What Makes World Action Models Generalize? An Empirical Study of Test-Time Future Modeling

Renping Zhou[1,*], **Zanlin Ni**[1,*,†], **Zihao Fan**[2,*], **Guohao Fu**[3], **Zeyu Liu**[1], **Hao Shi**[1], **Jie Zhang**[1], **Chi Bene Chen**[1], **Yang Yue**[1], **Xueyang Fu**[2], **Gao Huang**[1,✉]

[1]Tsinghua University, [2]University of Science and Technology of China, [3]Beijing Institute of Technology

World action models (WAMs) predict the future alongside actions during *training*. Due to the heavy computation cost of video denoising, whether the future must still be generated during *inference* is disputed: Explicit WAMs denoise it into clean frames along with every action chunk, whereas Latent WAMs discard it entirely for acceleration. We conduct an empirical comparison of the two paradigms on robot manipulation using a matched backbone, data, budget, and protocol, varying only whether the action expert conditions on the future. Latent WAMs perform on par with explicit ones on in-distribution tasks, but fall considerably behind on all three generalization axes. Further analysis shows that the gap arises almost entirely from the first denoising step: the benefit comes from *preparing* the future, not *generating* it. We therefore propose **Simple-WAM**, which simplifies future modeling into a single forward pass of fully noised video tokens and adapts the training-time noise schedule to this inference behavior. Across simulation and real-world tasks, Simple-WAM achieves the best of both worlds, leading explicit WAMs in generalization performance with efficiency comparable to Latent WAMs.

Project Page: https://simple-wam.github.io

1. Introduction

Building generalizable robot policies is a long-standing goal of embodied AI. World Action Models (WAMs) pursue it by jointly generating future dynamics and actions conditioned on observations and language instructions, which supplies supervision in observation space beyond sparse action labels (Ye et al., 2026; Li et al., 2026b; Bi et al., 2026; Kim et al., 2026). Initialized from video generation models trained on web-scale video data, WAMs inherit rich spatiotemporal priors and shift action learning from dense state-action imitation toward inverse dynamics that aligns motor commands with predicted visual futures, and it is this capacity to model the future that is credited with their stronger robustness and generalization (Cheang et al., 2024; Pai et al., 2025; Hu et al., 2025). This sets them apart from Vision-Language-Action (VLA) models (Zitkovich et al., 2023; Black et al., 2024; 2025; Liu et al., 2025; Kim et al., 2025; Shi et al., 2026b), whose backbones are pretrained predominantly on static image-text pairs and optimized for understanding or reasoning rather than for generation, and which therefore carry little of the dynamic understanding of temporal evolution that manipulation demands (Zhou et al., 2025; Guruprasad et al., 2025; Ni et al., 2026).

While the value of future modeling to WAMs during training is widely accepted, whether the future must also be modeled at inference, and through what mechanism it would then act on the action, is still an underexplored question. Early WAMs (Hu et al., 2025; Cheang et al., 2024; Ye et al., 2026; Kim et al., 2026; Pai et al., 2025) adopt the *explicit paradigm*, in which the future is denoised into clean frames alongside every action chunk and the action is conditioned on them, and report that the success rate tracks the quality of that generated future, which established it as an indispensable component of inference (Ye et al., 2026; Li et al., 2026b). Because video tokens

[*] *Equal contribution.* [†] *Project leader.* [✉] *Corresponding author.*

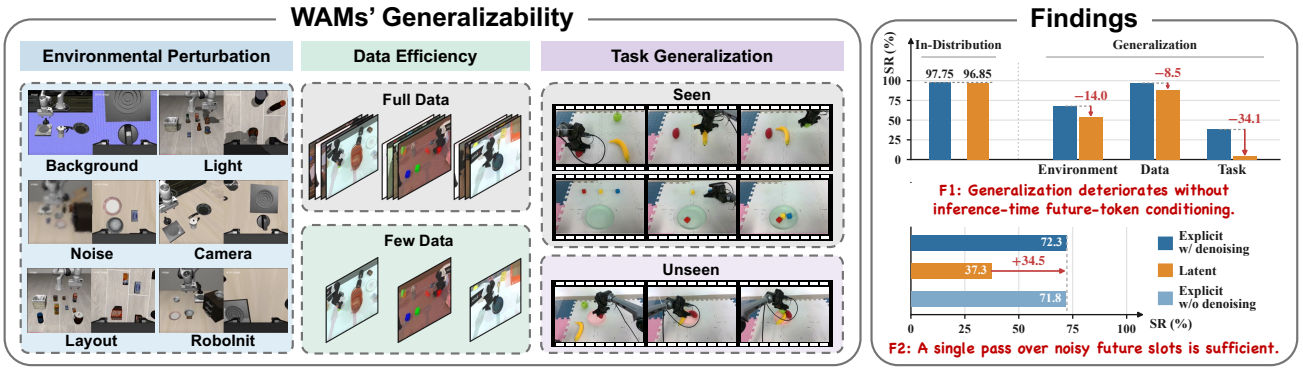

Figure 1: **An empirical study of test-time future modeling for WAM generalization. Left:** the three axes of the protocol, environmental perturbation, data efficiency, and task generalization. **Right:** the explicit and latent paradigms are within 0.9 points on in-distribution tasks yet separate on all three axes (F1), and a single forward pass over a fully noised future recovers almost all of that separation (F2).

far outnumber action tokens, however, that denoising dominates the per-chunk cost and hinders high-frequency closed-loop control. More recent works (Yuan et al., 2026; Ni et al., 2026) focused on efficiency argue instead that future modeling contributes to the policy mainly as a training objective rather than as a test-time requirement, and propose a *latent paradigm* that retains video supervision during training but skips the heavy video generation at inference, reporting little difference from explicit WAMs at a fraction of the cost. Motivated by these discussions, we raise the following question: is inference-time video generation necessary after all?

We answer this through a controlled comparison of the two paradigms, and find that neither claim is wrong on the evidence that supports it. A latent policy does match an explicit one on in-distribution performance, and it does so at a fraction of the cost. They stop holding on generalization, the ability that motivated future modeling to begin with and the one that the efficiency comparisons leave out. We conduct the comparison along three axes of generalization—robustness to environmental perturbations, data efficiency, and task generalization—with all other variables held fixed, so that the only variable is whether, and in what form, the action expert conditions on the future. We have two findings, demonstrated in Fig. 1. *(i) Generalization deteriorates without inference-time future modeling.* Under a strictly matched setting, latent WAMs match explicit WAMs in distribution, yet degrade consistently across all three generalization settings. Inference-time future modeling is therefore not an optional component to be traded away for efficiency; it is an important source of generalizable temporal information. *(ii) Fully noised future is sufficient for generalization.* Conditioning the policy on future video tokens that are still at pure noise, never denoised into clean future frames, already recovers most of the gap. The generalization benefit thus comes from inference-time future conditioning itself, rather than from the fidelity of the generated future.

Guided by these findings, we propose **Simple-WAM**, which restores inference-time future conditioning to latent WAMs at negligible additional cost. Simple-WAM retains the single-pass prefill and inference cost of the latent paradigm, but keeps the future in the policy's context as future video tokens left at noise. At inference, the video expert runs once instead of over the whole schedule, and training shifts its flow-time sampling toward that same point. As a result, neither stage pays for the denoising Finding 2 shows to be unnecessary. Simple-WAM leads the explicit paradigm on seven of the eight measurements of Tab. 2 and the latent one on all eight, reaching 79.5 under environmental perturbation against 67.7 and 53.8, while running at 3.8× the speed of the explicit paradigm and within 1.2× the latency of the latent one. Therefore, the choice between the two paradigms is not the trade-off between generalization and efficiency it is usually taken to be.

2. Related Works

2.1. World Action Models and Inference Efficiency

Using generated video as an intermediate for control predates the current formulation: early predictive policies imagine a future from the current observation and recover the action from it by inverse dynamics (Du et al., 2023;

Zhou et al., 2024; Hu et al., 2025; Shi et al., 2026a), and large-scale video pretraining was shown to transfer to manipulation ahead of any action label (Wu et al., 2024; Cheang et al., 2024). World action models tighten this coupling into a single network that predicts future frames and actions jointly (Ye et al., 2026; Li et al., 2026b; Bi et al., 2026; Kim et al., 2026; Pai et al., 2025). They attribute the advantage to a training signal defined over the observation rather than the action alone, and to a video backbone carrying a prior on how a scene evolves. These works demonstrate gains in sample efficiency, robustness to scene variation, and transfer to tasks and embodiments beyond the action data (Ye et al., 2026; Pai et al., 2025; Li et al., 2026b).

That prediction is paid for at inference, where the video branch is denoised once per action chunk over far more tokens than the action branch. A number of recent works therefore examine how an efficient WAM should be built, along the model architecture (Li et al., 2026c; Zhao et al., 2026), the inference-time formulation (Yuan et al., 2026; Zhang et al., 2026a; Ni et al., 2026), and the training recipe (Li et al., 2026a; Luo et al., 2026; Zhang et al., 2026b). Among them, the latent paradigm makes the strongest claim: that future prediction contributes as a training objective rather than as a test-time requirement, so its video branch is trained as usual but never asked to produce a future at deployment (Yuan et al., 2026; Ni et al., 2026). Success rates close to the explicit paradigm's are reported at a fraction of its latency, so whether the future must be generated at inference is still contested. That evidence is, however, collected in distribution. Concurrent work such as Faster-WAM (Zhao et al., 2026) attributes a shortfall on the perturbation axis to the absent future, with efficiency as its target. We ask whether removing test-time future modeling is genuinely free, and, where it is not, what relation holds between explicit future generation and generalization.

2.2. Generalization of Embodied Policies

Generalization is a key ability for embodied agents, and VLA and WAM policies alike are evaluated on it not as a single quantity but along dimensions that differ in what changes at deployment (Liu et al., 2023; Chen et al., 2026; Guruprasad et al., 2025). Closest to the training distribution is variation that leaves the task unchanged, which LIBERO-Plus decomposes into seven factors covering the camera, the scene, the language, and the robot's initial state (Fei et al., 2026). Holding the task fixed and reducing its supervision gives data efficiency, measured by success under fewer demonstrations per task, where video-pretrained policies report their clearest gains (Pai et al., 2025; Li et al., 2026b; Ye et al., 2026). Furthest is the case in which the task itself changes, studied as transfer to tasks withheld from training (Zhou et al., 2025; Guruprasad et al., 2025) or as the acquisition of a skill from action-free video of it (Ye et al., 2026; Li et al., 2026b). These properties are claimed or observed across prior WAM work, but each is demonstrated by one system under its own backbone, data, and training budget, and the dimensions are seldom placed side by side. We consolidate them into three axes and compare the paradigms under one matched setting that isolates inference-time future modeling, so as to ask what makes WAMs generalize and how that generalization can be retained at the smallest inference cost.

3. Preliminaries

3.1. World Action Models

We consider language-conditioned visuomotor control from demonstrations. At control step t, the policy receives one or more image observations o_t, a language instruction ℓ, and a proprioceptive state s_t, and predicts an action chunk $A_t = a_{t:t+H-1}$ of horizon H. The language instruction ℓ and proprioceptive state s_t condition every stage of the model. For notational simplicity, we omit them below.

A world action model couples a video expert with parameters θ, initialized from a pretrained video generator, and an action expert with parameters ψ (Ye et al., 2026; Li et al., 2026b; Bi et al., 2026; Kim et al., 2026). The video expert reads a single sequence of latents spanning the current and future frames. Let $\mathcal{E}$ be the visual encoder of the backbone and $y_t = \mathcal{E}(o_t)$ the latent of the current observation, which is given and therefore clean, while the latents $y_{t+1:t+T}$ of the T future frames are unknown at inference and carry a flow time $\tau \in [0, 1]$, with $\tau = 0$ at clean data and $\tau = 1$ at Gaussian noise. Writing $y^{\tau}_{t+1:t+T} = (1 - \tau)\, y_{t+1:t+T} + \tau\epsilon$ for the interpolant, the video expert operates on

$$Y^{\tau}_t = \left[\, y_t \,;\, y^{\tau}_{t+1:t+T} \,\right], \tag{1}$$

and regresses the velocity field at the future video tokens,

$$\mathcal{L}_{\text{vid}} = \mathbb{E}_{\tau,\epsilon} \left\| u_\theta(Y_t^\tau, \tau) - (\epsilon - y_{t+1:t+T}) \right\|^2 . \tag{2}$$

The action expert regresses its own velocity field v_ψ over an independent flow time σ and interpolant A_t^σ, and training minimizes $\mathcal{L} = \mathcal{L}_{\text{act}} + \lambda \mathcal{L}_{\text{vid}}$. Both flow times follow a schedule $\mathcal{S}$ that serves training and inference alike: the expectation over τ in Eq. (2) draws $\tau = \mathcal{S}(u)$ with $u \sim \mathcal{U}[0, 1]$, and the steps τ_k of Sec. 3.2 are the same schedule discretized. Because τ and σ are sampled independently, the action expert is trained against future latents at every noise level, including τ near 1.

3.2. Inference-Time Future Conditioning

At inference the action chunk is produced by integrating the action velocity field from $\sigma_1 = 1$ to $\sigma_{K+1} = 0$ over K steps, and the executed chunk is the resulting A_t^0. Let Φ_θ denote the map from the video expert to the representation read by the action expert. At step k the action latent is updated by

$$A_t^{\sigma_{k+1}} = A_t^{\sigma_k} + (\sigma_{k+1} - \sigma_k)\, v_\psi(A_t^{\sigma_k},\, c_{t,k},\, \sigma_k) , \tag{3}$$

where the future conditioning feature supplied at that step is

$$c_{t,k} = \Phi_\theta(Y_t^{\tau_k}) . \tag{4}$$

Both paradigms are instances of Eq. (3), and two quantities in Eq. (4) distinguish them: whether future video tokens are present in the video sequence, and what schedule the flow time τ_k of those positions follows.

Explicit WAMs keep the future video tokens and drive their flow time toward clean data,

$$c_{t,k} = \Phi_\theta(Y_t^{\tau_k}) , \qquad \tau_1 = 1 \;\longrightarrow\; \tau_{K+1} = 0. \tag{5}$$

The action expert is therefore conditioned on a progressively sharper future rather than on a single fully denoised one: at every step it reads the future at that step's noise level. Jointly denoising models step the video and action experts along this shared schedule (Bi et al., 2026; Li et al., 2026b; Ye et al., 2026; Kim et al., 2026). Generate-then-act models are the special case in which the video is denoised to $\tau = 0$ first and the action expert reads $\Phi_\theta(Y_t^0)$ at every step.

Latent WAMs drop the future video tokens from the sequence at inference while retaining future prediction during training (Yuan et al., 2026; Ni et al., 2026),

$$c_{t,k} = \Phi_\theta([\, y_t \,]) \quad \text{for all } k. \tag{6}$$

The video expert still makes one forward pass, on the current-frame latent alone, but no flow time is defined and the feature read by the action expert is constant across the K steps.

Inference cost. Let C_{vid} be the cost of one video-expert forward pass carrying future video tokens, C_{cur} the cost of one pass on $[\, y_t \,]$ alone, and C_{act} the cost of one action step. Reaching $\tau_{K+1} = 0$ requires denoising the video over the whole schedule, so the explicit paradigm costs $K\, C_{\text{vid}} + K\, C_{\text{act}}$, whereas the latent paradigm costs $C_{\text{cur}} + K\, C_{\text{act}}$. Two factors make C_{vid} large relative to C_{act}. The video expert carries the future video tokens and therefore a far longer sequence than the action expert, $|y_{t+1:t+T}| \gg |A_t|$, and it is also the larger of the two, since it is initialized from a pretrained video generator while the action expert is comparatively small. The video term therefore dominates in the explicit case and is the source of the latency gap reported in Sec. 5. The cost is set by how far τ_k is driven toward 0, not by how many times the action expert reads the future: a schedule held near $\tau = 1$ requires no denoising at all.

Prior work has compared only these two settings, which differ in both factors at once. Our study holds every other component of Eq. (3) fixed and varies them separately.

4. A Controlled Study of Inference-Time Future Conditioning

4.1. Evaluation Protocol for Generalizability

The generalization benefits of WAMs have been widely discussed (Ye et al., 2026; Pai et al., 2025; Li et al., 2026b). However, a systematic evaluation of this capability, and a controlled comparison across paradigms

and components, are still lacking. We therefore propose a protocol including three axes, as shown in Fig. 1, to systematically evaluate the generalization capabilities of WAMs.

Environmental perturbation. Trained to predict how a scene evolves and initialized from video models carrying rich spatiotemporal priors, WAMs are expected to remain robust under conditions not observed during training (Ye et al., 2026). We evaluate the in-distribution checkpoint without retraining, under the seven perturbation factors of LIBERO-Plus (Fei et al., 2026): robot initial states, camera viewpoints, language instructions, sensor noise, backgrounds, object layouts, and lighting.

Data efficiency. WAMs are reported to retain their competence when the demonstrations available per task are reduced (Ye et al., 2026; Pai et al., 2025; Kim et al., 2026). Such data efficiency is attributed to video pretraining, which already supplies the physical dynamics of how objects move and interact. LIBERO provides 40 to 50 demonstrations per task; we retrain each configuration from the same initialization with the per-task count reduced to 10, and evaluate in distribution to isolate the impact of demonstration count.

Task generalization. WAMs trained across diverse tasks are reported to acquire skills for unseen tasks without any additional data or merely by seeing the operation video (Ye et al., 2026; Li et al., 2026b). We evaluate this ability under two settings: *without video*, where no data of the held-out tasks is available at any stage, and *with video*, where video of those tasks is available but carries no action labels. We use it to train the video branch only. Both settings use four-fold cross-validation over the four LIBERO suites, spatial, object, goal, and long (Liu et al., 2023): each fold withholds one suite and trains on the remaining three, and we report the average success rate.

To compare the paradigms rather than their implementations, every configuration in this section follows the architecture and training hyperparameters of Fast-WAM (Yuan et al., 2026): a pretrained Wan2.2-5B video DiT (Wan et al., 2025) as the video backbone, reusing its text encoder and video VAE, and a 1B action expert initialized from the interpolation of the video DiT. Following Fast-WAM, we use the same flow-time sampling strategy during training and the corresponding discretized schedule during inference. The explicit and latent paradigms differ only in a structured attention mask, which decides whether the action tokens may attend to the future video tokens. On each of the three axes the training budget is the same as in distribution.

4.2. Finding 1: Generalization Deteriorates without Inference-time Future Modeling

We first ask whether the future can be removed at inference without cost. Tab. 1 compares the explicit and latent paradigms. In distribution the two are comparable, at 97.75 against 96.85, in line with what latent WAM works report (Ni et al., 2026; Yuan et al., 2026). However, a clear gap emerges under the three axes of our protocol: the explicit paradigm leads by 13.97 points under environmental perturbation and by 8.45 under data efficiency. Task generalization shows the largest separation. With no data of the held-out tasks both paradigms stay near the floor, the explicit one slightly ahead. Acquiring a skill from action-free video is one of the properties claimed for WAMs (Ye et al., 2026; Li et al., 2026b), and only the explicit paradigm realizes it: such video is worth 63.65 points to it and 3.80 to the latent paradigm, which reaches 5.90 against 69.90.

Table 1: **Generalization Deteriorates without Inference-time Future Modeling.** Success rate (%) under the matched setting of Sec. 4.1. w/ vid: action-free video of the held-out tasks with video available. Δ is Explicit minus Latent; the in-distribution column is greyed, and the generalization deltas are **bold**.

	In-dist.	Generalization			
		Perturb.	Data Eff.	Task Gen.	
Paradigm	LIBERO	LIBERO-Plus	10-shot	w/o vid	w/ vid
Explicit	97.75	67.72	96.95	6.25	69.90
Latent	96.85	53.75	88.50	2.10	5.90
Δ	0.90	**13.97**	**8.45**	**4.15**	**64.00**

The gap shows that an in-distribution match is not sufficient for comparing the two paradigms. The latent paradigm removes future modeling at inference to reduce cost, and loses generalization as a result. Future modeling is therefore a requirement at inference, not only a training objective.

4.3. Finding 2: Fully Noised Future is Sufficient for Generalization

Finding 1 establishes that the future must be modeled at inference for a WAM to generalize, and leaves open how much of it is required. A natural question is whether a future denoised more clearly yields a stronger policy, and

what kind of future is sufficient for that generalization. The same question matters for efficiency, since denoising the future dominates the per-chunk cost (Sec. 3.2).

To answer this question, we conduct an experiment that changes how far the future is denoised. We take a model trained under the explicit paradigm and run its video expert for only the first t_{denoise} of the K steps of the schedule, holding the feature it supplies fixed for the rest of the action chunk, so that $c_{t,k} = c_{t,t_{\text{denoise}}}$ for every $k > t_{\text{denoise}}$. t_{denoise} counts how many times the video expert is run before the action expert stops seeing the future change. It is set at inference alone: nothing is retrained, the weights and the action denoising are those of the explicit paradigm, and t_{denoise} is the only quantity that varies. At $t_{\text{denoise}} = K$ this recovers Eq. (5), the explicit paradigm itself.

The sweep is reported in Fig. 2, and almost all of the gain comes from the first pass. At $t_{\text{denoise}} = 1$ the video expert is run once, at $\tau_1 = 1$, where $y^{\tau}_{t+1:t+T} = \epsilon \sim \mathcal{N}(0, I)$: the future video tokens of Eq. (1) are pure Gaussian noise, so nothing about the future has been explicitly denoised when the action expert conditions on it. Even so, this single pass outperforms the latent paradigm by 26.44, 8.2, 13.4 and 89.8 points across the four settings. The nine forward passes that follow, which carry the entire cost of denoising the video, reach at best -1.62, $+1.8$, $+0.8$ and $+0.8$ against it.

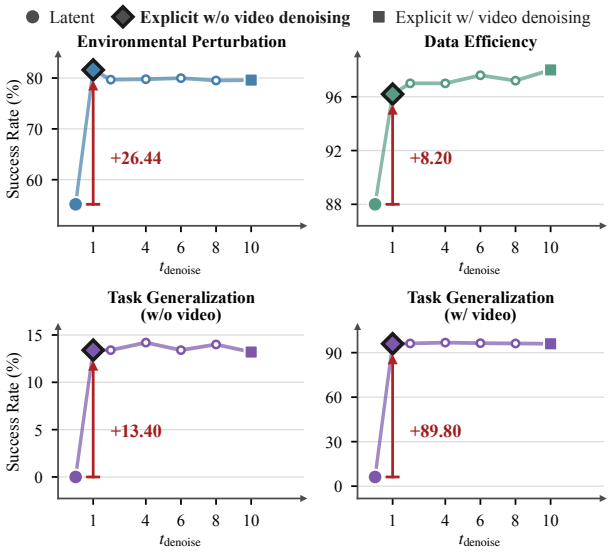

Figure 2: **Fully Noised Future Is Sufficient for Generalization.** Success rate (%) on LIBERO-Spatial for a model trained under the explicit paradigm. The video expert is run for t_{denoise} of the K denoising steps; no retraining is performed.

The gain therefore does not come from the future the video expert generates, but from the single pass that forms the intermediate representation the action expert reads. This implies a misalignment between the visual fidelity the video model is trained for and the feature the action expert requires: a fully noised future is already sufficient.

5. Method

Guided by two findings in Sec. 4, in this section, we propose **Simple-WAM**, which makes two changes to the explicit paradigm, and keeps only the part of the video expert that the two findings call for. Since each denoising step runs the video expert once, dropping the steps Finding 2 shows to be unnecessary removes most of the inference cost. Under Eq. (4), Simple-WAM balances the two paradigms: the future video tokens are conditioned on, as in the explicit one, and the video expert makes a single forward pass, as in the latent one (Fig. 3). Neither change adds a module or a loss term, and together they approach the inference cost of a latent WAM with the generalization of an explicit one.

5.1. Inference: One Pass over a Fully Noised Future

During inference, Simple-WAM give up the iterative denoising of the future video tokens from the explicit paradigm. The video expert makes one forward pass over the sequence with its future video tokens left at Gaussian noise, and the feature it produces conditions all K action steps,

$$c_{t,k} = \Phi_\theta\left(Y_t^{\tau=1}\right) \quad \text{for all } k. \tag{7}$$

The action denoising of Eq. (3) is unchanged. Against the two paradigms of Sec. 3.2, this keeps the future video tokens the explicit one reads and pays only for the single forward pass the latent one makes: $C_{\text{vid}} + K\,C_{\text{act}}$, against $K\,C_{\text{vid}} + K\,C_{\text{act}}$ and $C_{\text{cur}} + K\,C_{\text{act}}$.

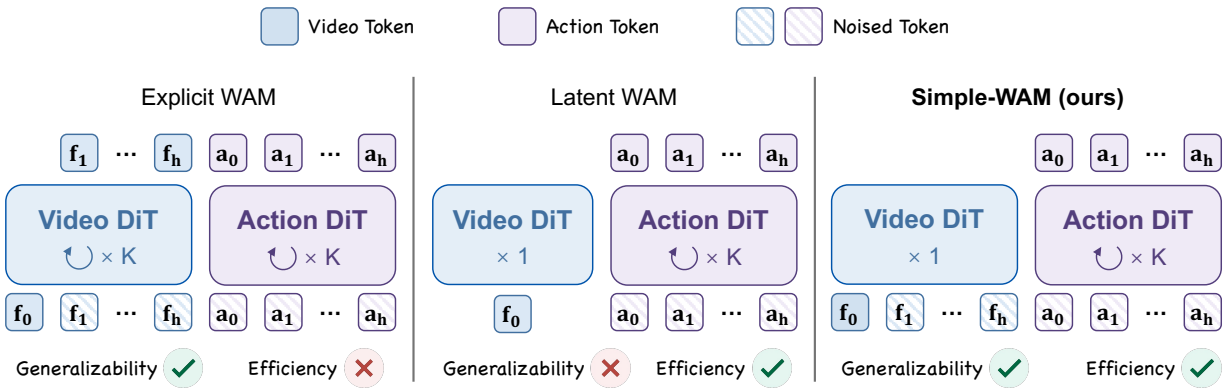

Figure 3: **The three conditioning schemes.** The explicit paradigm denoises the future video tokens across the schedule; the latent paradigm drops them and keeps only the single forward pass. We propose Simple-WAM, which keeps both the future video tokens and the single forward pass: future video tokens are input as pure Gaussian noise at $\tau = 1$ and are never denoised.

5.2. Training: A Mixed Schedule

Sec. 5.1 rests the entire future feature on an input at $\tau = 1$, while the explicit paradigm trains its video expert by denoising across the whole schedule. The share of the training budget spent at $\tau = 1$ is therefore negligible, while at inference it is the only flow time used. To strengthen the ability to read a feature out of pure noise, we raise that share with a single scalar p and draw the flow time of the future video tokens as

$$\tau = \begin{cases} 1, & \text{with probability } p, \\ \mathcal{S}(u), \quad u \sim \mathcal{U}[0,1], & \text{otherwise,} \end{cases} \tag{8}$$

where $\mathcal{S}$ is the flow-time schedule of the explicit paradigm. Training is otherwise unchanged: the two experts are trained jointly as in Sec. 3, the action expert reads the future video tokens through Φ_θ, and $\mathcal{L}_{\text{vid}}$ is applied at the sampled τ. The remaining $(1-p)$ keeps the video expert supervised across the rest of the schedule. The mixture therefore sharpens the feature the video expert produces at the one flow time inference reads, while $(1-p)$ leaves it under the flow-time sampling it was originally trained with. We set $p = 0.5$ and ablate it in Sec. 6.4.

6. Experiments

6.1. Implementation Details

Simple-WAM is built on the setting of Sec. 4.1. We use pretrained Wan2.2-5B (Wan et al., 2025) as the video backbone, set the action chunk to $H = 32$, and use 9 video frames per chunk. During training we set the video loss weight to $\lambda = 3.0$. And we use $K = 10$ denoising steps with classifier-free guidance (CFG) scale set to 1.0 at inference time. All remaining training settings follow Fast-WAM (Yuan et al., 2026). All models are trained on NVIDIA A800 GPUs, and latency numbers are measured on a single NVIDIA GeForce RTX 5090 GPU under each configuration's own denoising settings. We provide more hyperparameter settings in Appendix A.1.

6.2. Experimental Setup

We evaluate Simple-WAM on LIBERO (Liu et al., 2023), RoboTwin 2.0 (Chen et al., 2026), and LIBERO-Plus (Fei et al., 2026), along the three axes of Sec. 4.1.

LIBERO. LIBERO comprises four suites covering spatial relations, object-centric skills, goal-conditioned tasks, and long-horizon behaviors, each with 10 tasks and 500 expert demonstrations. We follow the standard protocol in distribution, and form the other two axes as in Sec. 4.1: for data efficiency we retrain with the demonstrations per task reduced to 10 and to 5, and for task generalization we run four-fold cross-validation over the suites.

RoboTwin 2.0. RoboTwin 2.0 is a bimanual manipulation benchmark with tasks that require coordinated dual-arm

Table 2: **Main results across the three generalization axes.** Success rate (%) under the protocol of Sec. 4.1, per factor for environmental perturbation and averaged over suites elsewhere. w/o vid. and w/ vid. denote task generalization without and with action-free video of the held-out tasks; such video provides no training signal in a VLA, so "*" repeats the w/o vid. entry. "Emb. P.T." denotes large-scale embodied pretraining; those rows are grey, and **bold** indicates the best among the rest. "–": not covered by the official release. Detailed results are in Tab. A2.

		Environmental perturbation								Data efficiency			Task generalization			
		LIBERO-Plus								LIBERO		RoboTwin	LIBERO		RoboTwin	
Method	Emb. P.T.	Init.	Cam.	Lang.	Noise	Bg.	Lay.	Light	Avg.	5-shot	10-shot	10-shot	w/o vid	w/ vid	w/o vid	w/ vid
Vision-language-action models																
$\pi_{0.5}$ (2025)	✓	73.6	78.4	80.8	89	94.1	84.5	96.2	84.4	87.5	91.5	33.9	9.4	9.4*	2.5	2.5*
VLA-Adapter (2025)	✗	37.4	36.4	73.8	57.2	**76.6**	70.2	71.0	59.0	77.6	85.9	–	0.3	0.3*	–	–
Explicit WAMs																
Lingbot-VA (2026b)	✓	83.0	86.4	82.3	53.1	40.9	64.4	76.2	70.5	78.5	87.6	3.9	15.2	71.1	0.0	1.1
FastWAM-Joint (2026)	✗	64.9	36.4	94.8	54.8	54.7	78.6	94.9	67.7	91.5	97.0	35.0	6.3	69.9	5.4	**47.3**
Latent WAMs																
FastWAM (2026)	✗	49.5	21.0	73.3	46.7	52.0	63.7	77.5	53.8	78.2	88.5	4.8	2.1	5.9	3.5	4.8
Simple-WAM	✗	**82.1**	**55.6**	**96.2**	**74.2**	72.9	**83.4**	**95.8**	**79.5**	**92.4**	**97.2**	**37.2**	**10.1**	**73.6**	**6.2**	44.5

control, reported under both clean and randomized scenes. We repeat the same two axes here: for data efficiency we retrain with 10 demonstrations per task, and for task generalization we withhold 10 randomly selected tasks.

LIBERO-Plus. LIBERO-Plus extends the LIBERO tasks with seven kinds of variation, which form the environmental perturbation axis of Sec. 4.1. The overall score is computed by weighting the seven variation factors according to their respective task counts, following the official evaluation protocol.

Real-World Evaluation. We conduct real-world experiments on an AgileX Aloha dual-arm platform. We consider four tasks, *Stack the Bowls*, *Store the Blocks*, *Sort in Row*, and *Move in Order* (Fig. 6); the last two are language-conditioned, with the instruction naming the order in which the objects are to be handled. We collect 200 demonstrations per task and jointly train a single policy with a global batch size of 512, concatenating the three camera views into a single image as in RoboTwin. Each task is evaluated over 30 trials on three axes: grasping, placement, and, for the two tasks that specify one, whether the order was followed.

6.3. Main Results

Simulation Results. Tab. 2 reports all three axes. Under environmental perturbation Simple-WAM averages 79.5 over the seven LIBERO-Plus factors and is best on six among models without embodied pretraining, against 67.7 for the explicit paradigm and 53.8 for the latent one. It comes within 4.9 points of $\pi_{0.5}$, which is initialized with large-scale embodied pretraining, and leads every model trained without it by at least 11.8. The margin falls on the four factors that change what the camera sees: a visual shift corrupts the future the explicit paradigm generates, while a future left at noise has nothing to corrupt, consistent with Finding 2 on this axis. With the demonstrations per task reduced it reaches 92.4 and 97.2 on LIBERO at 5 and 10 shots and 37.2 on RoboTwin, comparable to the explicit paradigm on all three, while the latent one collapses on RoboTwin to 4.8. On tasks withheld from training it reaches 10.1 without any data of them and 73.6 with action-free video, against 6.3 and 69.9; the corresponding RoboTwin pairs are 6.2 against 5.4 and 44.5 against 47.3; Tab. A2 breaks both axes down by suite. The latent paradigm falls behind on every axis, as Finding 1 reports: future modeling is a test-time requirement, not only a training objective.

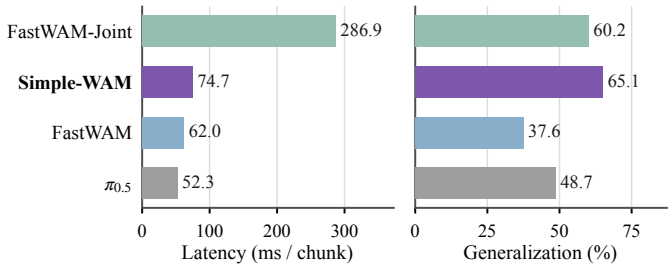

Figure 4: **Inference Latency.** Per-chunk latency and mean success rate over the four generalization settings of Tab. 2.

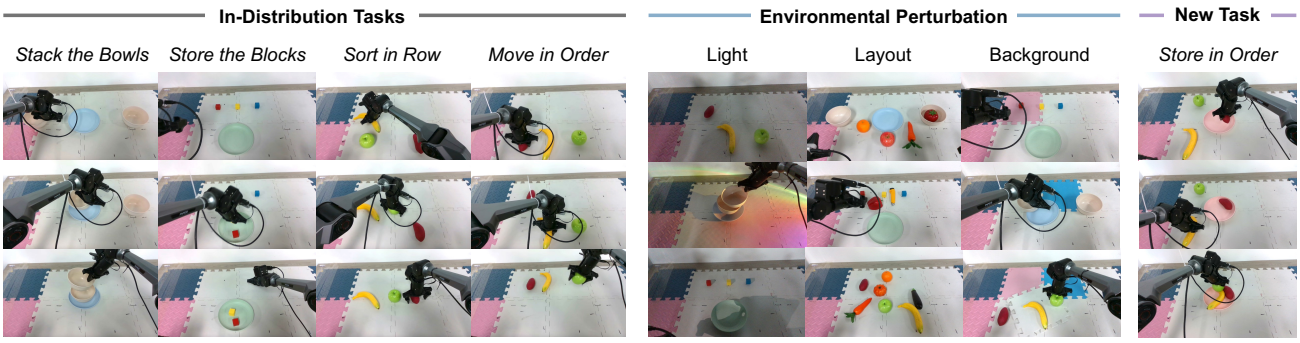

Figure 6: **The real-world setup. Left:** the four training tasks, one per column, each shown at three moments of a single rollout. **Middle:** the three environmental perturbations, varying the lighting, the object layout and the background. **Right:** *Store in Order*, the held-out task.

Efficiency. Fig. 4 places these gains against their cost. Simple-WAM runs at 74.7 ms per action chunk, 3.8× faster than the explicit paradigm at 286.9 ms and within 12.7 ms of the latent one at 62.0 ms, while averaging the best across the four generalization settings. The additional computation does not yield a consistent improvement.

Real-World Evaluation. Fig. 5 reports the four tasks of Fig. 6. In distribution the three models are within 2.3 points of each other, reproducing the parity of Finding 1 on real-world tasks. Under the environmental perturbations the latent paradigm falls to 25.2 while Simple-WAM holds 69.2, and with the demonstrations reduced to 50 per task it holds 69.6 against 37.2. On held-out *Store in Order*, action-free video of the task raises Simple-WAM from 2.2 to 87.8, against 68.9 for the explicit paradigm and 10.0 for the latent one. Tab. A3 gives the per-task numbers and their standard deviations.

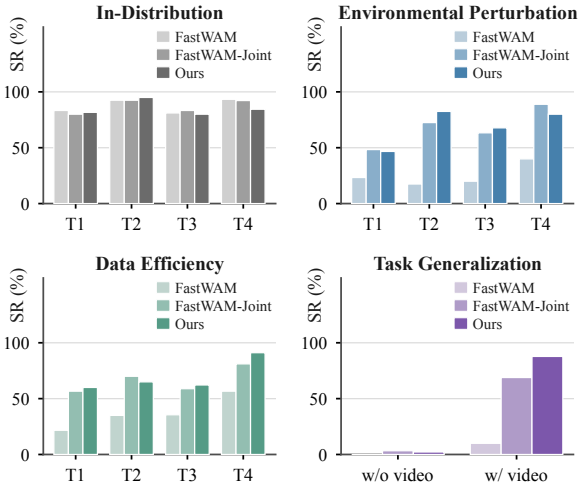

Figure 5: **Real-world results.** Success rate (%) over 30 evaluation runs. T1–T4 are the four training tasks of Fig. 6, in order; task generalization is the held-out *Store in Order*.

6.4. Ablation Study

We ablate LIBERO-Spatial along the three generalization axes, using task generalization with-video.

Probability of the mixed schedule. Tab. 3a sweeps p over its whole range. On the three-axis average, $p = 0.25$, 0.5 and 0.75 differ by 1.3 points, showing that performance is robust to the choice of p. Both endpoints fall lower, to 90.8 and 88.6. At $p = 0$ training reverts to the explicit paradigm and leaves the mismatch that Sec. 5.2 mitigates; at $p = 1$ the flow time never varies, and training departs from the objective the video backbone was pretrained under. Both extremes therefore perform poorly.

Future video tokens. In this part, we investigate whether pure Gaussian noise is a good input for the spatiotemporal information the action expert conditions on. We replace the future video tokens the action expert reads, with learnable query embeddings or with zeros, and keep the video diffusion target in place so that the video branch trains as before. As shown in Tab. 3b, every axis drops, by 27.7, 10.2 and 86.6 points for the queries and by 22.2, 8.2 and 87.4 for the zeros. We attribute this to how far the substitution departs from pretraining: the video backbone saw neither in pretraining, while noised video tokens are exactly what it denoises. These results show that our design is simple yet effective because it keeps the original training strategy and thus benefits most from the video pretraining, stressing pretraining-aligned conditioning.

Table 3: **Ablations on LIBERO-Spatial.** The shaded entries are the default setting and **bold** marks the best.

(a) Mixture probability p

	0	0.25	0.5	0.75	1
Environmental perturbation	83.3	**90.2**	87.5	89.1	85.7
Data efficiency	97.0	97.4	**97.8**	94.4	96.8
Task generalization	92.2	95.8	**97.2**	96.2	83.4
Average	90.8	**94.5**	94.2	93.2	88.6

(b) Future tokens

	Noise	Learn.	Zeros
Environmental perturbation	**87.5**	59.8	65.3
Data efficiency	**97.8**	87.6	89.6
Task generalization	**97.2**	10.6	9.8
Average	**94.2**	52.7	54.9

7. Conclusion

In this work, we compared the explicit and latent paradigms under a matched protocol. The two match in distribution yet diverge on all three generalization axes, and future video tokens left at pure noise recover almost all of that gap. Simple-WAM therefore conditions on a fully noised future in one pass, leading the explicit paradigm on nearly every measurement at $3.8\times$ its speed. Our conclusions rest on a 5B backbone without embodied pretraining; whether they hold at larger scale, or with it, remains open.

References

Bi, H., Tan, H., Xie, S., Wang, Z., Huang, S., Liu, H., Zhao, R., Feng, Y., Xiang, C., Rong, Y., Zhao, H., Liu, H., Su, Z., Ma, L., Su, H., and Zhu, J. 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.

Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., et al. pi_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Black, K., Brown, N., Darpinian, J., Dhabalia, K., Driess, D., Esmail, A., Equi, M. R., Finn, C., Fusai, N., Galliker, M. Y., Ghosh, D., Groom, L., Hausman, K., ichter, b., Jakubczak, S., Jones, T., Ke, L., LeBlanc, D., Levine, S., Li-Bell, A., Mothukuri, M., Nair, S., Pertsch, K., Ren, A. Z., Shi, L. X., Smith, L., Springenberg, J. T., Stachowicz, K., Tanner, J., Vuong, Q., Walke, H., Walling, A., Wang, H., Yu, L., and Zhilinsky, U. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In Lim, J., Song, S., and Park, H.-W. (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 27–30 Sep 2025.

Cheang, C.-L., Chen, G., Jing, Y., Kong, T., Li, H., Li, Y., Liu, Y., Wu, H., Xu, J., Yang, Y., Zhang, H., and Zhu, M. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation, 2024. https://arxiv.org/abs/2410.06158.

Chen, T., Chen, Z., Chen, B., Cai, Z., Liu, Y., Li, Z., Liang, Q., Lin, X., Ge, Y., Gu, Z., Deng, W., Guo, Y., Nian, T., Xie, X., Chen, Q., Su, K., Xu, T., Liu, G., Hu, M., ang Gao, H., Wang, K., Liang, Z., Qin, Y., Yang, X., Luo, P., and Mu, Y. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. In *Forty-third International Conference on Machine Learning*, 2026.

Du, Y., Yang, S., Dai, B., Dai, H., Nachum, O., Tenenbaum, J., Schuurmans, D., and Abbeel, P. Learning universal policies via text-guided video generation. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 9156–9172. Curran Associates, Inc., 2023. doi: 10.52202/075280-0403.

Fei, S., Wang, S., Shi, J., Dai, Z., Cai, J., Qian, P., Ji, L., He, X., Zhang, S., Fei, Z., Fu, J., Gong, J., and Qiu, X. Libero-plus: A progressive robustness benchmark for visual-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 38574–38583, June 2026.

Guruprasad, P., Wang, Y., Chowdhury, S., Sikka, H., and Liang, P. P. Benchmarking vision, language, & action models in procedurally generated, open ended action environments. *arXiv preprint arXiv:2505.05540*, 2025.

Hu, Y., Guo, Y., Wang, P., Chen, X., Wang, Y.-J., Zhang, J., Sreenath, K., Lu, C., and Chen, J. Video prediction policy: A generalist robot policy with predictive visual representations. In Singh, A., Fazel, M., Hsu, D., Lacoste-Julien, S., Berkenkamp, F., Maharaj, T., Wagstaff, K., and Zhu, J. (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 24328–24346. PMLR, 13–19 Jul 2025.

Kim, M. J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E. P., Sanketi, P. R., Vuong, Q., Kollar, T., Burchfiel, B., Tedrake, R., Sadigh, D., Levine, S., Liang, P., and Finn, C. Openvla: An open-source vision-language-action model. In Agrawal, P., Kroemer, O., and Burgard, W. (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 06–09 Nov 2025.

Kim, M. J., Gao, Y., Lin, T.-Y., Lin, Y.-C., Ge, Y., Lam, G., Liang, P., Song, S., Liu, M.-Y., Finn, C., and Gu, J. Cosmos policy: Fine-tuning video models for visuomotor control and planning. In Vondrick, C., Hariharan, B., Raffel, C., Pinto, L., Yang, D., and Faust, A. (eds.), *International Conference on Learning Representations*, volume 2026, pp. 71531–71552, 2026.

Li, J., Guo, T., Ye, Y., Zhang, R., Chi, X., Sun, Q., Li, Y., Lou, Y., Huang, Y., Lu, Z., et al. Efficient-WAM: A 1b-parameter world-action model with low-cost future imagination. *arXiv preprint arXiv:2606.10040*, 2026a.

Li, L., Zhang, Q., Luo, Y., Yang, S., Wang, R., Han, F., Yu, M., Gao, Z., Xue, N., Zhu, X., et al. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026b.

Li, Z., Cheng, D., Wang, Y., Wang, S., Xu, X., Weng, L., Wang, J., and Wang, J. Light-WAM: Efficient world action models with state-fusion action decoding. *arXiv preprint arXiv:2606.08242*, 2026c.

Liu, B., Zhu, Y., Gao, C., Feng, Y., Liu, Q., Zhu, Y., and Stone, P. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Liu, S., Wu, L., Li, B., Tan, H., Chen, H., Wang, Z., Xu, K., Su, H., and Zhu, J. Rdt-1b: a diffusion foundation model for bimanual manipulation. In Yue, Y., Garg, A., Peng, N., Sha, F., and Yu, R. (eds.), *International Conference on Learning Representations*, volume 2025, pp. 29982–30009, 2025.

Luo, H., Zhang, W., Feng, Y., Zheng, S., Xu, H., Xu, C., Xi, Z., Fu, Y., and Lu, Z. Being-h0. 7: A latent world-action model from egocentric videos. *arXiv preprint arXiv:2605.00078*, 2026.

Ni, C., Zhou, X., Zhou, Y., Liu, J., Wang, X., Zhu, Z., Wang, Y., Deng, Q., Ye, Y., Li, H., Liu, Z., Lv, J., Wang, B., Zhao, G., Huang, G., Cao, M., and Mei, W. Gigaworld-policy: An efficient action-centered world–action model. In Favaro, P., Kukelova, Z., Maki, A., Rohrbach, A., Schindler, K., and Tombari, F. (eds.), *Computer Vision – ECCV 2026*, pp. 315–333, Cham, 2026. Springer Nature Switzerland. ISBN 978-3-032-37553-7. doi: 10.1007/978-3-032-37553-7_18.

Pai, J., Achenbach, L., Montesinos, V., Forrai, B., Mees, O., and Nava, E. mimic-video: Video-action models for generalizable robot control beyond vlas. *arXiv preprint arXiv:2512.15692*, 2025.

Shi, H., Li, W., Xie, B., Wang, Y., Zhou, R., Wang, T., Zhang, X., Luo, P., and Huang, G. Memoryvla++: Temporal modeling via memory and imagination in vision-language-action models. *arXiv preprint arXiv:2606.09827*, 2026a.

Shi, H., Xie, B., Liu, Y., Sun, L., Liu, F., Wang, T., Zhou, E., Fan, H., Zhang, X., and Huang, G. Memoryvla: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. In Vondrick, C., Hariharan, B., Raffel, C., Pinto, L., Yang, D., and Faust, A. (eds.), *International Conference on Learning Representations*, volume 2026, pp. 18567–18602, 2026b.

Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.-W., Chen, D., Yu, F., Zhao, H., Yang, J., Zeng, J., Wang, J., Zhang, J., Zhou, J., Wang, J., Chen, J., Zhu, K., Zhao, K., Yan, K., Huang, L., Feng, M., Zhang, N., Li, P., Wu, P., Chu, R., Feng, R., Zhang, S., Sun, S., Fang, T., Wang, T., Gui, T., Weng, T., Shen, T., Lin, W., Wang, W., Wang, W., Zhou, W., Wang, W., Shen, W., Yu, W., Shi, X., Huang, X., Xu, X., Kou, Y., Lv, Y., Li, Y., Liu, Y., Wang, Y., Zhang, Y., Huang, Y., Li, Y., Wu, Y., Liu, Y., Pan, Y., Zheng, Y., Hong, Y., Shi, Y., Feng, Y.,

Jiang, Z., Han, Z., Wu, Z.-F., and Liu, Z. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Wang, Y., Ding, P., Li, L., Cui, C., Ge, Z., Tong, X., Song, W., Zhao, H., Zhao, W., Hou, P., Huang, S., Tang, Y., Wang, W., Zhang, R., Liu, J., and Wang, D. Vla-adapter: An effective paradigm for tiny-scale vision-language-action model. *arXiv preprint arXiv:2509.09372*, 2025.

Wu, H., Jing, Y., Cheang, C., Chen, G., Xu, J., Li, X., Liu, M., Li, H., and Kong, T. Unleashing large-scale video generative pre-training for visual robot manipulation. In Kim, B., Yue, Y., Chaudhuri, S., Fragkiadaki, K., Khan, M., and Sun, Y. (eds.), *International Conference on Learning Representations*, volume 2024, pp. 10641–10662, 2024.

Ye, S., Ge, Y., Zheng, K., Gao, S., Yu, S., Kurian, G., Indupuru, S., Tan, Y. L., Zhu, C., Xiang, J., Malik, A. N., Lee, K., Liang, W., Arachchige, N. R., Gu, J., Xu, Y., Wang, G., Hu, F., Narayan, A., Bjorck, J., Wang, J., Kim, G., Niu, D., Zheng, R., Xie, Y., Wu, J., Wang, Q., Xu, D., Du, Y., Julian, R., Chebotar, Y., Reed, S., Kautz, J., Zhu, Y., Fan, L., and Jang, J. World action models are zero-shot policies. In *ICLR 2026 the 2nd Workshop on World Models: Understanding, Modelling and Scaling*, 2026.

Yuan, T., Dong, Z., Liu, Y., and Zhao, H. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026.

Zhang, C., Tong, J., Li, X., Wang, Y., and Li, H. Keep the future, drop the rollout: RIFT for world action models. *arXiv preprint arXiv:2608.11521*, 2026a.

Zhang, Y., Zhang, W., Qi, Z., Zhang, H., Lin, H., Zhang, J., Mu, Y., Yang, X., Zeng, W., and Jin, X. Imagewam: Do world action models really need video generation, or just image editing? *arXiv preprint arXiv:2606.19531*, 2026b.

Zhao, W., Jiang, H., Shi, X., Liu, L., Huang, F., Su, Z., Sui, W., and Wang, X. Faster-WAM: Efficient inference-time future conditioning for robust world action models. *arXiv preprint arXiv:2608.04404*, 2026.

Zhou, J., Ye, K., Liu, J., Ma, T., Wang, Z., QIU, R., Lin, K.-Y., Zhao, Z., and Liang, J. Exploring the limits of vision-language-action manipulation in cross-task generalization. In Belgrave, D., Zhang, C., Lin, H., Pascanu, R., Koniusz, P., Ghassemi, M., and Chen, N. (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 139899–139927. Curran Associates, Inc., 2025. doi: 10.52202/085713-4674.

Zhou, S., Du, Y., Chen, J., Li, Y., Yeung, D.-Y., and Gan, C. RoboDreamer: Learning compositional world models for robot imagination. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 61885–61896. PMLR, 21–27 Jul 2024.

Zitkovich, B., Yu, T., Xu, S., Xu, P., Xiao, T., Xia, F., Wu, J., Wohlhart, P., Welker, S., Wahid, A., Vuong, Q., Vanhoucke, V., Tran, H., Soricut, R., Singh, A., Singh, J., Sermanet, P., Sanketi, P. R., Salazar, G., Ryoo, M. S., Reymann, K., Rao, K., Pertsch, K., Mordatch, I., Michalewski, H., Lu, Y., Levine, S., Lee, L., Lee, T.-W. E., Leal, I., Kuang, Y., Kalashnikov, D., Julian, R., Joshi, N. J., Irpan, A., Ichter, B., Hsu, J., Herzog, A., Hausman, K., Gopalakrishnan, K., Fu, C., Florence, P., Finn, C., Dubey, K. A., Driess, D., Ding, T., Choromanski, K. M., Chen, X., Chebotar, Y., Carbajal, J., Brown, N., Brohan, A., Arenas, M. G., and Han, K. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In Tan, J., Toussaint, M., and Darvish, K. (eds.), *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 06–09 Nov 2023.

A. Appendix

A.1. Training Details

Tab. A1 lists our training configuration. The other baselines follow their official settings.

Table A1: Training configuration.

	LIBERO	RoboTwin 2.0	Real world
Optimizer		AdamW, $\beta = (0.9, 0.95)$	
Learning rate		1×10^{-4}	
Weight decay		1×10^{-2}	
LR schedule		cosine, 5% warmup	
Precision		BF16	
Batch size	128	1,024	512
Training steps	20k	30k	30k
Resolution	224×448	384×320	384×320

A.2. Per-Suite Simulation Results

Tab. A2 gives the detailed results of Tab. 2 on Data Efficiency and Task Generalization axes. For LIBERO each suite is the held-out fold of the four-fold cross-validation; RoboTwin 2.0 is reported under its clean and randomized scenes, and its data efficiency is measured at 10 demonstrations per task rather than 5.

Table A2: Success rate (%) behind the data-efficiency and task-generalization columns of Tab. 2.

	LIBERO					RoboTwin 2.0		
	Spatial	Object	Goal	Long	Avg.	Clean	Rand.	Avg.
Data efficiency, 5-shot								
$\pi_{0.5}$	92.1	94.5	89.7	73.6	87.5	–	–	–
VLA-Adapter	85.7	95.0	75.8	54.0	77.6	–	–	–
Lingbot-VA	59.0	96.3	88.6	70.2	78.5	–	–	–
FastWAM-Joint	**97.0**	**99.4**	**92.4**	77.2	91.5	–	–	–
FastWAM	84.4	93.4	77.6	57.4	78.2	–	–	–
Simple-WAM	**97.0**	**99.4**	89.6	**83.6**	**92.4**	–	–	–
Data efficiency, 10-shot								
$\pi_{0.5}$	95.4	96.6	91.0	82.8	91.5	35.5	32.3	33.9
VLA-Adapter	92.8	92.4	89.4	69.0	85.9	–	–	–
Lingbot-VA	89.1	95.8	89.2	76.4	87.6	4.1	3.6	3.9
FastWAM-Joint	**98.0**	**98.4**	97.2	94.2	97.0	36.8	33.2	35.0
FastWAM	87.4	95.2	88.0	83.4	88.5	9.6	0.1	4.8
Simple-WAM	97.8	**98.4**	**97.8**	**94.8**	**97.2**	**38.8**	**35.5**	**37.2**
Task generalization, w/o video								
$\pi_{0.5}$	18.2	7.2	12.2	0.0	9.4	2.8	2.1	2.5
VLA-Adapter	1.2	0.0	0.0	0.0	0.3	–	–	–
Lingbot-VA	29.7	12.4	18.8	0.0	15.2	0.0	0.0	0.0
FastWAM-Joint	13.4	1.6	**10.0**	0.0	6.3	6.4	4.3	5.4
FastWAM	0.0	8.4	0.0	0.0	2.1	4.6	2.4	3.5
Simple-WAM	**19.8**	**10.4**	**10.0**	0.0	**10.1**	**7.4**	**4.9**	**6.2**
Task generalization, w/ video								
$\pi_{0.5}$	18.2*	7.2*	12.2*	0.0*	9.4*	2.8*	2.1*	2.5*
VLA-Adapter	1.2*	0.0*	0.0*	0.0*	0.3*	–	–	–
Lingbot-VA	96.2	90.3	58.2	39.6	71.1	1.2	1.0	1.1
FastWAM-Joint	95.4	94.8	**57.6**	31.8	69.9	**47.7**	**46.8**	**47.3**
FastWAM	6.2	17.4	0.0	0.0	5.9	4.3	5.3	4.8
Simple-WAM	**97.2**	**99.2**	56.4	**41.4**	**73.6**	45.1	43.8	44.5

A.3. Per-Task Real-World Results

Tab. A3 reports the mean and standard deviation of real-world results, where T1–T4 are the four training tasks of Fig. 6 and the held-out task is evaluated without and with action-free video of it. A run is not a binary success: it receives the fraction of the sub-goals listed in Sec. 6.2 that the policy completes, and each cell averages thirty such runs. "Avg." averages the four task means and is the quantity quoted in Sec. 6.3.

Table A3: Real-world results per task.

	T1	T2	T3	T4	Avg.
In-distribution					
FastWAM	83.3±19.2	92.5±16.9	81.1±12.9	93.3±14.1	87.6
FastWAM-Joint	80.0±21.9	92.5±16.9	83.3±13.1	92.2±11.8	87.0
Simple-WAM	81.7±16.6	95.0±10.5	80.0±15.5	84.4±18.3	85.3
Environmental perturbation					
FastWAM	23.3±26.3	17.5±23.7	20.0±13.7	40.0±23.5	25.2
FastWAM-Joint	48.3±16.6	72.5±29.9	63.3±35.2	88.9±21.0	68.3
Simple-WAM	46.7±18.9	82.5±16.9	67.8±22.5	80.0±16.4	69.2
Data efficiency					
FastWAM	21.7±20.9	35.0±26.9	35.6±18.0	56.7±32.5	37.2
FastWAM-Joint	56.7±22.5	70.0±30.7	58.9±19.6	81.1±18.2	66.7
Simple-WAM	60.0±21.1	65.0±33.7	62.2±32.4	91.1±11.5	69.6
Task generalization, held-out task					
	w/o video	w/ video			
FastWAM	0.0±0.0	10.0±9.7			
FastWAM-Joint	3.3±5.4	68.9±20.2			
Simple-WAM	2.2±4.7	87.8±19.2			