Wenxing Tan, Zihan Tang, Ruochuan Shi, Cheng Peng, Yuheng Ji, Yihao Wang, Badong Chen, Pengwei Wang, Zhongyuan Wang, and Xiaoguang Zhao. Dido: Distilling interaction-centric dynamics into one-step denoising for world action models, 2026. URL https://arxiv.org/abs/2609.15570.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/song23a.html.

GigaWorld Team, Angen Ye, Angyuan Ma, Boyuan Wang, Chaojun Ni, Fangzheng Ye, Guan Huang, Guo Li, Guosheng Zhao, Haodong Yan, et al. Gigaworld-policy-0.5: A faster and stronger wam empowered by autoresearch. *arXiv preprint arXiv:2607.13960*, 2026.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Haodong Yan, Junfeng Li, Junjie He, Zhide Zhong, MingMing Yu, Wenxuan Song, Jiaguan Zhu, Yangyang Zheng, Yuqiao Du, Jiadi You, Yingjie Cai, Xu Yan, Guanyi Zhao, Bingbing Liu, and Haoang Li. Robust-wam: Bridging generative pretraining and semantic foresight in world-action models, 2026. URL https://arxiv.org/abs/2608.05903.

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, 2026. URL https://arxiv.org/abs/2602.15922.

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 (CVPR)*, pp. 6613–6623, June 2024.

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.

Jianjun Zhang, Jian Zhu, Taiyi Su, Chong Ma, Zitai Huang, Yi Xu, and Hanli Wang. Learning 4d geometric priors for inference-efficient world action models, 2026. URL https://arxiv.org/abs/2607.05468.

Weiheng Zhao, Haoyi Jiang, Xin Shi, Liu Liu, Fan Huang, Zhizhong Su, Wei Sui, and Xinggang Wang. Faster-wam: Efficient inference-time future conditioning for robust world action models. *arXiv preprint arXiv:2608.04404*, 2026.

A Additional Analysis of Video-Side Stochasticity

This appendix documents the analyses of stochastic future prediction. We first examine how video-sampling variation affects a fixed action model, then describe the clustering and closed-loop interventions. Finally, we consolidate the additional multimodality diagnostic and its controls. These experiments address different questions: whether sampled futures induce different actions, whether resampling benefits execution, and whether detectable visual modes exist. The control conclusions do not require the future distribution to be unimodal.

A.1 Paired Sampling and Offline Variation Propagation

Sampling protocol. For each fixed demonstration context h_i, we generate S future predictions using independent video-noise seeds and evaluate them with a common frozen single-step action model $\mathcal{A}$:

$$z_i^{(s)} \sim p_\theta(\cdot \mid h_i), \qquad a_i^{(s)} = \mathcal{A}(z_i^{(s)}, h_i; \bar{\xi}_{a,i}), \qquad s = 1, \ldots, S. \tag{4}$$

The action-side input is held fixed across video samples within each context. The recorded RoboTwin analysis uses 237 contexts; LIBERO uses 64 contexts with 16 samples per context. The model is the learned single-step head, not the teacher's native multi-step action sampler. This isolates the effect of replacing the future supplied to the same model.

Distance and variance summaries. Let $D_V(i; s, s')$ and $D_A(i; s, s')$ denote the recorded distances between two same-context future predictions and their decoded actions. We report their Pearson correlation and mean pairwise distance ratio. The latter depends on the representation and action scaling and should not be interpreted as a scale-independent fraction of visual variation transmitted to control.

To characterize sampling variation relative to contextual variation, we also report, separately for video and action,

$$R_X = \frac{W_X}{W_X + B_X}, \qquad X \in \{V, A\}, \tag{5}$$

where W_X denotes within-context sampling variance and B_X denotes between-context variance. This is a fraction of total variance, not the ratio W_X/B_X. The report additionally provides a median attenuation summary based on per-context sampling RMS normalized by between-context RMS. This RMS-based summary is distinct from the quotient R_V/R_A.

Table 4: **Offline propagation of video-sampling variation.** All actions within a benchmark are evaluated with the same frozen single-step model. R_V and R_A are fractions of total variance. The final column reports the separately recorded median RMS attenuation.

Benchmark	Contexts × samples	$\rho(D_V, D_A)$	Mean D_A/D_V	R_V	R_A	RMS attenuation
RoboTwin	237×32	0.69	0.025	6.6%	0.030%	20×
LIBERO	64×16	0.80	0.138	1.6%	0.25%	3×

The positive correlations indicate that video-sampling variation is reflected in the decoded actions, rather than being completely ignored. Nevertheless, its relative contribution to action variance is small compared with its contribution to video variance (Table 4). This is an offline observation under the chosen model; its implications for execution are examined through the closed-loop interventions below.

The propagation statistics above are reported for the original stores. The multimodality diagnostic in Appendix A.4 uses a separately audited 237-context RoboTwin subset. These sample counts should not be treated as interchangeable.

A.2 Clustering-Based Action-Separation Test

We complement the continuous summaries with a test of whether representation-based groups induce different mean actions. For each context, we apply K-means with $K = 2$ and ten initializations to the PCA-reduced future representations. This imposes a two-way partition without assuming two

natural modes. For contexts flagged by the multimodality diagnostic, a two-component Gaussian mixture model provides an additional partition.

For groups $\mathcal{C}_{i,1}$ and $\mathcal{C}_{i,2}$, we compute

$$\bar{a}_{i,j} = \frac{1}{|\mathcal{C}_{i,j}|} \sum_{s \in \mathcal{C}_{i,j}} a_i^{(s)}, \qquad S_{\mathrm{act},i} = \frac{\|\bar{a}_{i,1} - \bar{a}_{i,2}\|_2}{\|a_i^\star\|_2}, \tag{6}$$

where $a_i^\star$ is the demonstrated action chunk. The reported separation for the forced partitions is 0.2% of the demonstrated action magnitude. This measures differences between cluster means, not equality of the complete action distributions. Its role is therefore supplementary: under this representation-based partition, the two groups induce little mean-action separation.

A.3 Closed-Loop Sampling Interventions

Protocol and interventions. We evaluate both the iterative teacher and the single-step checkpoint on RoboTwin Clean using 50 tasks and ten episodes per task ($n = 500$ per setting). Scene initialization and instruction settings are shared across variants, and the action-noise schedule is paired. Unlike the common-model offline analysis, these evaluations retain each policy's own action pathway.

Standard sampling draws fresh video noise at each control chunk. *Fixed noise* reuses one in-distribution realization throughout an episode, evaluated separately for seeds 123, 456, and 789. *Avg-4* averages four independently predicted future latents at each chunk before action inference:

$$\xi_{v,t} = \bar{\xi}_v \quad \text{(Fixed noise)}, \qquad \bar{z}_t = \frac{1}{4} \sum_{m=1}^{4} z_t^{(m)} \quad \text{(Avg-4)}. \tag{7}$$

These are input interventions on the same checkpoint, not separately trained models. Fixing a sampled noise tensor is distinct from replacing it with a zero tensor.

Table 5: **Video-sampling interventions on RoboTwin Clean.** Success rates are percentages over 500 episodes per entry (50 tasks, 10 episodes per task). Fixed-noise mean is the arithmetic mean of the three seed-specific success rates.

Model	Standard	Fixed 123	Fixed 456	Fixed 789	Fixed-noise mean	Avg-4
Iterative teacher	91.6	92.4	91.6	89.6	91.2	92.0
Single-step model	87.0	87.4	87.6	89.4	88.1	87.0

The reported paired tests give absolute success-rate changes of at most 2.4 percentage points across the eight interventions, with McNemar $p \geq 0.12$ and bootstrap 95% confidence intervals containing zero. For example, seed 789 changes teacher success by -2.0 points ($[-4.4, 0.4]$) but student success by $+2.4$ points ($[-0.4, 5.2]$). These results show no detected systematic benefit of resampling under the evaluated protocol; they do not establish exact performance equivalence. Individual episode outcomes can change even when aggregate success remains similar.

The recorded standard-sampling baseline uses an unseeded global video RNG, so pairing concerns episodes and action noise rather than identical video-noise realizations. The teacher Avg-4 checkpoint was matched by recorded provenance rather than a stored hash.

Repeated trajectories. We additionally evaluate 30 scene–instruction contexts from 15 tasks, using two scene seeds per task selected for stronger single-chunk multimodality evidence. Each context is executed eight times from the same initial state, with paired action-noise schedules and varying video noise. This targeted set is not a random benchmark sample.

Let $x_s(t) \in \mathbb{R}^3$ be the commanded position of the moving end effector in rollout s. Trajectories are truncated to their common prefix, $T = \min_s T_s$. We compute the mean path and time-averaged RMS spread as

$$\bar{x}(t) = \frac{1}{S} \sum_{s=1}^{S} x_s(t), \qquad d = \frac{1}{T} \sum_{t=1}^{T} \sqrt{\frac{1}{S} \sum_{s=1}^{S} \|x_s(t) - \bar{x}(t)\|_2^2}, \quad S = 8. \tag{8}$$

The normalized spread is

$$R_s = \left\| \max_{1\le t\le T} x_s(t) - \min_{1\le t\le T} x_s(t) \right\|_2, \qquad R = \mathrm{median}_s\, R_s, \qquad D_{\mathrm{traj}} = d/R, \tag{9}$$

where maxima and minima are coordinatewise.

In 25 contexts, all eight rollouts succeed; four contexts have mixed outcomes, and one fails in all repetitions. The median normalized spread is 2% across all 30 contexts and remains 2% within the 25 all-success contexts. For these 25 contexts, the median absolute spread is $0.86\,\mathrm{cm}$ and the median motion range is $43\,\mathrm{cm}$. These are separately aggregated statistics, not a conversion obtained by dividing the two medians. The metric summarizes commanded positions over the shared prefix; it does not measure full action-distribution equivalence or all aspects of execution.

A.4 Multimodality Analysis of Sampled Futures

This diagnostic asks whether repeated future samples exhibit detectable multimodal structure. It supplements, rather than establishes a premise for, the action and closed-loop analyses above.

Data and held-out representation. We use the RoboTwin teacher with 25 video denoising steps and guidance scale 5, generating 32 samples per context. The original store contains 256 contexts from 50 tasks. Nineteen contexts have mismatched history latents and prompts due to an indexing issue; the multimodality analysis therefore uses the remaining 237 contexts from 47 tasks. Each future has shape $48 \times 2 \times 24 \times 20$. Averaging the two latent frames and applying 4×4 spatial average pooling produces a 1,440-dimensional vector per sample.

We split contexts into two disjoint sets, fit a PCA basis to sampling deviations in one set, and project samples in the other; the roles are then exchanged. For context i,

$$y_i^{(s)} = \phi_{-i}(z_i^{(s)}), \tag{10}$$

where ϕ_{-i} is fitted without context i. We retain $d = 8$ principal components, with $d = 16$ as a robustness check. This prevents a tested context from determining its own PCA basis.

Projection-based testing. We evaluate 28 predefined one-dimensional projections per context. In the $d = 8$ setting, these comprise the eight PCA coordinate axes and twenty fixed random unit directions. The directions are shared across real samples and the corresponding controls. For direction u_m, define

$$q_{i,s}^{(m)} = u_m^\top y_i^{(s)}, \qquad H_{i,m}(x) = \frac{1}{S}\sum_{s=1}^{S} \mathbf{1}[q_{i,s}^{(m)} \le x]. \tag{11}$$

Hartigan's dip statistic Hartigan & Hartigan (1985) is

$$D_{i,m} = \inf_{G\in\mathcal{F}_{\mathrm{uni}}} \|H_{i,m} - G\|_\infty, \tag{12}$$

where $\mathcal{F}_{\mathrm{uni}}$ is the family of unimodal CDFs. Let $p_{i,m}$ be the p-value returned by the dip test under its unimodal null. A context is flagged when

$$\min_{m=1,\dots,28} p_{i,m} < 0.05/28, \tag{13}$$

using a Bonferroni correction across projections.

Calibration and positive controls. For the matched-unimodal check, we estimate the mean μ_i and covariance Σ_i in PCA space and draw matched Gaussian samples,

$$\tilde{y}_i^{(s)} \sim \mathcal{N}(\mu_i, \Sigma_i), \qquad s = 1, \dots, 32. \tag{14}$$

The existing calibration report contains 1,280 null sets, generated as five replicates per reference context. The reported detection rates are 0.16% for PCA-8 and 0% for PCA-16.

Positive controls combine 16 samples from each of two different contexts of the same task. We evaluate 190 such two-context mixtures using the same held-out representation and corrected testing procedure. Detection rates are 97.4% for PCA-8 and 99.5% for PCA-16. These controls demonstrate sensitivity to the constructed balanced mixtures, not to every possible form of multimodality.

Table 6: **Multimodality diagnostic and controls.** Real-data entries give detected contexts over the 237-context audited set. Control entries report detection rates.

Sample set	PCA-8	PCA-16
Same-context futures	$0/237$ (0.0%)	$1/237$ (0.4%)
Matched Gaussian controls	0.16%	0.0%
Two-context mixtures	97.4%	99.5%

Interpretation and limitations. We find little evidence of multimodality at the tested representation resolution and sample count. The sole PCA-16 detection is not consistently separated across the reported checks. Non-detection does not establish global unimodality: discarded representation directions may contain modes, 32 samples provide limited coverage of rare modes, and balanced positive controls are easier than highly imbalanced mixtures. Moreover, this diagnostic uses one predicted chunk under teacher-forced history. Its conclusions need not describe the full state distribution reached in closed-loop execution.

B TRAINING DETAILS

Both benchmarks initialize from the released LingBot-VA post-trained checkpoints. The documented two-stage runs use a single node with eight NVIDIA H100 GPUs, fully sharded data parallelism, a per-GPU batch size of one, and AdamW Loshchilov & Hutter (2019) with $\beta = (0.9, 0.95)$ and weight decay 0.1.

Stage 1: single-step action learning. The video expert is frozen while the action expert is trained against demonstrated action chunks. Supervision is applied to the final action prediction. On LIBERO-Long, this stage uses 600 optimization steps and an effective batch size of 16. On RoboTwin, it uses 8,000 steps at learning rate 10^{-5}, taking approximately 3.5 hours.

Stage 2: action-sufficient future learning. The future predictor is trained through the action decoded from its own predicted future, with the action expert frozen during the plotted training phase. LIBERO-Long uses 400 steps, an effective batch size of eight, and no reconstruction anchor ($\lambda_v = 0$); the two stages together take approximately 21 minutes. RoboTwin uses 6,500 steps at learning rate 10^{-5} with $\lambda_v = 0.01$, taking approximately six hours for Stage 2. The training curves and their interpretation are given in Appendix C.1.

C Additional Analysis of Action-Sufficient Learning

We examine the learned future predictor from two complementary perspectives. Training curves describe how action and reconstruction objectives evolve during Stage 2. Selective replacement then tests which components of the difference between the learned predictors account for their downstream action improvement, first through offline action error and then through closed-loop control.

C.1 Stage-2 Training Dynamics

Figure 6 reports the phase in which the action expert is held fixed, and the video expert is updated. We first consider LIBERO-Long, where $\lambda_v = 0$. Despite receiving no reconstruction supervision, the future-prediction pathway learns to reduce the action loss over training. The reduction therefore does not require updating the action expert or explicitly minimizing a video reconstruction objective.

RoboTwin provides a complementary setting with $\lambda_v = 0.01$. Here, action loss decreases substantially while the weighted anchor loss remains approximately stable. Improved action regression is therefore not accompanied by a comparable reduction in the measured reconstruction error. This is consistent with using reconstruction as an anchor while evaluating future usefulness through the action objective.

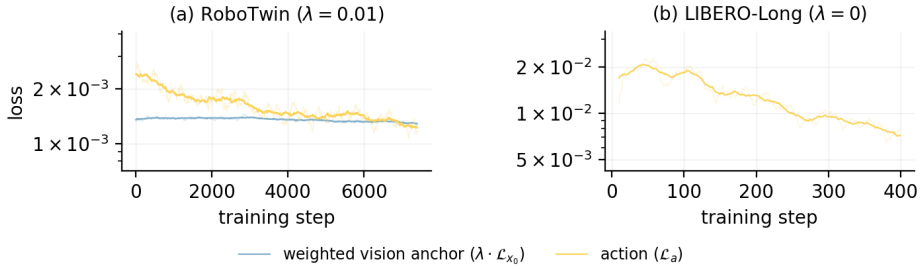


Figure 6: **Stage-2 training dynamics.** The action expert is frozen during the plotted phase. RoboTwin (left) uses $\lambda_v = 0.01$ and shows both action loss and the weighted future-anchor loss. LIBERO-Long (right) uses $\lambda_v = 0$ and shows only action loss. The vertical scale is logarithmic.

These curves characterize training objectives, not success rates. They do not establish that visual quality is unchanged, that the anchor has no optimization effect, or that all action improvement lies in a particular subspace. The next experiment examines the last question directly in the learned predictions.

C.2 Which Learned Prediction Differences Matter for Control?

Comparison and intervention. We evaluate 64 RoboTwin contexts with demonstrated history. Let z_i^{V} be the vision-supervised student's future and z_i^{A} DirectWAM's future. Both are decoded by the same frozen single-step action model $\mathcal{A}$ with paired action-side inputs. This comparison holds the model fixed while replacing the predicted future.

For each context, let J_i be the local Jacobian of this model at z_i^{V}. We map 32 sampled action-output directions back to future space and orthonormalize them:

$$Q_i = \mathrm{orth}\left([J_i^\top u_1, \ldots, J_i^\top u_{32}]\right), \qquad P_i = Q_i Q_i^\top. \tag{15}$$

This is a context-local probe of action-sensitive directions, not a claim to recover the complete sensitive subspace or the leading singular vectors. We selectively transfer the learned prediction difference,

$$\Delta z_i = z_i^{\mathrm{A}} - z_i^{\mathrm{V}}, \qquad z_i^{\mathrm{sens}} = z_i^{\mathrm{V}} + P_i \Delta z_i. \tag{16}$$

Unlike the teacher-referenced correction in the main text, this intervention moves between two learned single-step predictions. Its full-replacement reference is z^{A}, not the teacher.

Evaluation. Let $\mathcal{E}(q)$ be the aggregate offline distance between the actions decoded from candidate futures q and the demonstrated actions, using the evaluated action coordinates. The report uses an L_2 action distance, not a closed-loop success metric. We define pooled recovery as

$$\mathrm{Recovery}(q) = \frac{\mathcal{E}(z^{\mathrm{V}}) - \mathcal{E}(q)}{\mathcal{E}(z^{\mathrm{V}}) - \mathcal{E}(z^{\mathrm{A}})}. \tag{17}$$

Recovery is computed from aggregate errors before forming the ratio. It is distinct from a median of per-context recovery ratios. Equal-norm random-subspace and Gaussian perturbations provide controls for correction magnitude.

Table 7: **Selective replacement of learned prediction differences.** Results use 64 RoboTwin contexts and the same frozen action model. The selected component accounts for 1.9% of the prediction-difference energy and recovers approximately 80% of the pooled offline action-error improvement. Dashes indicate that only the recovery summary is reported for the random controls.

Prediction	GT-action error ↓	Recovery ↑
Vision-supervised z^{V}	0.257	0%
Full replacement z^{A}	0.216	100%
Selective replacement z^{sens}	0.224	80%
Equal-norm random controls	–	1–2%

Results and scope. Table 7 shows that the effect of the learned prediction difference is highly non-uniform under the frozen action model. Full replacement reduces the offline demonstration-action error from 0.257 to 0.216, while selectively transferring a locally action-sensitive component containing only 1.9% of the prediction-difference energy reduces the error to 0.224, recovering approximately 80% of the full offline improvement. Equal-norm random controls recover only 1–2%, indicating that the benefit depends strongly on the direction of the correction rather than its magnitude alone. We further test whether this effect persists in closed-loop control. As shown in Table 8, starting from the vision-supervised future, transferring only the locally action-sensitive component of the learned DirectWAM difference, corresponding to 1.4% of the difference energy, increases RoboTwin success from 78.2% to 87.3%, essentially matching the 87.0% obtained by full replacement with the DirectWAM future. Together, these interventions show that the downstream consequences of the learned prediction difference are highly non-uniform: a small locally action-sensitive component is sufficient to recover a disproportionate share of the offline action improvement and essentially the full closed-loop control gain. These results do not imply that the complete learned correction is concentrated in a fixed action-relevant subspace; rather, they show that a small local component can affect its downstream control consequences.

C.3 Teacher-Referenced Action-Sensitive Error Recovery

This section details the diagnostic in Fig. 3b. It compares selective corrections toward a full-teacher future for the vision-supervised student, Flash-WAM, and naive one- and two-step predictors. The reference is the teacher prediction, rather than the demonstrated future or the output of another learned student.

Local projection. Let $\mathcal{A}(z, h)$ denote the frozen action model for conditioning context h and future representation z. At the base prediction z^{B}, we form the local Jacobian $J = \partial\mathcal{A}(z, h)/\partial z|_{z=z^{\mathrm{B}}}$. We map $r = 32$ sampled action-output directions u_i back into representation space and orthonormalize the resulting vectors:

$$Q = \mathrm{orth}\big([J^\top u_1, \ldots, J^\top u_r]\big), \qquad P = QQ^\top. \tag{18}$$

This Jacobian-based construction supplies a context-local diagnostic subspace of rank at most 32. It does not identify the complete action-sensitive subspace or necessarily recover the leading singular vectors of J.

Table 8: **Closed-loop selective replacement of learned prediction differences.** We keep the same downstream action model fixed and vary only the future representation supplied to it. The selective correction transfers only the locally action-sensitive component of the difference between the vision-supervised and DirectWAM futures.

Future representation	Transferred energy	Success rate (%) ↑
Vision-supervised z^V	0%	78.2
Selective replacement $z^V + P(z^A - z^V)$	1.4%	**87.3**
DirectWAM z^A	100%	87.0

Selective correction. For $\Delta z = z^{\mathrm{T}} - z^{\mathrm{B}}$, the two components in Eq. 1 define candidate futures

$$z^{\mathrm{sens}} = z^{\mathrm{B}} + P\Delta z, \qquad z^{\mathrm{comp}} = z^{\mathrm{B}} + (I - P)\Delta z. \tag{19}$$

Each candidate is decoded separately using the frozen model. The full correction yields $z^{\mathrm{B}} + \Delta z = z^{\mathrm{T}}$ and provides the reference improvement. Because P is an orthogonal projector, the squared discrepancy decomposes as $\|\Delta z\|_2^2 = \|P\Delta z\|_2^2 + \|(I - P)\Delta z\|_2^2$. The sensitive energy fraction is therefore $\|P\Delta z\|_2^2 / \|\Delta z\|_2^2$ for a nonzero discrepancy. Here, discrepancy refers to distance from the teacher prediction, not necessarily error relative to the true future.

Action-error recovery. Let $\mathcal{E}(q)$ denote the offline error of the actions decoded from candidate futures q relative to demonstrated actions, with the same evaluation cases and error aggregation used for the base and teacher references. The normalized recovery compares the candidate's improvement with that obtained by full teacher replacement:

$$\mathrm{Recovery}(q) = \frac{\mathcal{E}(z^{\mathrm{B}}) - \mathcal{E}(q)}{\mathcal{E}(z^{\mathrm{B}}) - \mathcal{E}(z^{\mathrm{T}})}. \tag{20}$$

The denominator is the available teacher-referenced improvement, not the total base action error; interpreting the ratio as recovery presumes a positive reference improvement. A value of 100% means matching the teacher-reference action error, rather than eliminating all action error. The metric measures offline action prediction, not closed-loop success. Recoveries from the two separate corrections need not sum to 100%, because the model and the action-error functional can respond nonlinearly to finite changes in the future representation.

Interpretation and scope. Sensitivity is favored by the construction of P. What is measured is whether correcting the actual teacher–prediction discrepancy along those directions reduces demonstration-action error, and by how much; local sensitivity alone does not specify that finite-correction outcome. Across the four predictors, the selected component contains 0.31–0.85% of discrepancy energy and recovers 16.3–34.0% of the available action-error improvement. For the vision-supervised student, the corresponding comparison is 0.85% energy and 34.0% recovery for the selected component, versus 99.15% energy and 24.8% recovery for its complement. The complement remains useful and is not an action-insensitive nullspace. These observations support using downstream action consequences in addition to representation distance; they do not establish that this probe explains the entire closed-loop performance gap or that learning should be restricted to its subspace. The projection is used only for diagnosis and is not required by the training objective.

For scale, an independent uniformly random rank-r subspace in a d-dimensional representation captures an expected fraction r/d of the energy of a fixed discrepancy, approximately 0.07% for rank 32 in this representation. This is a geometric reference for energy concentration, not an empirical random-correction baseline for action-error recovery.

C.4 VISUAL PREDICTION QUALITY

Setting. We evaluate predicted future videos offline under a teacher-forced protocol, where ground-truth history is committed to the KV cache and each model predicts the next future chunk independently. On RoboTwin, we evaluate 4,739 valid contexts sampled from the 2,500 clean demonstrations. We compare multi-step teacher variants, the one-step teacher, Flash-WAM, and DirectWAM, using the same random seed and benchmark-specific guidance settings. Predicted and ground-truth latent chunks are decoded independently with the Wan2.2 VAE, and metrics are computed per camera and then averaged. We report PSNR, SSIM, LPIPS, latent MSE, and distribution-level FID/KID; confidence intervals are computed over contexts.

Table 9: **Visual prediction quality.** Teacher variants use guidance scale 5 with the indicated number of video denoising steps. Flash-WAM and DirectWAM use one step. Bold indicates the best value among the listed methods.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	KID $\times 10^3$ ↓
Teacher, 25 steps	30.37	0.938	**0.051**	**2.8**	**0.56**
Teacher, 8 steps	30.16	0.936	0.054	3.5	0.93
Teacher, 4 steps	29.64	0.932	0.060	5.0	1.78
Teacher, 2 steps	28.57	0.920	0.075	8.1	3.74
Teacher, 1 step	25.80	0.885	0.123	17.4	10.24
Flash-WAM	26.62	0.928	0.077	10.5	4.11
DirectWAM (ours)	**31.79**	**0.949**	0.052	5.9	2.06

DirectWAM retains high-quality visual predictions despite being trained primarily for downstream control. As shown in Tab. 9, directly truncating the diffusion teacher progressively degrades future prediction quality, with the one-step teacher dropping to 25.80 PSNR, 0.885 SSIM, and 0.123 LPIPS. Flash-WAM partially recovers this degradation, while DirectWAM achieves the strongest reference-based reconstruction quality among the single-step methods, reaching 31.79 PSNR, 0.949 SSIM, and 0.052 LPIPS. Notably, its LPIPS is nearly identical to the full 25-step teacher (0.051), although distribution-level metrics remain worse than the full teacher (FID 5.9 vs. 2.8; KID 2.06 vs. 0.56). These results indicate that optimizing the predicted future through downstream action supervision does not collapse the intermediate representation into an opaque action code: DirectWAM preserves substantial visual structure while converting the iterative future predictor into a single-step mapping.