# FORESIGHT WITHOUT SEEING: LATENT FUTURES FOR WORLD ACTION MODELS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

World Action Models (WAMs) connect visual prediction with robot control, yet supplying predictive context often comes at the cost of expensive future-video generation. Direct policies avoid this cost but lack an explicit interface for accessing future-indexed predictive information. Our key insight is that forecasting and rendering can be separated: a policy can access latent predictive computation without reconstructing future observations. Building on this insight, we introduce **ForeWAM**, a World Action Model that exposes and shapes latent predictive context for efficient control. Its core mechanism, **Future-KV**, performs a single Video DiT prefill over the current visual latent and noise-initialized future slots, then reuses the resulting key–value states throughout action denoising. Because predictive context alone does not ensure relevance to control, we further introduce dynamics registers supervised by latent actions from a frozen teacher during training, encouraging representations of interaction-induced transitions. Concentrating predictive computation in this reusable context supports a lightweight, single-layer action decoder. We evaluate ForeWAM on LIBERO, LIBERO-Plus, RoboCasa, and real-world manipulation tasks. Without additional policy-level embodied pretraining, ForeWAM improves RoboCasa success by 9.7 percentage points over Fast-WAM at the same budget of 50 demonstrations per task, reaching 59.2%. With a single-layer decoder, it achieves 77.6% success on LIBERO-Plus and reduces policy-query latency to 88.7 ms on an NVIDIA A800, delivering a 6.27× speedup over Fast-WAM. These results show that exposing and shaping latent predictive computation provides useful foresight for robust, efficient control without explicit future-video generation.

## 1 INTRODUCTION

World Action Models (WAMs) couple visual world modeling with robot control so that action generation can benefit from predicted scene evolution (Du et al., 2023; Hu et al., 2024; Li et al., 2025; Ye et al., 2026b; Zhu et al., 2025). Yet existing WAMs face a tension between predictive context and inference efficiency. Cascaded and joint WAMs explicitly expose predicted futures to the policy, but require costly iterative video generation (Du et al., 2023; Ye et al., 2026b; Bi et al., 2026). Direct-policy WAMs such as Fast-WAM (Yuan et al., 2026b) avoid this cost by acting directly on current-observation world-model features, but consequently provide no explicit interface for future-indexed predictive computation to reach the action pathway.

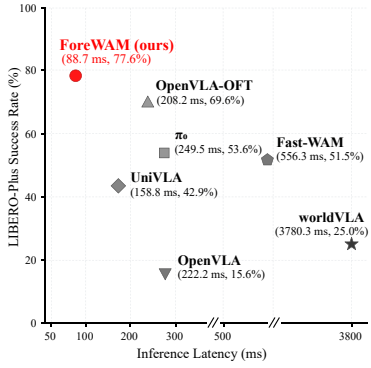



Figure 1: **OOD performance vs. latency on A800.**

We argue that this tension stems from conflating *forecasting* with *rendering*. A future observation is only the final output of predictive computation: before a clean future frame is produced, a Video DiT already forms hidden states at future temporal positions from the current visual context, instruction, and robot state. For control, the policy need not recover the future in pixel space; it primarily needs the predictive cues encoded along the way. This suggests a different interface between world modeling and action generation: expose the latent computation of forecasting



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.

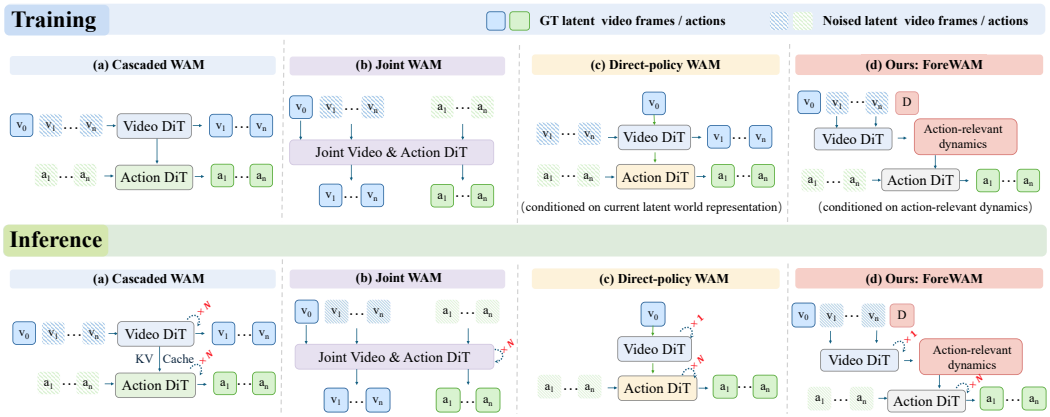



Figure 2: **World Action Model paradigms.** (a) Cascaded WAMs first generate future observations and then condition action prediction on them. (b) Joint WAMs generate future observations and actions within a unified generative process. (c) Direct-policy WAMs skip future rollout at inference and condition action prediction on a latent world representation extracted from the current observation. (d) ForeWAM retains direct action prediction while conditioning the action model on current-observation features, hidden future-slot K/V states, and dynamics registers trained with latent-action supervision. Hatched tokens denote noisy variables; future slots are stochastic internal states rather than observed future frames.

itself, rather than its rendered output. In other words, a policy may obtain *foresight without seeing the future*.

Based on this insight, we introduce **ForeWAM**, a **Foresight-without-Seeing World Action Model**. Its core mechanism, **Future-KV**, exposes future-indexed predictive states without generating future video. ForeWAM appends noise-initialized future slots to the current visual latent and performs a single Video DiT prefill. The resulting key–value states from the current and future positions are cached and reused throughout action denoising. The action policy can therefore access predictive context computed by the video model, while avoiding iterative future-video denoising and decoding.

Yet predictive context is not necessarily *action-relevant* context. Video prediction models broad scene evolution, including many changes that may be irrelevant to manipulation, whereas action generation depends particularly on interaction-induced transitions and task progress. We therefore introduce **dynamics registers** and supervise them with latent actions extracted by a frozen LaWM teacher (Chen et al., 2026a) from demonstrated current–future observation pairs. This training-only signal shapes the exposed predictive representation toward dynamics that matter for control. The distinction is strongly reflected in our ablation: on LIBERO-Plus, Future-KV alone improves the matched baseline from 53.6% to 58.5%, and latent-action supervision alone reaches 54.8%, whereas combining them yields 76.7% (Table 1).

With predictive reasoning concentrated in the reusable video-side context, the action branch can be substantially simplified. The Video DiT is evaluated only once per policy query, whereas the Action DiT runs repeatedly throughout action denoising; shifting computation toward the former is therefore particularly beneficial for inference. We find that a single-layer Action DiT reaches 77.6% on LIBERO-Plus, compared with 76.7% for a 30-layer decoder, while a one-layer Fast-WAM reaches only 57.9% (Table 2). This suggests that the shallow decoder is enabled by the richer predictive context, rather than action decoding being intrinsically easy.

Experiments across LIBERO, LIBERO-Plus, RoboCasa, and real-world manipulation validate this design. Without additional policy-level embodied pretraining, ForeWAM achieves 59.2% success on RoboCasa with only 50 demonstrations per task, outperforming Fast-WAM by 9.7 percentage points under the same demonstration budget. It retains near-saturated performance on standard LIBERO while substantially improving robustness under the distribution shifts of LIBERO-Plus. Meanwhile, ForeWAM reduces policy-query latency from 556.3 ms to 88.7 ms on an NVIDIA A800, a 6.27× speedup over Fast-WAM (Figure 1). Together, these results show that useful world-model foresight need not be materialized as future video: it can instead be exposed and shaped directly in latent space for efficient control.

## 2 Related Work

**Vision-language-action policies.** VLA models map visual observations and language instructions to executable robot actions (Brohan et al., 2022; 2023; Kim et al., 2024; Liu et al., 2025; Team et al., 2024; Yang et al., 2026), with many recent models adapting pretrained vision-language backbones for control (Black et al., 2024; Intelligence et al., 2025; Zhao et al., 2025). Diffusion and flow objectives support multimodal continuous action generation (Black et al., 2024; Chi et al., 2025; Lipman et al., 2022), while large-scale robot pretraining can improve transfer across tasks and embodiments (Bjorck et al., 2025; Bu et al., 2025; Zheng et al., 2026). However, the action-prediction objective alone does not explicitly require learning predictive representations of how the environment evolves during interaction, motivating the integration of world modeling into policy learning.

**World-action models.** World Action Models (WAMs) couple world prediction with action generation through cascaded or joint architectures. Cascaded approaches follow an *imagine-then-act* pipeline, extracting actions from generated future visual observations as intermediate plans (Du et al., 2023; 2024), geometric representations (Bharadhwaj et al., 2024; Ko et al., 2024; Xu et al., 2024; Zhi et al., 2025) or predictive latent features (Hu et al., 2024; Huang et al., 2024; Lou et al., 2026; Yan et al., 2026b). Joint WAMs learn future states and actions together. Autoregressive variants organize them within a unified temporal sequence (Cen et al., 2025a;b; Cheang et al., 2024; Wu et al., 2024; Team et al., 2026). Diffusion- and flow-based variants use either unified prediction backbones (Chen et al., 2026b; Guo et al., 2024; Kim et al., 2026; Shen et al., 2026; Ye et al., 2026b; Zhao et al., 2026; Zheng et al., 2025; Zhu et al., 2025) or coordinated modality-specific streams, coupling their predictive streams through shared attention (Bi et al., 2026; Lyu et al., 2026; Won et al., 2025; Yang et al., 2025; Yuan et al., 2026a), intermediate hidden states (Li et al., 2026a; Ma et al., 2026), or shared representations (Li et al., 2025). Iterative future generation and tightly coupled world–action computation can incur substantial inference overhead. Direct-policy WAMs, including Fast-WAM and GigaWorld-Policy, retain world-modeling supervision during training but use action-only inference (Ye et al., 2026a; Yuan et al., 2026b). Concurrent work explores complementary forms of predictive supervision: Enfold (Zeng et al., 2026) distills world-generator states into representations predicted from current observations and language, while Robust-WAM (Yan et al., 2026a) aligns action-stream queries with future visual semantics. ForeWAM retains noisy future slots at inference and conditions action denoising on their hidden K/V states from a single Video DiT prefill, together with dynamics registers supervised by LaWM latent-action targets (Chen et al., 2026a), without generating future video.

## 3 From Predictive Context to Actionable Foresight

To understand what an action policy actually needs from a video world model, we conduct a controlled study on LIBERO-Plus using Fast-WAM retrained with Wan2.1 as the matched direct-policy baseline (Yuan et al., 2026b). We focus on two questions: whether future-indexed latent context is sufficient for control, and how the resulting context changes the capacity required by the action pathway. The observations below motivate the design of ForeWAM in Section 4.

### 3.1 Useful Foresight Requires Both Access and Shaping

We first examine how predictive information should be exposed to the action policy. The matched Fast-WAM baseline conditions action generation on current-observation world-model features and achieves 53.6% average success. Providing additional future-indexed context through Future-KV improves this result to 58.5% (Table 1), despite requiring no explicitly generated future observation at inference. This suggests that internal states associated with future temporal positions provide useful context beyond current-observation features alone.

Table 1: **Predictive-context design study on LIBERO-Plus (%).** Future-KV exposes future-indexed context, while LA supervision shapes interaction-relevant dynamics. Full results: Appendix D.

| LA | Future-KV | Avg |
|---|---|---|
| × | × | 53.6 |
| ✓ | × | 54.8 |
| × | ✓ | 58.5 |
| ✓ | ✓ | **76.7** |

However, access to such context is clearly not sufficient. Video prediction models scene evolution broadly, whereas control depends particularly on interaction-induced transitions and task progress.

Consistent with this distinction, latent-action (LA) supervision alone provides only a marginal gain, from 53.6% to 54.8%, while Future-KV alone reaches 58.5%. Strikingly, combining the two increases success to 76.7%, outperforming the stronger individual component by 18.2 percentage points. The same combination improves all seven LIBERO-Plus perturbation categories over either component alone (Table 9 in Appendix D).

These results reveal two complementary requirements for useful foresight. Future-KV determines *what predictive context is accessible* to the action pathway, whereas LA supervision shapes *what information that context emphasizes*. In other words, predictive context becomes substantially more useful for control when future-indexed representations are both **exposed** and **shaped toward interaction-relevant dynamics**.

### 3.2 Rich Predictive Context Enables Lightweight Action Decoding

The representation supplied by the video model also determines how much computation remains necessary in the action pathway. The Video DiT constructs context once per policy query, whereas the Action DiT is evaluated repeatedly throughout action denoising. If the video-side context already captures the dynamics needed for control, a deep action decoder may therefore be redundant.

Table 2: Action-expert depth comparison. LIBERO-Plus success rate (%), without additional policy-level embodied pretraining. All configurations retain the 30-layer Wan2.1 Video DiT; action-layer mappings and initializations follow Appendix A. Bold marks the best result in each column.

| Action DiT depth | Camera | Robot | Language | Light | Background | Noise | Layout | Avg |
|---|---|---|---|---|---|---|---|---|
| Fast-WAM / Wan2.1 (30 layers) | 37.0 | 36.5 | 64.1 | 81.5 | 49.3 | 50.1 | 63.7 | 53.6 |
| Fast-WAM / Wan2.1 (1 layer) | 44.9 | 38.0 | 71.5 | 81.6 | 51.3 | 54.2 | 69.0 | 57.9 |
| ForeWAM (30 layers) | 64.6 | 61.6 | **86.6** | 95.3 | 74.1 | 77.7 | 81.7 | 76.7 |
| ForeWAM (12 layers) | **67.7** | **63.7** | 86.0 | 90.8 | 71.8 | 80.0 | 80.5 | 76.9 |
| ForeWAM (6 layers) | 61.5 | 61.4 | 79.6 | 92.2 | 76.4 | 78.6 | 78.4 | 74.7 |
| ForeWAM (1 layer) | 61.7 | 56.8 | 85.1 | **95.8** | **76.5** | **89.1** | **83.0** | **77.6** |

We test this hypothesis by retaining all 30 layers of the Video DiT while varying the Action DiT depth. ForeWAM achieves 76.7%, 76.9%, 74.7%, and 77.6% average success with 30, 12, 6, and 1 action layers, respectively (Table 2). Increasing action depth provides no monotonic benefit; remarkably, the single-layer decoder achieves the highest overall success among these variants.

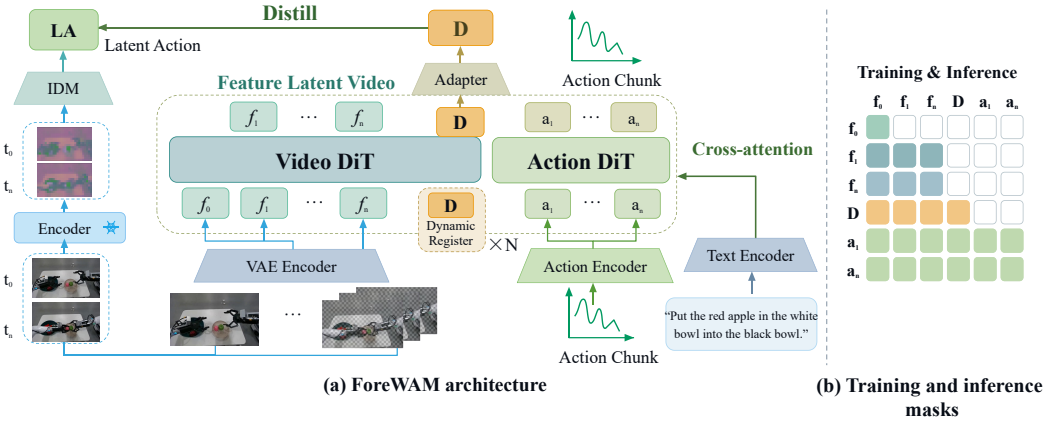
This result is not simply explained by action decoding being intrinsically easy. Reducing the matched Fast-WAM action branch from 30 layers to one improves its success only from 53.6% to 57.9%. At the same one-layer depth, ForeWAM reaches 77.6%, a 19.7-point advantage. The lightweight decoder is therefore effective in conjunction with the richer predictive context, rather than because the original action pathway is unnecessarily deep.

Together, these observations suggest a clear division of computation: the video branch should expose and organize predictive dynamics into an action-relevant representation, while the repeatedly evaluated action branch can remain lightweight. ForeWAM instantiates this principle with Future-KV, latent-action-supervised dynamics registers, and a shallow Action DiT, which we detail next.

## 4 ForeWAM

Guided by the observations in Section 3, we introduce **ForeWAM**, a direct-policy World Action Model that exposes predictive context from a video world model without explicitly generating future video. ForeWAM follows a simple decomposition. **Future-KV** extracts future-indexed latent context through a single Video DiT prefill; learnable **dynamics registers** shape this context toward interaction-relevant transitions through latent-action supervision; and a lightweight Action DiT reuses the resulting predictive context throughout action denoising. Figure 3 illustrates the overall architecture.

**Problem formulation.** Given synchronized multi-camera observations $o$, a language instruction $l$, and proprioceptive state $p$, the policy predicts an executable action chunk $a_{1:H} \in \mathbb{R}^{H \times d_{\text{act}}}$. We retain the direct-policy interface $p_\theta(a_{1:H} \mid o, l, p)$ (Yuan et al., 2026b). Let $z_{\text{cur}}$ denote the current-




(a) ForeWAM architecture (b) Training and inference masks

Figure 3: **ForeWAM.** (a) Architecture and training pipeline. A supervision pass uses demonstrated futures for video and latent-action learning, while an independent action-conditioning pass uses pure-noise future slots to construct the predictive context consumed by the action policy. At inference, only the latter is required: one Video DiT prefill supplies cached K/V throughout action denoising, without future observations or the latent-action teacher. (b) Structured attention mask shared by training and inference. Rows are queries and columns are keys; colored cells indicate allowed attention. Here $f_0 = C$, $f_{1:n} = F$, and $a_{1:n} = A$. Future slots attend bidirectionally to one another.

frame latent encoded by the video VAE, and $z_{1:T}$ the demonstrated video latents used for training supervision. Future observations and latent-action teacher outputs are used only during training.

### 4.1 Latent Foresight through Future-KV

The first component exposes predictive computation associated with future temporal positions without materializing future observations. For the action-conditioning pathway, we concatenate the clean current latent with Gaussian future slots:

$$z^{\text{init}} = [z_{\text{cur}}; \epsilon_F], \qquad \epsilon_F \sim \mathcal{N}(0, I), \tag{1}$$

where $[\cdot;\cdot]$ denotes temporal concatenation. The Video DiT processes this sequence once at $\sigma = 1.0$, conditioned on the instruction and proprioception. We retain the resulting hidden key–value states and reuse them throughout action denoising. We refer to this interface as **Future-KV**.

Importantly, the future slots contain no observed future information. Instead, they provide future temporal positions at which a video model trained with future supervision can construct contextual states from the available inputs. ForeWAM exposes these internal states directly to the action pathway rather than iteratively denoising them into valid future frames. Thus, predictive context is computed once and reused, while explicit future-video generation is avoided.

### 4.2 Shaping Predictive Context with Latent Actions

Future-KV provides access to future-indexed context, but does not by itself specify which aspects of scene evolution should be emphasized for control. We therefore introduce learnable dynamics registers $D = \{D_i\}_{i=1}^{N_D}$ into the Video DiT and supervise their pooled representation with a latent-action target extracted from demonstrated transitions.

**Latent-action supervision.** We use a frozen LaWM teacher (Chen et al., 2026a) to encode a demonstrated transition. Let $o_{\text{ext}}$ and $o^{+}_{\text{ext}}$ denote the current and future observations from the external-camera view. A frozen visual encoder $f_{\text{vis}}$ and latent-action extraction mapping $E_{\text{IDM}}$ produce

$$\begin{aligned} u_{\text{cur}} = f_{\text{vis}}(o_{\text{ext}}), \quad u_{\text{fut}} = f_{\text{vis}}(o^{+}_{\text{ext}}), \\ z_{\text{LA}} = E_{\text{IDM}}(u_{\text{cur}}, u_{\text{fut}}). \end{aligned} \tag{2}$$

The teacher representation $z_{\text{LA}}$ describes the demonstrated transition and is used only as a training target. It is distinct from the video VAE latents $z_{1:T}$ and is never supplied directly to the action

decoder. A trainable projection $g_\psi$ aligns the pooled dynamics-register representation with this latent-action target.

**Routing predictive context.** ForeWAM organizes information flow among current-frame tokens $C$, future-slot tokens $F$, dynamics registers $D$, and action tokens $A$ using the structured attention mask in Figure 3(b). Within the video branch, current-frame tokens attend only to $C$, while future-slot tokens attend to $C$ and all tokens in $F$, with bidirectional attention across future slots. Dynamics registers attend to $C$, $F$, and $D$, allowing them to aggregate current and future-indexed information without feeding back into the visual tokens.

The resulting video-side context is exposed to the action branch through cached K/V. At each action-denoising step, action queries attend to cached states from $C$, $F$, and $D$, together with the current action-token K/V from $A$. Both branches additionally receive text embeddings and projected proprioceptive conditioning, which are omitted from the mask diagram for clarity.

### 4.3 Lightweight Action Decoding from Cached Predictive Context

As shown in Section 3.2, once the video branch provides rich predictive context, the action pathway can be substantially simplified. We therefore use a single-layer Action DiT that reads the final Video DiT layer, $L = 30$.

A single video prefill produces the dynamics-register states and final-layer K/V cache:

$$(D, K, V) = \mathrm{Prefill}_\phi(z^{\mathrm{init}}, l, p), \tag{3}$$

where $K = K^{(L)}$ and $V = V^{(L)}$ contain the final-layer states of the current-frame, future-slot, and dynamics-register tokens. These features condition action prediction as

$$p_\theta(a_{1:H} \mid o, l, p, D, K, V). \tag{4}$$

The dependence of $(D, K, V)$ on $(o, l, p, \epsilon_F)$ is implicit.

At inference, the Video DiT is evaluated only once for each action-chunk prediction. During subsequent action denoising, the single-layer Action DiT repeatedly reads the same cached predictive context together with the evolving action tokens. ForeWAM therefore concentrates visual and predictive computation in a reusable video-side representation while keeping repeated action-side computation lightweight.

### 4.4 Training and Inference

Training uses two Video DiT forward passes with shared parameters but independent hidden states and K/V caches. The *supervision pass* receives noised demonstrated futures and provides the video-prediction and latent-action objectives. The independent *action-conditioning pass* instead receives freshly sampled pure-noise future slots at $\sigma = 1.0$, matching the inference-time input in Eq. 1, and produces $(D, K, V)$ for the action objective. This separation ensures that demonstrated future observations supervise the model during training without being directly exposed to the action-conditioning pathway.

**Video flow matching.** The video branch learns demonstrated visual dynamics through continuous flow matching (Lipman et al., 2022). Writing the target video latents as $z = z_{1:T}$, we sample Gaussian noise $\epsilon_z$ and flow time $\tau \in (0, 1)$ to construct $z_\tau = (1 - \tau)z + \tau\epsilon_z$, while keeping the current-frame latent clean. The video velocity predictor $v_\phi$ is optimized with

$$\mathcal{L}_{\mathrm{video}} = \mathbb{E}_{z,\epsilon_z,\tau}\left[\left\|v_\phi(z_\tau, \tau \mid o, l, p) - (\epsilon_z - z)\right\|_2^2\right]. \tag{5}$$

Let $D^{\mathrm{sup}}$ denote the dynamics-register states produced by this supervision pass.

**Action flow matching.** Let $a = a_{1:H}$ be the demonstrated action chunk. We sample Gaussian noise $\epsilon_a$ and action flow time $s \in (0, 1)$, and form $a_s = (1 - s)a + s\epsilon_a$. Conditioned on the predictive context $(D, K, V)$ from the independent pure-noise prefill, the Action DiT predicts the action velocity:

$$\mathcal{L}_{\mathrm{action}} = \mathbb{E}\left[\left\|v_\theta(a_s, s \mid o, l, p, D, K, V) - (\epsilon_a - a)\right\|_2^2\right]. \tag{6}$$

The expectation is over training examples, action noise and flow time, and fresh future-slot noise $\epsilon_F$. Crucially, the noised demonstrated future $z_\tau$, the supervision-pass registers $D^{\text{sup}}$, and their K/V states never enter this action-conditioning forward pass.

**Latent-action supervision.** The projection $g_\psi$ aligns the mean-pooled supervision-pass dynamics registers with the frozen latent-action target from Eq. 2:

$$\mathcal{L}_{\text{LA}} = \left\| g_\psi\left(\frac{1}{N_D}\sum_{i=1}^{N_D} D_i^{\text{sup}}\right) - \text{sg}(z_{\text{LA}})\right\|_2^2, \tag{7}$$

where sg stops gradients through the frozen teacher. The complete training objective is

$$\mathcal{L} = \mathcal{L}_{\text{video}} + \mathcal{L}_{\text{action}} + \lambda_{\text{LA}}\mathcal{L}_{\text{LA}}. \tag{8}$$

Action gradients propagate through the video-to-action K/V interface, allowing the predictive context to be optimized jointly for control.

**Inference.** At inference, the supervision pass, demonstrated future observations, and latent-action teacher are all removed. ForeWAM encodes the current observation, samples pure-noise future slots, performs a single Video DiT prefill, and reuses the resulting $(D, K, V)$ cache throughout action denoising. No future video is generated or decoded.

# 5 Experiments

## 5.1 Experimental Setup

We evaluate ForeWAM on LIBERO, RoboCasa, the OOD LIBERO-Plus benchmark, and real-world manipulation tasks, using success rate as the primary metric.

**Benchmarks and protocols.** We train on the original LIBERO demonstrations (500 per suite) and evaluate on its four standard suites and the seven LIBERO-Plus perturbation categories, without using augmented LIBERO-Plus training data (Liu et al., 2023; Fei et al., 2025). We evaluate RoboCasa on 24 static kitchen tasks over three seeds (3,600 trials), using successful human demonstrations without additional MimicGen data.

**Real-world experiments.** We jointly train on 125 teleoperated demonstrations (71,702 frames) from four tabletop tasks using a 6-DoF AgileX Piper arm with third-person and wrist RGB cameras. ID evaluation uses 10 trials per task with matched randomized object placements; OOD evaluation perturbs object attributes in pen placement and drawer manipulation (Figure 4). Full benchmark protocols and task definitions appear in Appendix F; additional real-world details are in Appendix C.

## 5.2 Main Results

Table 3: Results on RoboCasa.

| Method | P.T. | Demos/ task | **SR (%)** |
|---|---|---|---|
| GR00T-N1 (Bjorck et al., 2025) | ✓ | 300 | 49.6 |
| DP-VLA (Han et al., 2024) | ✓ | 3000 | 57.3 |
| GR00T-N1 + DreamGen (Jang et al., 2025) | ✓ | 300 | 57.6 |
| GR00T-N1 + DUST (Won et al., 2025) | ✓ | 300 | 58.5 |
| UVA (Li et al., 2025) | × | 50 | 50.0 |
| Fast-WAM (Yuan et al., 2026b) | × | 50 | 49.5 |
| **ForeWAM** | × | 50 | **59.2** |

P.T.: policy-level embodied pretraining (Appendix A).

Table 4: Results on LIBERO.

| Method | P.T. | **Avg.** |
|---|---|---|
| OpenVLA (Kim et al., 2024) | ✓ | 76.5 |
| WorldVLA (Cen et al., 2025b) | ✓ | 81.8 |
| $\pi_0$-Fast (Pertsch et al., 2025) | ✓ | 85.5 |
| $\pi_0$ (Black et al., 2024) | ✓ | 94.1 |
| UniVLA (Bu et al., 2025) | ✓ | 95.2 |
| $\pi_{0.5}$ (Intelligence et al., 2025) | ✓ | 96.9 |
| CogVLA (Li et al., 2026b) | ✓ | 97.4 |
| Fast-WAM (Yuan et al., 2026b) | × | <u>97.6</u> |
| **ForeWAM** | × | **97.9** |

**Results on RoboCasa.** ForeWAM achieves strong manipulation performance with a limited demonstration budget (Table 3). With 50 demonstrations per task and no additional policy-level embodied pretraining, it reaches 59.2% average success across 24 kitchen manipulation tasks, compared with 49.5% for Fast-WAM under the same demonstration budget and pretraining setting, a gain of 9.7 percentage points. ForeWAM also exceeds GR00T-N1 (Bjorck et al., 2025) by 9.6 percentage points while using one-sixth as many demonstrations per task (50 versus 300), achieving the highest success rate among the methods listed in the table.

**Results on LIBERO & LIBERO-Plus.** ForeWAM preserves the near-saturated performance of Fast-WAM on standard LIBERO while substantially improving robustness under the distribution shifts evaluated by LIBERO-Plus (Tables 4 and 5). On LIBERO, ForeWAM achieves 97.9% average success compared with 97.6% for Fast-WAM, with neither method using additional policy-level embodied pretraining. Suite-level results are provided in Appendix E. On the full LIBERO-Plus benchmark, the improvement is much larger: ForeWAM raises average success from 51.5% to 77.6%, a gain of 26.1 percentage points, and exceeds OpenVLA-OFT by 8.0 points. It leads five of the seven perturbation categories, including lighting (95.8%) and sensor noise (89.1%). These results show that the main gain over Fast-WAM lies in robustness to the evaluated distribution shifts, while standard manipulation performance remains high.

Table 5: Success rates (%) on seven LIBERO-Plus perturbation categories.

| Method | Params | P.T. | Camera | Robot | Language | Light | Background | Noise | Layout | Avg |
|---|---|---|---|---|---|---|---|---|---|---|
| OpenVLA (Kim et al., 2024) | 7B | ✓ | 0.8 | 3.5 | 23.0 | 8.1 | 34.8 | 15.2 | 28.5 | 15.6 |
| OpenVLA-OFT | 7.5B | ✓ | 56.4 | 31.9 | <u>79.5</u> | <u>88.7</u> | **93.3** | 75.8 | <u>74.2</u> | <u>69.6</u> |
| $\pi_0$ (Black et al., 2024) | 3.3B | ✓ | 13.8 | 6.0 | 58.8 | 85.0 | <u>81.4</u> | <u>79.0</u> | 68.9 | 53.6 |
| $\pi_0$-Fast (Pertsch et al., 2025) | 3.3B | ✓ | **65.1** | 21.6 | 61.0 | 73.2 | 73.2 | 74.4 | 68.8 | 61.6 |
| UniVLA (Bu et al., 2025) | 7B | ✓ | 1.8 | <u>46.2</u> | 69.6 | 69.0 | 81.0 | 21.2 | 31.9 | 42.9 |
| WorldVLA (Cen et al., 2025b) | 7B | ✓ | 0.1 | 27.9 | 41.6 | 43.7 | 17.1 | 10.9 | 38.0 | 25.0 |
| Fast-WAM (Yuan et al., 2026b) | 6B | × | 16.4 | 44.5 | 68.9 | 78.2 | 53.7 | 37.7 | 60.7 | 51.5 |
| ForeWAM | 1.5B | × | <u>61.7</u> | **56.8** | **85.1** | **95.8** | 76.5 | **89.1** | **83.0** | **77.6** |

**Results on Real-World Tasks.** ForeWAM achieves the highest average success rate across the four in-distribution (ID) real-world tasks (Figure 4). Without additional policy-level embodied pretraining, it reaches 92.5% success, outperforming Fast-WAM by 42.5 percentage points and $\pi_0$ by 5.0 points. The largest gain over Fast-WAM occurs on placing blocks in the drawer and closing it, where success increases from 0% to 90%. Under object-attribute perturbations in the two OOD tasks, ForeWAM achieves 40.0% average success, compared with 10.0% for Fast-WAM and 15.0% for Cosmos Policy (Kim et al., 2026). While ForeWAM outperforms both $\pi_0$ and $\pi_{0.5}$ on ID tasks, all three evaluated WAMs remain below $\pi_0$ (60.0%) and $\pi_{0.5}$ (45.0%) in average OOD success. This shared gap may partly reflect differences in pretraining, as ForeWAM, Fast-WAM, and Cosmos Policy do not undergo the large-scale policy-level robot pretraining used by these two policies.

### 5.3 Analysis

**Action-to-image Attention.** Figure 5 visualizes the action-to-image attention maps of Fast-WAM and ForeWAM across two manipulation examples. ForeWAM exhibits more localized attention around interaction-relevant regions, including manipulated objects, target receptacles, and contact areas, particularly in wrist views, while Fast-WAM shows more diffuse responses and stronger activations in less task-relevant regions. This indicates that ForeWAM provides more interaction-focused visual context to the action pathway.

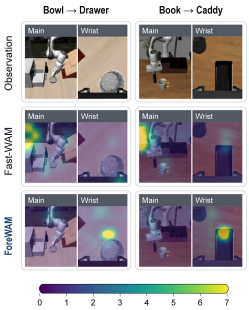


Figure 5: Attention maps.

**Inference Efficiency.** Table 6 compares overall and component-wise inference latency on LIBERO-10 using a single NVIDIA A800 GPU and 10 action-denoising steps. ForeWAM reduces overall latency from 556.3 ms to 88.7 ms, yielding a 6.27× speedup over Fast-WAM.

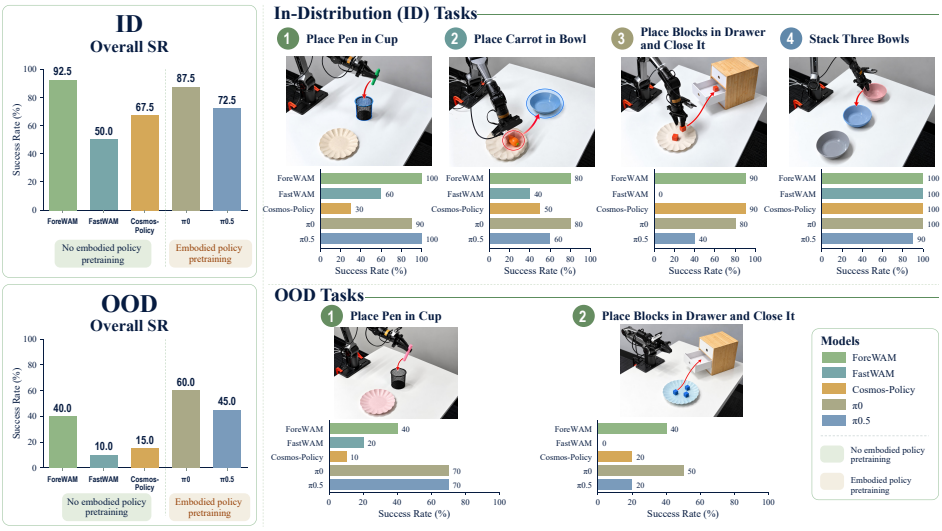



Figure 4: Real-world task success rates (%). Top: four in-distribution tasks. Bottom: OOD evaluation with perturbed object attributes in pen placement and drawer manipulation. Left panels report the mean success rate across the tasks in each setting; right panels show task examples and per-task results. ForeWAM and Fast-WAM use no additional policy-level embodied pretraining.

The largest reduction occurs in action generation, which decreases from 470.6 ms to 21.7 ms, while visual processing remains comparable (46.4 ms versus 47.7 ms). Thus, the measured speedup primarily comes from the action-generation stage. ForeWAM's single Video DiT prefill supplies cached Future-KV features to the one-layer Action DiT throughout denoising, retaining latent future conditioning with a lightweight action pathway. Together with the success-rate results above, these measurements show that ForeWAM combines strong manipulation performance with low inference latency. Appendix B provides details of the latency comparison.

Table 6: Inference latency (ms) on LIBERO-10.

| Stage | Fast-WAM | ForeWAM |
|---|---|---|
| VAE Enc. | 12.7 | **10.0** |
| Visual | 47.7 | **46.4** |
| Action | 470.6 | **21.7** |
| Other | 25.3 | **10.6** |
| Overall | 556.3 | **88.7** |

## 6 Limitations and Discussion

Our evaluation covers LIBERO, LIBERO-Plus, RoboCasa, and real-world tabletop manipulation. However, the real-world experiments are limited to a single robot platform, four tasks, and a small number of trials. Generalization under the evaluated object-attribute shifts remains challenging, while transfer to other embodiments and broader task distributions has yet to be established. Extending the evaluation to more diverse robot platforms, interaction dynamics, and longer-horizon tasks is an important direction for future work.

## 7 Conclusion

We introduced ForeWAM, a direct-policy World Action Model that provides predictive context for control without explicit future-video generation. ForeWAM combines Future-KV with latent-action-supervised dynamics registers: a single Video DiT prefill supplies cached context that a lightweight Action DiT reuses throughout action denoising. Future observations and the latent-action teacher are required only during training.

Experiments across simulation benchmarks and real-world manipulation tasks demonstrate the effectiveness of this design, while component ablations show the complementary benefits of Future-KV and latent-action supervision. With a single-layer action decoder, ForeWAM achieves 77.6% success on LIBERO-Plus and a policy-query latency of 88.7 ms on an NVIDIA A800. These results support exposing and shaping latent predictive context as an effective approach to robust, efficient control without rendering future observations.

### AI USE STATEMENT

In this work, generative AI tools were used to assist with preliminary literature search and review, collaborative manuscript drafting and revision, implementation of the ForeWAM code, and figure preparation. The authors reviewed all AI-assisted contributions and take responsibility for the final manuscript, code, and figures.

## REPRODUCIBILITY STATEMENT

We describe the ForeWAM architecture, training objectives, and inference procedure in Section 4. Appendix A provides implementation and training details, including model initialization, hyperparameters, and the use of the frozen latent-action teacher. Section 5.1 and Appendix F describe the datasets and benchmark evaluation protocols, while Appendix C details the real-world platform, data collection, task definitions, and evaluation settings. Appendix B specifies the hardware and timing protocol used for the latency comparison. Detailed component ablations and suite-level LIBERO results are provided in Appendices D and E, respectively.

## REFERENCES

Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *European Conference on Computer Vision*, pp. 306–324. Springer, 2024.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35101–35113, 2026.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. $\pi_0$: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.

Jun Cen, Siteng Huang, Yuqian Yuan, Kehan Li, Hangjie Yuan, Chaohui Yu, Bohan Hou, Yuming Jiang, Jiayan Guo, Xin Li, et al. Rynnvla-002: A unified vision-language-action and world model. *arXiv preprint arXiv:2511.17502*, 2025a.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. Worldvla: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025b.

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024.

Jialei Chen, Kai Wang, Kang Chen, Shuaihang Chen, Feng Gao, Wenhao Tang, Zhiyuan Li, Weilin Liu, Zhuyu Yao, Boxun Li, et al. Lawam: Latent world action models for efficient dynamics-aware robot policies. *arXiv preprint arXiv:2606.15768*, 2026a.

Jiayi Chen, Wenxuan Song, Pengxiang Ding, Ziyang Zhou, Han Zhao, Barrett Tang, Donglin Wang, and Haoang Li. Unified diffusion vla: Vision-language-action model via joint discrete denosing diffusion process. In *International Conference on Learning Representations*, volume 2026, pp. 139291–139311, 2026b.

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

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. *Advances in neural information processing systems*, 36:9156–9172, 2023.

Yilun Du, Sherry Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B Tenenbaum, Leslie Kaelbling, et al. Video language planning. In *International Conference on Learning Representations*, volume 2024, pp. 31138–31155, 2024.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

Yanjiang Guo, Yucheng Hu, Jianke Zhang, Yen-Jen Wang, Xiaoyu Chen, Chaochao Lu, and Jianyu Chen. Prediction with action: Visual policy learning via joint denoising process. *Advances in Neural Information Processing Systems*, 37:112386–112410, 2024.

ByungOk Han, Jaehong Kim, and Jinhyeok Jang. A dual process vla: Efficient robotic manipulation leveraging vlm. *arXiv preprint arXiv:2410.15549*, 2024.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. *arXiv preprint arXiv:2412.14803*, 2024.

Shuaiyi Huang, Mara Levy, Zhenyu Jiang, Anima Anandkumar, Yuke Zhu, Linxi Fan, De-An Huang, and Abhinav Shrivastava. Ardup: Active region video diffusion for universal policies. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8465–8472. IEEE, 2024.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

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

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, et al. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026.

Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B Tenenbaum. Learning to act from actionless videos through dense correspondences. In *International Conference on Learning Representations*, volume 2024, pp. 40938–40958, 2024.

Runze Li, Hongyin Zhang, Junxi Jin, Qixin Zeng, Zifeng Zhuang, Yiqi Tang, Shangke Lyu, and Donglin Wang. World-value-action model: Implicit planning for vision-language-action systems. *arXiv preprint arXiv:2604.14732*, 2026a.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model, 2025. *arXiv preprint arXiv:2503.00200*, 2025.

Wei Li, Renshan Zhang, Rui Shao, Jie He, and Liqiang Nie. Cogvla: Cognition-aligned vision-language-action models via instruction-driven routing & sparsification. *Advances in neural information processing systems*, 38:137646–137675, 2026b.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

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

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. In *International Conference on Learning Representations*, volume 2025, pp. 29982–30009, 2025.

Yunfan Lou, Xiaowei Chi, Xiaojie Zhang, Zezhong Qian, Chengxuan Li, Rongyu Zhang, Yaoxu Lyu, Guoyu Song, Chuyao Fu, Haoxuan Xu, et al. Mask world model: Predicting what matters for robust robot policy learning. *arXiv preprint arXiv:2604.19683*, 2026.

Jiangran Lyu, Kai Liu, Xuheng Zhang, Haoran Liao, Yusen Feng, Wenxuan Zhu, Tingrui Shen, Jiayi Chen, Jiazhao Zhang, Yifei Dong, et al. Lda-1b: Scaling latent dynamics action model via universal embodied data ingestion. *arXiv preprint arXiv:2602.12215*, 2026.

Teli Ma, Jia Zheng, Zifan Wang, Chunli Jiang, Andy Cui, Junwei Liang, and Shuo Yang. Dit4dit: Jointly modeling video dynamics and actions for generalizable robot control. *arXiv preprint arXiv:2603.10448*, 2026.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

Yichao Shen, Fangyun Wei, Zhiying Du, Yaobo Liang, Yan Lu, Jiaolong Yang, Nanning Zheng, and Baining Guo. Videovla: Video generators can be generalizable robot manipulators. *Advances in neural information processing systems*, 38:95597–95621, 2026.

Kairos Team, Fei Wang, Shan You, Qiming Zhang, Tao Huang, Zuoyi Fu, Zhisheng Zheng, Yunlong Xi, Feng Lv, Xiaoming Wu, et al. Kairos: A native world model stack for physical ai. *arXiv preprint arXiv:2606.16533*, 2026.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

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

John Won, Kyungmin Lee, Huiwon Jang, Dongyoung Kim, and Jinwoo Shin. Dual-stream diffusion for world-model augmented vision-language-action model. *arXiv preprint arXiv:2510.27607*, 2025.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *International Conference on Learning Representations*, volume 2024, pp. 10641–10662, 2024.

Mengda Xu, Zhenjia Xu, Yinghao Xu, Cheng Chi, Gordon Wetzstein, Manuela Veloso, and Shuran Song. Flow as the cross-domain manipulation interface. *arXiv preprint arXiv:2407.15208*, 2024.

Haodong Yan, Junfeng Li, Junjie He, Zhide Zhong, MingMing Yu, Wenxuan Song, Jiaguan Zhu, Yangyang Zheng, Yuqiao Du, Jiadi You, et al. Robust-wam: Bridging generative pretraining and semantic foresight in world-action models. *arXiv preprint arXiv:2608.05903*, 2026a.

Haodong Yan, Zhide Zhong, Jiaguan Zhu, Junjie He, Weilin Yuan, Wenxuan Song, Xin Gong, Yingjie Cai, Guanyi Zhao, Xu Yan, et al. S-vam: Shortcut video-action model by self-distilling geometric and semantic foresight. *arXiv preprint arXiv:2603.16195*, 2026b.

Liudi Yang, Yang Bai, George Eskandar, Fengyi Shen, Mohammad Altillawi, Dong Chen, Ziyuan Liu, and Abhinav Valada. Covar: Co-generation of video and action for robotic manipulation via multi-modal diffusion. *arXiv preprint arXiv:2512.16023*, 2025.

Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Dekang Qi, Junjin Xiao, Haoyun Liu, Ronghan Chen, Yuzhi Chen, Dongjie Huo, et al. Abot-m0: Vla foundation model for robotic manipulation with action manifold learning. *arXiv preprint arXiv:2602.11236*, 2026.

Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, Jie Li, Jindi Lv, Jingyu Liu, et al. Gigaworld-policy: An efficient action-centered world–action model. *arXiv preprint arXiv:2603.17240*, 2026a.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026b.

Ge Yuan, Qiyuan Qiao, Jing Zhang, and Dong Xu. Adaworldpolicy: World-model-driven diffusion policy with online adaptive learning for robotic manipulation. *arXiv preprint arXiv:2602.20057*, 2026a.

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*, 2026b.

Weili Zeng, Yitong Xing, Fulong Liu, Chengqun Yang, Antao Xiang, Feng Tian, Jingnan Gao, Jisong Cai, Xin Wang, Xiaomin Wu, et al. Enfold: Folding world model imagination into predictive representations for ultra-efficient embodied control. *arXiv preprint arXiv:2607.26657*, 2026.

Han Zhao, Jingbo Wang, Wenxuan Song, Shuai Chen, Yang Liu, Yan Wang, Haoang Li, and Donglin Wang. Frappe: Infusing world modeling into generalist policies via multiple future representation alignment. *arXiv preprint arXiv:2602.17259*, 2026.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1702–1713. IEEE, 2025.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *International Conference on Learning Representations*, volume 2026, pp. 60580–60606, 2026.

Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loic Magne, et al. Flare: Robot learning with implicit world modeling. *arXiv preprint arXiv:2505.15659*, 2025.

Hongyan Zhi, Peihao Chen, Siyuan Zhou, Yubo Dong, Quanxi Wu, Lei Han, and Mingkui Tan. 3dflowaction: Learning cross-embodiment manipulation from 3d flow world model. *arXiv preprint arXiv:2506.06199*, 2025.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. *arXiv preprint arXiv:2504.02792*, 2025.

# A TRAINING DETAILS

**Architecture and inputs.** We initialize the visual branch from Wan2.1-T2V-1.3B, retaining its video DiT, text encoder, and video VAE (Wan et al., 2025). We precompute instruction embeddings with the corresponding Wan2.1 text encoder. The video DiT comprises 30 transformer blocks, while the default action expert uses one block and attends to the final-layer video K/V cache. We additionally evaluate action experts with 6, 12, and 30 blocks in the depth ablation; the component and register–teacher ablations use 30-block action experts. The video branch uses hidden dimension $d_v = 1536$, whereas the action expert uses $d_a = 1024$ and is initialized from a linearly interpolated Wan2.1 ActionDiT checkpoint. The action horizon is $H = 32$.

Each training example spans 33 observation time steps and a 32-step action chunk. The data loader samples video frames at indices $0, 4, \ldots, 32$, yielding nine RGB frames. The VAE then compresses these to $1 + (9 - 1)/4 = 3$ latent temporal positions: one current-frame position and two future slots. The teacher observation pair uses the first and last sampled frames, corresponding to original indices 0 and 32. For LIBERO, we concatenate the two synchronized camera views along the image width before VAE encoding, producing a $224 \times 448$ image composed of two $224 \times 224$ views. The policy additionally receives an 8-dimensional proprioceptive state. Each action is seven-dimensional, comprising a 6-DoF end-effector pose and one gripper-control dimension.

**Optimization and inference.** We jointly optimize the video expert, action expert, dynamics registers, latent-action projection, and proprioceptive encoder using AdamW, with the settings in Table 7. Video and action training use continuous flow matching. Latent-action supervision uses mean-squared error against detached targets from the frozen LaWM teacher. We disable readability registers and allow action gradients to propagate through the video K/V interface. At inference, one video prefill at $\sigma = 1.0$ supplies the cache reused throughout action denoising. None of the reported variants receives additional policy-level embodied pretraining before LIBERO training.

Table 7: Shared training and inference hyperparameters.

| Parameter | Value |
|---|---|
| Optimizer | AdamW |
| Learning rate | $1 \times 10^{-4}$ |
| Weight decay | 0.01 |
| Learning-rate schedule | Cosine decay |
| Maximum gradient norm | 1.0 |
| Flow schedules | 1,000 timesteps; shift 5.0 |
| Video / action / latent-action loss weights | 1.0 / 1.0 / 1.0 |
| Dynamics registers / teacher target dimension | 16 / 32 |
| Action horizon / inference denoising steps | 32 / 10 |
| Training seed | 42 |

**Training budgets.** Table 8 summarizes the dataset-specific training configurations. Piper uses BF16 with an online FP32 LaWM teacher and produces a checkpoint at 10,500 optimizer updates.

Table 8: Dataset-specific training configurations.

| Setting | LIBERO | RoboCasa | Piper |
|---|---|---|---|
| Epochs | 20 | 20 | 20 |
| Global batch size | 128 | 128 | 128 |

**Action-to-video layer mapping.** Each action block reads K/V from a specified video block. With zero-based video block indices, the mappings for action depths 1, 6, and 12 are $[29]$, $[0, 6, 12, 17, 23, 29]$, and $[0, 3, 5, 8, 11, 13, 16, 18, 21, 24, 26, 29]$, respectively. The 30-block action expert uses the identity mapping $[0, 1, \ldots, 29]$. Each depth uses its corresponding interpolated Action DiT initialization. The video expert retains all 30 blocks in every case.

**Policy initialization and pretraining scope.** The Video DiT is initialized from pretrained Wan2.1 weights, and the Action DiT backbone is initialized from the same checkpoint through layer mapping and weight interpolation. We do not perform an additional large-scale robot-data pretraining stage for either branch before task-specific training.

**Separate training forward passes.** For Future-KV with Video DiT registers, training requires two video forward passes to prevent demonstrated future observations from leaking into the action-conditioning features, matching the information available at inference. The first combines the clean current latent with noised demonstrated future latents and supplies the video loss and register states for LaWM supervision. The second combines the clean current latent with newly sampled Gaussian future slots at $\sigma = 1.0$ and recomputes the register states and layer-wise K/V, including register-token K/V, for the action loss. The default one-layer Action DiT reads the final-layer cache. These passes share model parameters and learnable register initializations, but do not reuse each other's hidden states or caches. Demonstrated futures influence the action policy through parameter learning, not as inputs to its conditioning pass. Only the second video pass is required at inference.

**Frozen LaWM teacher.** We use the publicly released LaWM checkpoint from LaWAM (Chen et al., 2026a) as a frozen teacher for latent-action supervision. The teacher provides training targets and is discarded at inference. Its pretraining on external embodied data provides indirect access to external embodied knowledge during policy training; this data provenance remains relevant to comparisons even though the teacher is absent at inference.

**P.T. reporting convention.** In the tables, P.T. denotes large-scale embodied-data pretraining of the policy backbone or action expert before task-specific training, including inherited weights from an embodied-pretrained policy. It excludes general-purpose video pretraining and pretraining of auxiliary frozen teachers. Under this convention, ForeWAM is marked as not using policy-level embodied pretraining, while its use of the pretrained LaWM teacher is disclosed above. This marking does not imply that the complete training pipeline is free of external embodied knowledge.

**Ablation baseline.** The baseline with both components disabled in Table 1 is Fast-WAM retrained with Wan2.1 under the same experimental configuration as the 30-layer ForeWAM ablations. It achieves 53.6% overall success on LIBERO-Plus and is distinct from the 6B Fast-WAM configuration reported in Table 5 (51.5%).

## B Details of the Latency Comparison

**Evaluation setting.** The comparison in Table 6 measures the time required for one policy query on LIBERO-10 using a single NVIDIA A800 GPU with 80 GB of memory. Both methods use 10 action-denoising steps. We perform 5 warm-up iterations before collecting 10 timed runs and report the arithmetic mean in milliseconds. We time action-chunk prediction, excluding environment stepping, rendering, and episode resets.

**ForeWAM configuration.** ForeWAM uses the Wan2.1-T2V-1.3B visual branch and the one-layer Action DiT described in Appendix A. The two camera views form a $224 \times 448$ input, accompanied by the current proprioceptive state. The model encodes the current observation and performs one Video DiT prefill over the current latent, noisy future slots, and dynamics registers. The action expert reuses the final-layer video K/V cache across all 10 denoising steps to produce a 32-action chunk. Readability registers are disabled. Neither future observations nor the latent-action teacher is required at inference, and no future video is decoded.

**Timing breakdown.** *VAE Enc.* measures encoding of the current visual observation into latent space. *Visual* covers visual-context computation, including ForeWAM's Video DiT prefill and construction of the K/V cache. *Action* covers action generation across the full denoising loop. *Other* accounts for the remaining inference overhead outside these three stages, while *Overall* reports the total policy-query latency. For the reported means, these components sum to 88.7 ms for ForeWAM and 556.3 ms for Fast-WAM.

**Latency reduction.** ForeWAM reduces total latency by 84.1%, corresponding to a $6.27\times$ speedup. Action generation accounts for 448.9 ms of the 467.6 ms total saving, whereas visual-context computation differs by only 1.3 ms.

## C Real-World Experiment Details

**Platform and task definitions.** We use a 6-DoF AgileX Piper arm with an external RGB camera and a wrist-mounted RGB camera. The four tasks follow the naming and order in the main text. **Place Pen in Cup (T1)** requires placing the pen in the target cup. **Place Carrot in Bowl (T2)** requires picking up the carrot and placing it in the bowl. **Place Blocks in Drawer and Close It (T3)** requires placing the blocks in the second drawer and then closing it. **Stack Three Bowls (T4)** requires forming a complete stack of three bowls.

**Dataset and observation format.** The dataset contains 125 demonstrations and 71,702 recorded frames collected through master–slave teleoperation. We train a single language-conditioned policy jointly on demonstrations from all four tasks.

The external and wrist cameras provide RGB images at $480 \times 640$ resolution. Each view is resized to $224 \times 224$ and the two views are concatenated horizontally to form a $224 \times 448$ video input. Both recorded proprioceptive states and actions are seven-dimensional. The Piper configuration preserves the recorded absolute action representation without converting it to a state-relative delta. State and action normalization uses the dataset's 1st and 99th percentile statistics. A training window contains 32 actions; its video sampling and temporal compression follow Appendix A. Instruction embeddings are cached with a maximum context length of 128 tokens.

The policy uses the default one-block action expert and 16 dynamics registers described in Appendix A. Online LaWM supervision uses only the external-camera image pair; the teacher does not receive robot states or actions.

**Evaluation protocol.** The ID evaluation protocol specifies 10 trials per task, or 40 trials per method. Object placements are randomized across trials and matched across methods, and success rates are reported separately for each task according to the task goals above. At inference, the model uses the current two-camera observation and proprioceptive state, performs one video prefill with pure-noise future slots, and generates a 32-action chunk using 10 action-denoising steps. Demonstrated future observations and the LaWM teacher are not used during policy execution.

## D Detailed Component Ablation

Table 9 provides the complete seven-category results for the component ablation in Section 3.1. All configurations use a 30-layer Wan2.1 Video DiT and a 30-layer Action DiT without additional policy-level embodied pretraining. The Avg column matches the compact comparison in Table 1.

Table 9: Complementarity of Future-KV and latent-action supervision. LIBERO-Plus success rate (%), without additional policy-level embodied pretraining, using a 30-layer Action DiT. The baseline is Fast-WAM retrained with Wan2.1 under matched settings. ✓/× indicate inclusion/omission. Bold marks the best result in each column.

| Latent action | Future feature | Camera | Robot | Language | Light | Background | Noise | Layout | Avg |
|---|---|---|---|---|---|---|---|---|---|
| × | × | 37.0 | 36.5 | 64.1 | 81.5 | 49.3 | 50.1 | 63.7 | 53.6 |
| ✓ | × | 49.0 | 36.8 | 65.1 | 75.9 | 48.5 | 50.8 | 61.7 | 54.8 |
| × | ✓ | 37.8 | 41.0 | 63.9 | 83.4 | 65.7 | 61.5 | 65.6 | 58.5 |
| ✓ | ✓ | **64.6** | **61.6** | **86.6** | **95.3** | **74.1** | **77.7** | **81.7** | **76.7** |

## E DETAILED LIBERO RESULTS

Table 10 reports the suite-level results corresponding to the average success rates in Table 4.

Table 10: Detailed success rates (%) on the four LIBERO suites.

| Method | P.T. | Spatial | Object | Goal | Long | **Avg.** |
|---|---|---|---|---|---|---|
| OpenVLA (Kim et al., 2024) | ✓ | 84.7 | 88.4 | 79.2 | 53.7 | 76.5 |
| WorldVLA (Cen et al., 2025b) | ✓ | 87.6 | 96.2 | 83.4 | 60.0 | 81.8 |
| $\pi_0$-Fast (Pertsch et al., 2025) | ✓ | 96.4 | 96.8 | 88.6 | 60.2 | 85.5 |
| $\pi_0$ (Black et al., 2024) | ✓ | 96.8 | 98.8 | 95.8 | 85.2 | 94.1 |
| $\pi_{0.5}$ (Intelligence et al., 2025) | ✓ | **98.8** | 98.2 | **98.0** | 92.4 | 96.9 |
| UniVLA (Bu et al., 2025) | ✓ | 96.5 | 96.8 | 95.6 | 92.0 | 95.2 |
| CogVLA (Li et al., 2026b) | ✓ | 98.6 | 98.8 | 96.6 | 95.4 | 97.4 |
| Fast-WAM (Yuan et al., 2026b) | × | 98.2 | **100.0** | 97.0 | 95.2 | 97.6 |
| **ForeWAM** | × | 97.6 | 99.8 | **98.0** | **96.2** | **97.9** |

## F BENCHMARK EVALUATION PROTOCOLS

**LIBERO and LIBERO-Plus.** We evaluate our model on LIBERO (Liu et al., 2023) and LIBERO-Plus (Fei et al., 2025). For LIBERO, we follow the standard benchmarking protocol and train on the four standard suites: Spatial, Object, Goal, and LIBERO-Long. Each suite contains 500 expert demonstrations across 10 tasks.

LIBERO-Plus extends the LIBERO tasks with additional visual and layout variations. For training, we use the original LIBERO demonstrations and exclude the augmented LIBERO-Plus training data. We evaluate the trained policies under the LIBERO-Plus protocol and report overall success rate. For every table reporting Avg, we aggregate evaluation episodes as

$$\text{Avg} = \text{Overall SR} = 100 \times \frac{\sum_c S_c}{\sum_c N_c}, \tag{9}$$

where $S_c$ and $N_c$ are the numbers of successful and total evaluation episodes in group $c$, respectively (e.g., a LIBERO suite or a LIBERO-Plus perturbation category). Thus, Avg weights each group by its episode count; it is not the unweighted arithmetic mean of category percentages unless all groups contain the same number of episodes. Aggregation uses the underlying counts before rounding.

**RoboCasa.** We evaluate on 24 static kitchen manipulation tasks with a single Franka Emika Panda arm. Each task uses five scenes with different floor plans and visual styles, with 10 trials per scene. We report the mean success rate over all 24 tasks and three random seeds, for 3,600 trials in total. The evaluation uses unseen object instances, and two of the five scenes for each task contain styles absent from training.

We train ForeWAM on the successful human demonstrations in the released RoboCasa dataset and do not use the additional MimicGen demonstrations.

**Real-world Experiments.** Our real-world setup consists of a 6-DoF AgileX Piper arm with third-person and wrist-mounted RGB cameras. The task suite includes four tabletop tasks: **Place Pen in Cup (T1)**, **Place Carrot in Bowl (T2)**, **Place Blocks in Drawer and Close It (T3)**, and **Stack Three Bowls (T4)**. These tasks cover precision placement, pick-and-place, articulated-object manipulation, and multi-stage stacking. In T3, the target is the second drawer. The training dataset contains 125 robot demonstrations collected through master–slave teleoperation, totaling 71,702 frames across the four tasks. We train ForeWAM jointly on demonstrations from all four tasks. The ID evaluation protocol specifies 10 trials per task, with object placements randomized across trials and matched across methods. We also evaluate out-of-distribution (OOD) robustness by perturbing object attributes in pen placement and drawer manipulation (Figure 4).