GLANCEWAM: SPARSE TEST-TIME IMAGINATION FOR WORLD-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Video generative models provide rich physical priors for robot learning, yet existing world-action models (WAMs) face a fundamental trade-off: synchronous video generation at control rate is latency-prohibitive, while abandoning test-time visual imagination sacrifices task success. We show that visual imagination achieves both real-time inference and superior success rates when generated asynchronously off the critical path and consumed directly in latent space. We introduce GlanceWAM, which decouples imagination from control on a single shared video DiT backbone: an asynchronous proposer glances ahead on a slow clock to imagine a single lookahead frame seconds into the future in the background, while an action head decodes action chunks at control rate (48 ms) purely in latent space without blocking. Enabled by a non-interfering attention mask that isolates video representations and staleness-robust horizon training that accommodates asynchronous lookahead aging, GlanceWAM breaks the speed–success dilemma. Trained purely on demonstrations, it attains 72.2% on the 24-task RoboCasa kitchen benchmark (vs. 67.1% for synchronous Cosmos Policy) and 99.0% on LIBERO while cutting per-chunk control latency $24\times$ relative to synchronous world-action models (48 ms on one A100). In single-arm and bimanual real-robot manipulation, it achieves higher average success than $\pi_{0.5}$ without any robot-data pretraining.

1 INTRODUCTION

Video generative models capture rich physical priors over object dynamics, contact physics, and 3D scene evolution (NVIDIA, 2025a; Wan Team, 2025; SkyReels Team, 2025), offering a promising foundation for autonomous robots to anticipate the consequences of their actions. In robot manipulation, world-action models (WAMs) leverage these predictive backbones through two distinct pathways: *(i) representation shaping*, where future-prediction losses enrich shared visual features, and *(ii) visual foresight*, where the model explicitly synthesizes future visual states to guide downstream policy execution (Kim et al., 2026). However, existing WAMs couple future prediction and action decoding synchronously at the control rate (Figure 1a) (Kim et al., 2026; Li et al., 2026c). This tight coupling incurs two fundamental costs: prohibitive inference latency and myopic foresight. Over the short duration of an action chunk the scene barely changes, so predicting only that far ahead reduces world modeling to near-trivial reconstruction of the current observation.

To bypass this latency bottleneck, recent approaches advocate abandoning test-time visual imagination entirely, relegating video modeling purely to offline pretraining or feature regularization (Yuan et al., 2026b; Cai et al., 2026). While this strategy eliminates generative overhead during control, it discards explicit visual foresight—depriving downstream policies of the future visual destinations and spatial guidance needed for long-horizon manipulation. Conversely, architectures that retain visual foresight continue to couple dense future synthesis directly to high-frequency action chunks (Kim et al., 2026; Li et al., 2026c), remaining bounded by prohibitive diffusion latencies and near-static prediction horizons. Current world-action models are thus caught in a *speed–success dilemma*: either synthesize video synchronously at the control rate and forfeit real-time reactivity, or discard test-time imagination and forfeit the performance benefits of visual foresight.

In this paper, we show that this dilemma is not an intrinsic property of visual foresight, but an artifact of coupling imagination synchronously to high-frequency control chunks. Downstream policies do

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

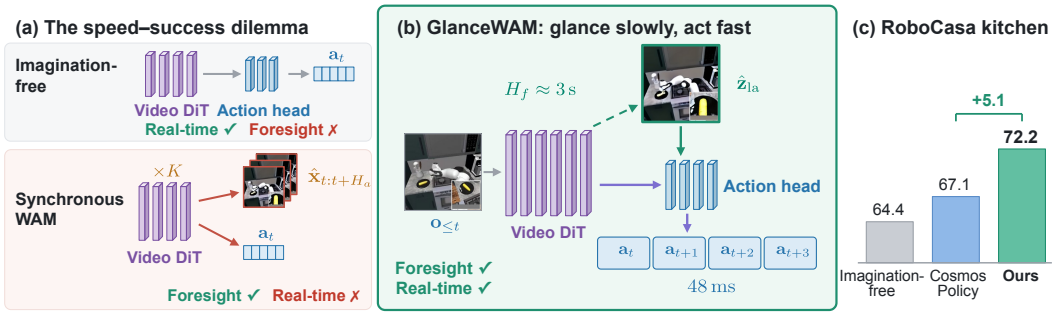

Figure 1: **Synchronous imagination versus sparse lookahead foresight. (a)** Existing WAMs either skip test-time foresight or imagine video synchronously with every action chunk, paying K heavy video denoising steps per chunk. **(b)** GlanceWAM glances ahead asynchronously to a single latent lookahead frame $H_f \approx 3\,\mathrm{s}$ away and reuses it across $\approx$4 action chunks, each decoded in 48 ms. **(c)** Success rate on RoboCasa kitchen (24 tasks, demos only).

not require dense, frame-by-frame future video of the immediate milliseconds; they primarily need a distant spatial destination ("where to go"). We introduce GlanceWAM, a framework that decouples visual imagination from real-time control on a single shared video DiT backbone. GlanceWAM makes visual foresight fast through asynchronous execution off the critical path: the model glances ahead on a slow clock to generate a single lookahead frame seconds into the future ($H_f \approx 3\,\mathrm{s}$) in the background, completely isolating generative sampling latency from the high-frequency control loop. Crucially, the entire imagination-and-control loop operates purely in latent space without decoding to raw pixels, allowing the action head to decode short action chunks at the control rate (48 ms) in a single forward pass. This architecture is enabled by two key mechanisms: *(i)* a non-interfering attention mask (prefix-LM) that isolates video representations by preventing lookahead tokens from contaminating observation encodings, and *(ii)* staleness-robust horizon training that supervises the policy across varying time offsets ($u \sim \mathcal{U}(0, H_f]$) to tolerate lookahead aging.

Trained purely on demonstrations, GlanceWAM establishes a new state-of-the-art among world-action models across both manipulation success rate and inference speed, resolving the speed–success dilemma in practice. On the 24-task RoboCasa kitchen benchmark (Nasiriany et al., 2024), GlanceWAM achieves 72.2% success, outperforming both synchronous Cosmos Policy (67.1%) and imagination-free co-training (64.4%), while reaching 99.0% on LIBERO (Liu et al., 2023). Concurrently, its action decoding executes in 48 ms per chunk on a single NVIDIA A100 GPU (24× lower control-path latency than synchronous WAMs), operating comfortably within real-time control budgets. In single-arm and bimanual real-robot manipulation, GlanceWAM achieves higher average success than $\pi_{0.5}$ (Physical Intelligence, 2025) without any robot-data pretraining (§4.3). Systematic diagnostics further confirm that lookahead conditioning is causally load-bearing and remains robust under asynchronous execution delays, demonstrating that world-action models do not need to choose between speed and imagination—sparse, asynchronous visual foresight delivers both.

2 Related Work

World-action models. Generative video models capture expressive physical priors over scene dynamics, prompting diverse strategies to harness them for robot manipulation. Early "predict-then-act" paradigms synthesize dense multi-frame video rollouts and decode actions through inverse dynamics (Du et al., 2023; Ajay et al., 2023; Li et al., 2026d), incurring severe generative latency on the critical control path. Recent world-action models (WAMs) tighten this integration: Cosmos Policy (Kim et al., 2026) co-denoises robot actions, single future observation frames, and value estimates within a unified diffusion sequence; LingBot-VA (Li et al., 2026c), VideoVLA (Shen et al., 2025), and GigaWorld-Policy (Ye et al., 2026) autoregressively interleave video and action tokens across matched intervals; and UWM (UWM authors, 2025) jointly trains video and action diffusion. In these architectures, future prediction is tightly coupled to the action chunk horizon (e.g., 0.8–1.6 s), paying multi-step diffusion sampling overhead on every action chunk while predicting near-static short-horizon transitions. Conversely, purely auxiliary frameworks such as FLARE (FLARE

authors, 2025) leverage future prediction strictly as a representation learning objective without test-time foresight. GlanceWAM unifies both pathways: it decouples timescales to generate seconds-scale lookaheads ($H_f \approx 3$ s) on an amortized slow clock, while retaining real-time 48 ms action decoding purely in latent space. Concurrent works such as DeVA (Zhang et al., 2026) and Flex-π (Yan et al., 2026) explore alternative joint video–action denoising and compute-flexible architectures (detailed comparisons in Appendix A).

Test-time imagination and efficiency. A recent line of inquiry questions whether world-action models require test-time visual imagination at all. Fast-WAM (Yuan et al., 2026b) and AHA-WAM (Cai et al., 2026) argue that future prediction is unnecessary during inference and can be relegated entirely to offline representation shaping. Both achieve this through causal attention masking, isolating observation token encodings so that future prediction tokens can be losslessly removed at test time. AHA-WAM further introduces observation-guided context routing and phase-offset training to refresh stale planner representations under temporal latency. Complementary acceleration efforts explore progressive distillation (Akbari et al., 2026), lightweight 1B architectures (Li et al., 2026b;e), persistent rolling memory (Yang et al., 2026b), or predictive representation folding (Zeng et al., 2026). While removing or distilling test-time generation mitigates sampling latency, amortizing world dynamics strictly into static weights deprives downstream policies of explicit visual targets. In this work, we demonstrate within a controlled, unified architecture that test-time visual foresight provides critical task guidance ($+7.8\%$ on RoboCasa kitchen), and show that sparse asynchronous amortization removes the latency bottleneck without discarding imagination. Unlike prior WAMs that retain test-time imagination, GlanceWAM takes generation off the control path.

Visual lookaheads and foresight. Conditioning visuomotor policies on future visual targets has a rich foundation in hierarchical robot learning. Prior methods instantiate visual subgoals through image-editing models (Black et al., 2024b), progress-filtered subgoal candidates (Hatch & and others, 2025), and dedicated high-level video planners refreshed every few seconds (Physical Intelligence, 2026). While GlanceWAM shares the principle of multi-second visual lookahead conditioning, it unifies the lookahead generator and policy backbone within a single video DiT rather than maintaining separate, disjoint models (e.g., BAGEL and π_0 in Physical Intelligence (2026)), ensuring that predictive world modeling directly shapes shared policy representations. On latency and representation grounds, an emerging line explores non-RGB and latent-space foresight: LaWAM (Chen et al., 2026) and RepWAM (Wang et al., 2026) propose predicting latent visual features or representation tokenizers rather than raw pixels; EgoWAM (Li et al., 2026a), DreamWAM (Yuan et al., 2026a), LiLa-WAM (Yang et al., 2026a), and VLA-JEPA (Sun et al., 2026) explore geometry, 3D flow, and joint-embedding predictive representations. However, in methods like LaWAM, subgoals are generated conditioned on the policy's already-predicted actions, rendering the subgoal downstream of action selection. In contrast, GlanceWAM synthesizes action-independent lookahead latents seconds in advance to provide explicit spatial destinations ("where to go") that guide subsequent action chunks directly in latent space.

3 METHOD

In this section, we present GlanceWAM, a world-action model that decouples test-time visual foresight from high-frequency action execution on one shared video DiT backbone, read by a lightweight flow-matching action head (Lipman et al., 2023; NVIDIA, 2025b). We formalize the dual-timescale problem (§3.1), then detail the architecture (§3.2), staleness-robust co-training (§3.3), and asynchronous latent-space inference (§3.4).

3.1 PROBLEM FORMULATION AND DUAL-TIMESCALE SETUP

Setting and video diffusion foundation. We consider language-conditioned visuomotor manipulation from demonstration trajectories. At each decision step t, the policy receives observation history $\mathbf{o}_{\le t}$ and instruction $\mathbf{c}$, predicting an action chunk $\mathbf{a}_t = [a_t, \dots, a_{t+H_a-1}] \in \mathbb{R}^{H_a \times D_a}$ spanning control horizon H_a (16 steps at 20 Hz $= 0.8$ s) (Zhao et al., 2023; Chi et al., 2023; Kim et al., 2026). Our framework builds upon a latent video diffusion transformer (SkyReels-V2-DF, 1.3B) (SkyReels Team, 2025) that compresses video frames into latent representations $\mathbf{z}_i$ via a frozen causal video VAE encoder $\mathcal{E}$ (Wan Team, 2025) (reproducibility details in Appendix E). We write $\mathbf{o}$ for observed frames, available at deployment, and $\mathbf{x}$ for future frames of the demonstration video, available only

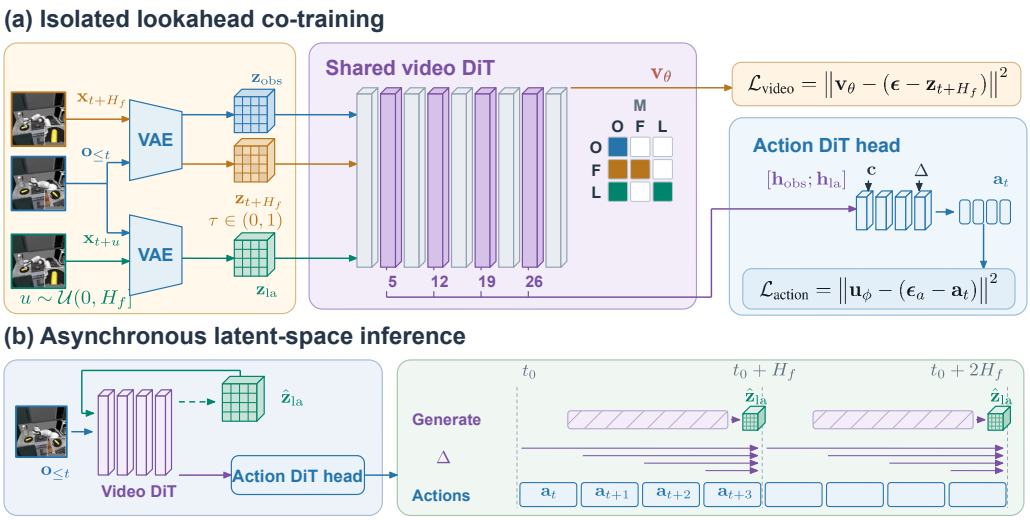

Figure 2: **GlanceWAM architecture.** **(a)** In co-training, the video DiT predicts the future target $\mathbf{x}_{t+H_f}$, while the action head learns inverse dynamics toward a lookahead frame $\mathbf{x}_{t+u}$ at a random offset $u \sim \mathcal{U}(0, H_f]$, conditioned on $\Delta = u$, so it sees every lookahead age it will meet at deployment. Two separate VAE passes and the prefix-LM mask $\mathbf{M}$ keep the lookahead out of video prediction. **(b)** At inference, the lookahead latent $\hat{\mathbf{z}}_{\text{la}}$ is generated asynchronously once per H_f and never decoded; each 0.8 s action chunk reads the held $\hat{\mathbf{z}}_{\text{la}}$ at the decaying offset Δ in 48 ms, within the offset range seen in training.

in training; both are encoded by $\mathcal{E}$ into latents $\mathbf{z}$. Under diffusion forcing (Chen et al., 2024), each latent frame carries an independent noise level $\tau_i \in [0, 1]$: clean observation frames have $\tau_i = 0$, while generative targets draw τ_i from a shifted logit-normal (Esser et al., 2024).

Dual-timescale formulation. A visual forward world model anticipates future scene evolution $\mathbf{x}_{t+H_f}$ over a foresight horizon H_f conditioned on context. Existing world-action models couple foresight synchronously to the control rate ($H_f \equiv H_a$) (Kim et al., 2026; Li et al., 2026c), incurring heavy sampling delays and myopic foresight (§1). To resolve this, GlanceWAM decouples the foresight horizon ($H_f \approx 3.0$ s on a slow background clock) from the control rate ($H_a = 0.8$ s on a fast latent clock). This decoupling introduces a central challenge: a lookahead latent generated once per H_f is held and reused across $\approx H_f/H_a$ consecutive action chunks. Consequently, the policy must act against visual foresight whose temporal offset Δ decays from H_f toward 0 between refreshes — a staleness that the training interface must anticipate.

3.2 Unified Latent World-Action Architecture

Three-role training window. Training a dual-timescale world-action model requires supervising two concurrent capabilities: generating visual foresight and conditioning actions on that foresight. To supervise both in a single forward pass, each training sequence provides three distinct visual inputs (Figure 2a): *(i) Observation history* $\mathbf{o}_{\leq t}$ ($\tau = 0$) provides clean visual context. *(ii)* A noised *future target* $\mathbf{x}_{t+H_f}$ ($\tau \in (0, 1)$) at the full foresight horizon H_f supervises the video backbone's generative forward dynamics. *(iii)* A clean *lookahead condition* $\mathbf{x}_{t+u}$ ($\tau = 0$) at a randomized intermediate offset $u \sim \mathcal{U}(0, H_f]$ provides teacher-forced visual guidance for action execution. Separating the future target from the lookahead condition is essential: while the world model must learn to predict long-horizon transitions at H_f, the policy at deployment executes against a held lookahead whose remaining offset Δ decays over time. Sampling $u \in (0, H_f]$ exposes the policy to this varying offset during training. The three roles occupy dedicated 3D rotary position embedding (RoPE) temporal slots (0, 1, and 2).

Two-pass causal visual encoding. Encoding these three frames into latent tokens requires preventing temporal information leakage during compression. Because standard 3D causal video VAEs (Wan Team, 2025; SkyReels Team, 2025) aggregate features temporally across frames, encod-

ing all three frames in a single pass would allow future information from $\mathbf{x}_{t+H_f}$ to contaminate the lookahead latent $\mathbf{z}_{t+u}$. To guarantee strict causal isolation, we encode the inputs in two independent passes of $\mathcal{E}$: *Pass 1* (generative stream) computes $\mathcal{E}([\mathbf{o}_{\leq t}, \mathbf{x}_{t+H_f}]) = [\mathbf{z}_{\leq t}, \mathbf{z}_{t+H_f}]$ to provide standard video co-training supervision; *Pass 2* (policy stream) computes $\mathcal{E}([\mathbf{o}_{\leq t}, \mathbf{x}_{t+u}]) = [\mathbf{z}_{\leq t}, \mathbf{z}_{t+u}]$, ensuring the lookahead latent depends solely on past context and its own frame.

Non-interfering 3-class attention mask. A second leakage path arises inside the transformer: under full self-attention, video generation queries could attend directly to the clean lookahead frame, turning future prediction into a trivial copying shortcut. To eliminate representation contamination, we design a structured 3-class prefix-LM block mask $\mathbf{M} \in \{0,1\}^{S\times S}$ (Figure 3) implemented via FlexAttention (He et al., 2024): *(i)* Observation queries attend exclusively to observations ($\mathbf{M}(\mathbf{z}_{\text{obs}}, \mathbf{z}_{\text{obs}}) = 1$). *(ii)* Future prediction queries attend to observations and future targets ($\mathbf{M}(\mathbf{z}_{\text{fut}}, \{\mathbf{z}_{\text{obs}}, \mathbf{z}_{\text{fut}}\}) = 1$), but are strictly blocked from lookahead tokens ($\mathbf{M}(\mathbf{z}_{\text{fut}}, \mathbf{z}_{\text{la}}) = 0$). *(iii)* Lookahead queries attend to observations and themselves ($\mathbf{M}(\mathbf{z}_{\text{la}}, \{\mathbf{z}_{\text{obs}}, \mathbf{z}_{\text{la}}\}) = 1$). Because no non-lookahead tokens attend to the lookahead frame ($\mathbf{M}(\cdot, \mathbf{z}_{\text{la}}) = 0$), the backbone representations for observations and future targets remain mathematically identical to standard video co-training, ensuring all policy gains stem strictly from the lookahead conditioning channel.

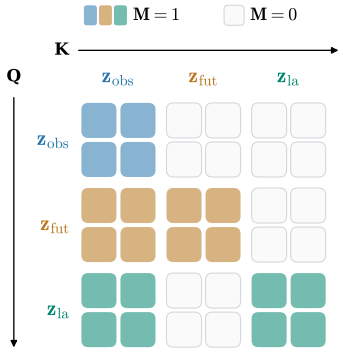

Figure 3: **Non-interfering attention mask.** 3-class prefix-LM mask $\mathbf{M}$.

3.3 Staleness-Robust Co-Training

Joint flow-matching objective. We train the shared video backbone θ_{dit} and the action head ϕ_{act} end-to-end via joint conditional flow matching (Lipman et al., 2023). The video objective supervises forward dynamics velocity prediction $\mathbf{v}_\theta$ on the noised future target $\mathbf{z}^{(\tau)}_{t+H_f}$:

$$\mathcal{L}_{\text{video}}(\theta_{\text{dit}}) = \mathbb{E}_{\tau,\boldsymbol{\epsilon},\mathbf{z}} \left\| \mathbf{v}_\theta \left(\mathbf{z}^{(\tau)}_{t+H_f}, \tau \mid \mathbf{z}_{\leq t}, \mathbf{c} \right) - (\boldsymbol{\epsilon} - \mathbf{z}_{t+H_f}) \right\|^2 . \tag{1}$$

Simultaneously, the action head learns an inverse dynamics model by flow matching on the continuous action chunk $\mathbf{a}_t \in \mathbb{R}^{H_a \times D_a}$, predicting its velocity $\mathbf{u}_\phi$ conditioned on the DiT backbone's multi-layer visual representations $\mathbf{h}$, task instruction $\mathbf{c}$, and lookahead offset Δ:

$$\mathcal{L}_{\text{action}}(\theta_{\text{dit}}, \phi_{\text{act}}) = \mathbb{E}_{\sigma,\boldsymbol{\epsilon}_a,\mathbf{a}} \left\| \mathbf{u}_\phi \left(\mathbf{a}^{(\sigma)}_t, \sigma \mid [\mathbf{h}_{\text{obs}}; \mathbf{h}_{\text{la}}], \mathbf{c}, \Delta \right) - (\boldsymbol{\epsilon}_a - \mathbf{a}_t) \right\|^2 , \tag{2}$$

where σ is the Beta-distributed action flow timestep (Black et al., 2024a), and the overall loss is $\mathcal{L} = \mathcal{L}_{\text{video}} + \mathcal{L}_{\text{action}}$.

Staleness-robust horizon randomization. During asynchronous deployment, a lookahead frame generated at time t_0 is held across multiple control cycles, meaning subsequent action chunks at $t_0 + k \cdot H_a$ execute with an aging visual guide whose remaining offset decays toward zero. To make the policy inherently robust to this staleness without frequent re-generation, we pair the randomized offset sampling $u \sim \mathcal{U}(0, H_f]$ with explicit temporal conditioning: the action head receives the exact offset $\Delta = u$ via a sinusoidal time embedding (Vaswani et al., 2017). Exposing the policy to all intermediate offsets during training teaches it to seamlessly follow visual foresight regardless of where the current execution step falls within the refresh cycle. To retain robust control when foresight is absent or degraded, we apply lookahead token dropout with probability $p = 0.1$.

Multi-layer visual extraction. Rather than extracting features solely from the final DiT block, the action head cross-attends to concatenated representations $[\mathbf{h}_{\text{obs}}; \mathbf{h}_{\text{la}}]$ pooled across four uniformly spaced transformer layers $\{5, 12, 19, 26\}$ (Figure 2a). This multi-layer conditioning combines low-level spatial details from shallow layers with high-level semantic destinations from deep layers, providing a $+0.7\%$ improvement on RoboCasa kitchen (§4.4).

3.4 Asynchronous Latent-Space Inference

Pure latent-space control path. At test time, the model conditions directly on its own generated visual foresight (Figure 2b). Once per horizon H_f ($\sim$3.0 s), the video DiT runs an ODE flow sampler

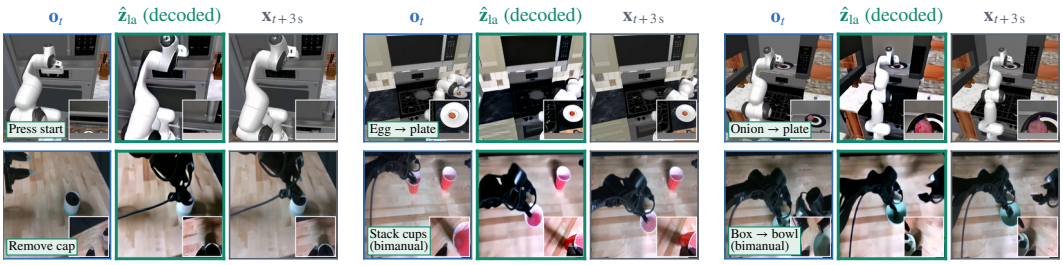

Figure 4: **Imagined lookaheads.** Observation $\mathbf{o}_t$, the GlanceWAM lookahead $\hat{\mathbf{z}}_{\text{la}}$ (decoded for display only), and the real frame $\mathbf{x}_{t+3\,\text{s}}$, in RoboCasa kitchen (top) and real IMETA Y1 rollouts (bottom). The lookahead gets the target layout right while coarsening texture. Wrist view inset; more in Figure 10.

for 1–10 steps to generate the lookahead latent $\hat{\mathbf{z}}_{\text{la}}$ from current observation tokens $\mathbf{z}_{\leq t}$. Crucially, $\hat{\mathbf{z}}_{\text{la}}$ is never decoded to raw RGB pixels (Figure 4 decodes some for display): it is retained entirely within the normalized latent space of the causal VAE, directly serving as the slot-2 conditioning tokens for subsequent action forward passes. Eliminating VAE decoding from the control loop removes substantial computational overhead and preserves fine-grained spatial representations.

Asynchronous amortization. While the lookahead latent is held, the action head decodes subsequent $0.8\,\text{s}$ action chunks in real time ($48\,\text{ms}$ per chunk on a single NVIDIA A100 GPU). Each chunk conditions on the decaying lookahead offset $\Delta = H_f - (t \bmod H_f) \in (0, H_f]$, matching the training distribution of u (§3.3). Because a single lookahead frame serves approximately 4 consecutive action chunks ($H_f/H_a \approx 4$), video sampling overhead is amortized across control cycles ($\sim$17% at 10 steps, and $\sim$2% at the 1-step regime validated in §4.5). By pipelining lookahead generation on a background thread behind active action execution, the lookahead proposer leaves the critical control path entirely, enabling low-latency, closed-loop manipulation.

4 Experiments

Our experimental evaluation addresses four questions: **(Q1)** How does GlanceWAM compare with state-of-the-art imitation policies and world-action models? (§4.2) **(Q2)** Does the sparse lookahead transfer to physical single-arm and bimanual robots? (§4.3) **(Q3)** Where does the gain come from, and is the lookahead causally load-bearing? (§4.4) **(Q4)** What does the lookahead cost at deployment, in latency and generative compute? (§4.5)

4.1 Experimental Setup

RoboCasa kitchen. RoboCasa (Nasiriany et al., 2024) comprises 24 kitchen manipulation tasks, spanning pick-and-place operations between counters and appliances, door and drawer articulation, knob turning, and button pressing with a Franka Emika Panda arm in procedurally generated scenes. We adopt the evaluation protocol of Cosmos Policy (Kim et al., 2026): reporting average success rates across 50 evaluation episodes per task ($n = 1200$ total) in five held-out kitchen layouts with unseen object instances (10 episodes per scene). Observations comprise three RGB views (two third-person camera views and one wrist view); Figures 4 and 10 show representative tasks alongside model-generated lookahead frames. Following Cosmos Policy, training uses the replay-filtered split of 50 demonstrations per task.

LIBERO. LIBERO (Liu et al., 2023) includes four benchmark suites (Spatial, Object, Goal, and Long) containing 10 manipulation tasks each, with 50 demonstrations per task. We evaluate 50 episodes per task (500 episodes per suite, 2000 episodes total) using two RGB camera views (third-person and wrist).

Baselines. Imitation policies: Diffusion Policy (Chi et al., 2023), π_0 (Black et al., 2024a), π_0-fast (Pertsch et al., 2025), $\pi_{0.5}$ (Physical Intelligence, 2025), OpenVLA-OFT (Kim et al., 2025), CogVLA (CogVLA authors, 2025), and GR00T-N1/N1.5 (NVIDIA, 2025b) with data augmentation (+DreamGen, +DUST, +HAMLET). World-action models: UVA (Li et al., 2025), UWM (UWM au-

Table 1: **RoboCasa kitchen.** 24-task average SR, held-out scenes; baselines as reported (NVIDIA, 2025b; Black et al., 2024a; UWM authors, 2025; Li et al., 2026d; FLARE authors, 2025; Kim et al., 2026). Blocks: VLA policies, WAMs.

Model	SR (%)	Demos/task
GR00T-N1	49.6	300
GR00T-N1 + DreamGen	57.6	300
GR00T-N1 + DUST	58.5	300
π_0	62.5	300
GR00T-N1.5	64.1	300
GR00T-N1.5 + HAMLET	66.4	300
UVA	50.0	300
UWM	60.8	300
Video Policy	66.0	300
FLARE	66.4	300
Cosmos Policy	<u>67.1</u>	50
GlanceWAM (ours)	**72.2**	50

Table 2: **LIBERO.** Per-suite SR as reported (Chi et al., 2023; Pertsch et al., 2025; Physical Intelligence, 2025; Kim et al., 2025; CogVLA authors, 2025; Kim et al., 2026; Li et al., 2026c; Yuan et al., 2026b; Zeng et al., 2026); blocks as in Table 1; best bold, second underlined. Sizes and latencies: Figure 8.

Method	Spatial	Object	Goal	Long	Avg
Diffusion Policy	78.3	92.5	68.3	50.5	72.4
π_0-fast	96.4	96.8	88.6	60.2	85.5
$\pi_{0.5}$	<u>98.8</u>	98.2	98.0	92.4	96.9
OpenVLA-OFT	97.6	98.4	97.9	94.5	97.1
CogVLA	98.6	98.8	96.6	95.4	97.4
Motus	96.8	<u>99.8</u>	96.6	97.6	97.7
DiT4DiT	98.4	99.6	<u>98.6</u>	97.6	<u>98.6</u>
Fast-WAM	98.2	**100.0**	97.0	95.2	97.6
Enfold-Flash	97.0	<u>99.8</u>	96.6	96.6	97.5
Cosmos Policy	98.1	**100.0**	98.2	97.6	98.5
LingBot-VA	98.5	99.6	97.2	**98.5**	98.5
GlanceWAM (ours)	**99.4**	**100.0**	**99.0**	<u>97.8</u>	**99.0**

thors, 2025), Video Policy (Li et al., 2026d), FLARE (FLARE authors, 2025), Cosmos Policy (Kim et al., 2026), LingBot-VA (Li et al., 2026c), Fast-WAM (Yuan et al., 2026b), Motus, Enfold-Flash, and DiT4DiT. Baseline numbers are from the original publications; all internal comparisons share datasets, backbones, and evaluation pipelines.

Training details. All GlanceWAM variants and internal baselines are implemented in StarVLA (StarVLA Community, 2026) and initialize from the pretrained SkyReels-V2-DF-1.3B backbone (SkyReels Team, 2025) and train strictly on demonstration data without online rollouts, specialized data curation, or auxiliary labels beyond RGB images and robot actions. All models are trained on $4\times$ NVIDIA H200 GPUs. Benefiting from effective pretraining and sparse lookahead conditioning, GlanceWAM converges rapidly: it requires only 10k training steps on RoboCasa kitchen and 15k steps on LIBERO to reach top performance, substantially faster than typical baseline training horizons (e.g., 60k–100k+ steps). We evaluate EMA checkpoints (details in Appendix E). In simulation, the client pauses the simulator while each lookahead is generated (asynchronous execution with zero delay; §4.5 tests a pipelined schedule); on the real robot, GlanceWAM runs fully asynchronously on a single machine.

4.2 (Q1) Comparison with State-of-the-Art Policies

Tables 1 and 2 evaluate GlanceWAM against top-performing imitation policies and world-action models. On the 24-task RoboCasa kitchen benchmark, GlanceWAM achieves an average success rate of 72.2% with 50 demonstrations per task, outperforming Cosmos Policy by $+5.1\%$ under matched demonstration data and exceeding all baselines trained with $6\times$ more demonstration episodes (300 vs. 50). On LIBERO, GlanceWAM attains 99.0% average success across the four suites.

4.3 (Q2) Real-Robot Evaluation

We deploy GlanceWAM on the IMETA Y1 platform against $\pi_{0.5}$ (Physical Intelligence, 2025) and Fast-WAM (Yuan et al., 2026b) in two configurations (Figure 5): a single arm (side and wrist cameras) on *stack cups*, *pick and place*, and *remove cup cap*, and a dual-arm setup (top camera plus one wrist camera per arm) on bimanual *stack cups* and *bowl* (pick up the bowl and put the box into it). All policies are fine-tuned on identical 50 demonstrations per task, deployed with LeRobot (Cadene et al., 2026) and real-time chunking (RTC) (Black et al., 2025), and evaluated over 50 physical trials per task under randomized initializations. GlanceWAM achieves 79.2% average success vs. 69.6% for $\pi_{0.5}$ and 48.0% for Fast-WAM, outperforming $\pi_{0.5}$ on macro-reconfigurations: single-arm *stack cups* (94% vs. 78%) and *remove cup cap* (88% vs. 68%), and both bimanual tasks (84% vs. 66% on *stack cups*, 72% vs. 64% on *bowl*). This holds although $\pi_{0.5}$ is pretrained on large-scale robot data, whereas GlanceWAM starts from an internet video prior and sees robot data only in the 50

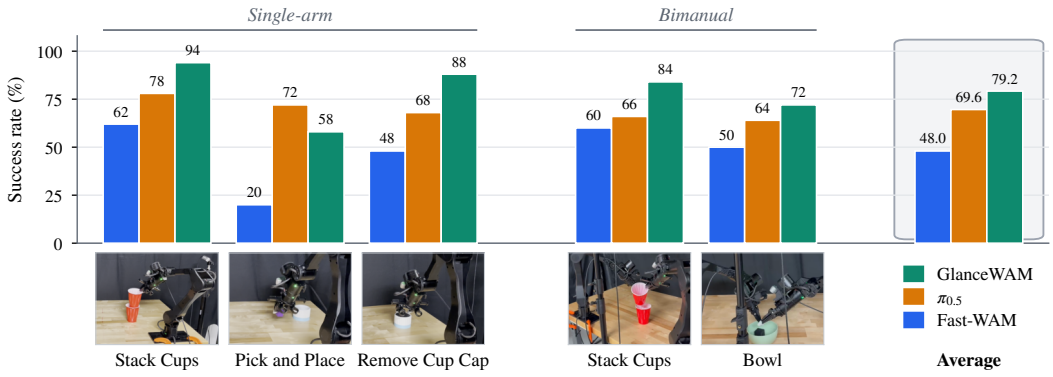

Figure 5: **Real-robot evaluation on IMETA Y1.** Success of GlanceWAM, $\pi_{0.5}$, and Fast-WAM over 50 trials per task; thumbnails show one GlanceWAM rollout per task.

demonstrations. Fast-WAM, which keeps video modeling at training time only, trails both on every task. GlanceWAM trails $\pi_{0.5}$ only on *pick and place* (58% vs. 72%), where the 3×3 cm cube spans just 1–2 latent tokens and millimeter grasp alignment falls to local visual feedback (Appendix D).

4.4 (Q3) Where Does the Gain Come From?

Component isolation. Table 3a systematically isolates the contribution of each design component. Standard video–action co-training without lookahead conditioning (using the same backbone, data, and compute budget) reaches 64.4% success, demonstrating that representation learning alone at a short action horizon does not bridge the performance gap to Cosmos Policy (67.1%). Introducing the lookahead conditioning channel via single-layer extraction yields a +7.1% improvement (71.5%), and extracting lookahead features across four transformer layers (§3.3) provides an additional +0.7%, reaching 72.2%. Replicated lookahead runs land at 71.4% and 71.2%, while lookahead-free co-training plateaus near 64% even at 60k steps. Retraining the single-layer model without the non-interfering mask (bidirectional attention, otherwise identical recipe) collapses success to 47.0%.

Table 3: **Internal comparison on RoboCasa kitchen** (24-task SR, held-out scenes, demos only; shared backbone, data, protocol). **(a)** Components. **(b)** Lookahead sampler steps, single-layer checkpoint, paired re-evaluation (1,200 episodes/point). †Correct lookahead, Δ fixed to one frame; paired sweep, 1,200 episodes (baseline 71.3, zeroed 62.7).

(a) System	SR (%)
Cosmos Policy (external anchor)	67.1
plain co-training (no lookahead)	64.4
GlanceWAM, single-layer lookahead	71.5
same checkpoint, lookahead zeroed	61.6
same checkpoint, offset Δ pinned†	61.8
retrained without mask (bidirectional)	47.0
GlanceWAM, multi-layer (final)	**72.2**

(b) Euler steps	1	2	5	10	30
SR (%)	71.2	71.7	71.4	71.5	69.8

The lookahead is causally load-bearing. To test whether lookahead features are causally responsible for action decisions rather than merely present, we apply an evaluation-time intervention that sets all lookahead tokens to zero on the trained checkpoint. Success drops from 71.5% to 61.6% (Table 3a), below the baseline trained without lookaheads (64.4%): the policy has learned to rely on the lookahead. Keeping the correct lookahead but pinning its offset Δ to one frame falls to the same floor (61.8%): the head also reads the offset.

The action head reads the lookahead. We probe the action head's cross-attention over $[\mathbf{h}_{\text{obs}}; \mathbf{h}_{\text{la}}]$ on 12 demonstration contexts (single-layer checkpoint of Table 3a). Lookahead tokens supply 40% of the action queries' cross-attention output (value-weighted share; 34% of the raw softmax mass). Two independent measurements agree on where the lookahead is read (Figure 6): the value-weighted attention of each lookahead token, and the causal action shift when a 2×2-token lookahead patch is overwritten with the corresponding observation tokens. Both peak on the interaction the lookahead depicts (the gripper at the cabinet-door edge; the mug at the dispenser in Figure 9), correlate



Figure 6: **The action head reads the lookahead.** Value-weighted cross-attention and action sensitivity on a decoded lookahead $\hat{\mathbf{z}}_{\text{la}}$; more in Figure 9.

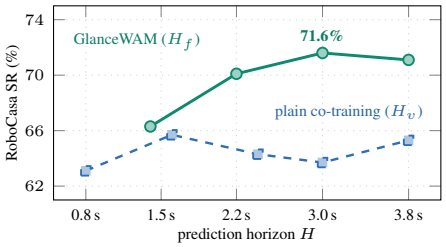

Figure 7: **Prediction horizon.** RoboCasa SR vs. lookahead H_f and plain co-training target H_v.

spatially within a context ($r = 0.55$, mean over the 12 contexts), and split across cameras alike ($0.28/0.32/0.41$ vs. $0.27/0.29/0.43$ for third-person L/R and wrist).

How far ahead should the prediction target be? Figure 7 examines the effect of prediction horizons across both paradigms. For plain co-training (representation shaping), predicting future targets at $H_v = 1.6$–2.4 s outperforms the action-chunk horizon of 0.8 s ($63.1\% \rightarrow 65.7\%$): predicting beyond immediate transitions enriches the representation. For GlanceWAM (visual foresight), success rises from 66.3% at $H_f = 1.4$ s to a peak of 71.6% at 3.0 s (60 frames) and plateaus at 3.8 s (71.1%): foresight helps most when it anticipates distal subgoals ($\sim$3 s) rather than short-horizon transitions.

4.5 (Q4) Deployment Cost

Real-time latent-space decoding. With one clean DiT pass plus the action head and no VAE decoding (§3.4), the control path takes 48 ms per 16-action chunk on LIBERO at $B = 1$ on an NVIDIA A100 40GB GPU (breakdown in Appendix B). Figure 8 illustrates the latency–performance landscape across world-action models: synchronous architectures that reach comparable success rates require 24–80$\times$ higher inference latency (1133–3812 ms), while accelerated alternatives such as Fast-WAM (91.5 ms compiled) and Enfold-Flash (49 ms) exhibit lower task success (97.5–97.6%) while maintaining 2–4$\times$ larger parameter counts (3–6B vs. 1.6B). Lookahead generation (478.6 ms per refresh at 10 steps, $\approx$145 ms at 1 step; Appendix B) runs in the background and never enters the control loop.

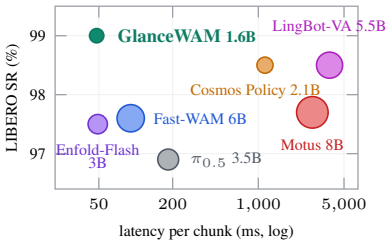

Figure 8: **Latency vs. success on LIBERO.** Per-chunk latency (B=1); marker area: model size.

A pipelined variant that refreshes the lookahead every chunk changes success by -0.5% relative to the blocking evaluation ($p = 0.72$), provided lookaheads are refreshed at the scheduled cadence (paired re-evaluation, 1,200 episodes).

One sampler step suffices. Sweeping the lookahead sampler over 1–30 Euler steps (a 30$\times$ compute span; Table 3b) moves success by at most 1.9% (71.2% at 1 step vs. 71.5% at 10), although image sharpness visibly differs (Figure 11, Appendix C). A 1-step sampler thus cuts proposer compute 10$\times$ at no measurable cost: the policy reads low-frequency spatial layout.

5 Conclusion

Synchronous world-action models tie visual prediction to the action-chunk rate, which makes their foresight both slow and myopic. GlanceWAM decouples the two on one video DiT: it imagines a single latent lookahead frame seconds ahead, off the control path, and decodes actions against it in 48 ms, reaching 72.2% on RoboCasa kitchen, 99.0% on LIBERO, and higher real-robot success than $\pi_{0.5}$ and Fast-WAM. Sparse, asynchronous foresight lets world-action models keep test-time imagination without paying for it in the control loop.

Reproducibility statement

The supplementary material provides anonymized source code, comprising training and simulation evaluation code built on StarVLA (StarVLA Community, 2026) and real-robot deployment code built on LeRobot (Cadene et al., 2026) with RTC (Black et al., 2025), together with real-robot rollout videos for all five tasks. Simulation experiments use the public RoboCasa and LIBERO benchmarks with their standard demonstration splits and evaluation protocols (§4.1). Model architecture, hyperparameters, and training configuration are given in Appendix E, latency measurement in Appendix B, and the physical robot setup in Appendix D.

AI use statement

In this work, we used generative AI tools to assist in implementing methods (writing and editing software code). We have not used generative AI tools to generate synthetic data, develop theoretical models or conceptual frameworks, formulate or prove mathematical claims, propose or refine hypotheses, design research methodology or experiments, clean or reformat datasets, or interpret results; translation and qualitative data analysis are not applicable to this work. Additionally, we used generative AI tools to edit the paper to improve readability. We have reviewed all AI-assisted work: the authors carefully read the full paper and all AI-assisted code, and verified the code through rigorous testing. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

References

Anurag Ajay, Seungwook Han, Yilun Du, Shuang Li, Abhi Gupta, Tommi Jaakkola, Joshua B. Tenenbaum, Leslie Kaelbling, Antonio Torralba, and Pulkit Agrawal. Compositional foundation models for hierarchical planning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. arXiv:2309.08587.

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

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024a.

Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Ilya Kostrikov, and Sergey Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. In *International Conference on Learning Representations (ICLR)*, 2024b. arXiv:2310.10639.

Kevin Black, Manuel Y. Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Remi Cadene, Simon Alibert, Francesco Capuano, Michel Aractingi, Adil Zouitine, Pepijn Kooijmans, Jade Choghari, Martino Russi, Caroline Pascal, Steven Palma, Mustafa Shukor, Jess Moss, Alexander Soare, Dana Aubakirova, Quentin Lhoest, Quentin Gallouédec, and Thomas Wolf. Lerobot: An open-source library for end-to-end robot learning. In *International Conference on Learning Representations (ICLR)*, 2026.

Jisong Cai, Long Ling, Shiwei Chu, et al. AHA-WAM: Asynchronous horizon-adaptive world-action modeling with observation-guided context routing. *arXiv preprint arXiv:2606.09811*, 2026.

Boyuan Chen, Diego Martí Monsé, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2407.01392.

Jialei Chen, Kai Wang, Kang Chen, Shuaihang Chen, Feng Gao, Wenhao Tang, Zhiyuan Li, Weilin Liu, Zhuyu Yao, Boxun Li, Yuanbo Xu, and Chao Yu. LaWAM: Latent world action models for efficient dynamics-aware robot policies. *arXiv preprint arXiv:2606.15768*, 2026.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*, 2023. arXiv:2303.04137.

CogVLA authors. Cogvla. *arXiv preprint*, 2025.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B. Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. arXiv:2302.00111.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning (ICML)*, 2024.

FLARE authors. Flare: Robot learning with implicit world modeling. *arXiv preprint arXiv:2505.15659*, 2025.

Kyle B. Hatch and and others. Ghil-glue: Hierarchical control with filtered subgoal images. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025. arXiv:2410.20018.

Horace He, Yanbo Feng, Andrew Wang, Albert Gu, and Zachary DeVito. Flexattention: Fast, flexible attention with pytorch. *PyTorch Technical Report*, 2024.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

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

Baoyu Li, Xinchen Yin, Mengying Lin, Yixin Zhang, and Danfei Xu. EgoWAM: World action models beyond pixels with in-the-wild egocentric human data. *arXiv preprint arXiv:2607.08436*, 2026a.

Jiajun Li, Tiecheng Guo, Yifan Ye, Rongyu Zhang, Xiaowei Chi, Qianpu Sun, Ying Li, Yunfan Lou, Yan Huang, Zhihe Lu, Meng Guo, and Shanghang Zhang. Efficient-WAM: A 1b-parameter world-action model with low-cost future imagination. *arXiv preprint arXiv:2606.10040*, 2026b.

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

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. In *Robotics: Science and Systems (RSS)*, 2025. arXiv:2503.00200.

Sizhe Lester Li, Evan Kim, Xingjian Bai, Tong Zhao, Tao Pang, Max Simchowitz, and Vincent Sitzmann. Turning video models into generalist robot policies. *arXiv preprint arXiv:2605.27817*, 2026d.

Ziang Li, Dongzhou Cheng, Yibin Wang, Shiyue Wang, Xiaoyang Xu, Lingxuan Weng, Juan Wang, and Jiaqi Wang. Light-WAM: Efficient world action models with state-fusion action decoding. *arXiv preprint arXiv:2606.08242*, 2026e.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2210.02747.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. arXiv:2306.03310.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems (RSS)*, 2024. arXiv:2406.02523.

NVIDIA. Cosmos world foundation model platform for physical AI. *arXiv preprint arXiv:2501.03575*, 2025a.

NVIDIA. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025b.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

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.

Physical Intelligence. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Physical Intelligence. $\pi_{0.7}$: A vision-language-action model with a generative high-level policy. *arXiv preprint arXiv:2604.15483*, 2026.

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. *arXiv preprint arXiv:2512.06963*, 2025.

SkyReels Team. Skyreels-v2: Infinite-length film generative model. *arXiv preprint arXiv:2504.13074*, 2025.

StarVLA Community. Starvla: A lego-like codebase for vision-language-action model developing. *arXiv preprint arXiv:2604.05014*, 2026.

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. VLA-JEPA: Enhancing vision-language-action model with latent world model. *arXiv preprint arXiv:2602.10098*, 2026.

UWM authors. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. *arXiv preprint arXiv:2504.02792*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Wan Team. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Junke Wang, Qihang Zhang, Shuai Yang, Yiming Luo, Yujun Shen, Zuxuan Wu, Yu-Gang Jiang, and Yinghao Xu. RepWAM: World action modeling with representation visual-action tokenizers. *arXiv preprint arXiv:2606.13674*, 2026.

Ge Yan, Jinghao Liu, Yuzhi Fan, Lei Cai, Minwen Liao, Jesse Zhang, and Dieter Fox. Flex-π: A multi-stream world-action model with compute flexibility. *arXiv preprint arXiv:2608.10860*, 2026.

Fan Yang, Yuting Su, Xiaobo Wang, Yuncheng You, Fugui Fan, Yuting Wu, Minghui Wu, Chenxu Zhao, JiaHong Ning, and Peiguang Jing. LiLa-WAM: Lightweight latent reasoning world-action model for robotic manipulation. *arXiv preprint arXiv:2608.03701*, 2026a.

Sizhe Yang, Juncheng Mu, Tianming Wei, Chenhao Lu, Xiaofan Li, Linning Xu, Zhengrong Xue, Zhecheng Yuan, Dahua Lin, Jiangmiao Pang, and Huazhe Xu. MemoryWAM: Efficient world action modeling with persistent memory. *arXiv preprint arXiv:2606.20562*, 2026b.

Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, Jie Li, Jindi Lv, Jingyu Liu, Min Cao, Peng Li, Qiuping Deng, Wenjun Mei, Xiaofeng Wang, Xinze Chen, Xinyu Zhou, Yang Wang, Yifan Chang, Yifan Li, Yukun Zhou, Yun Ye, Zhichao Liu, and Zheng Zhu. GigaWorld-Policy: An efficient action-centered world–action model. *arXiv preprint arXiv:2603.17240*, 2026.

Shanglin Yuan, Weiheng Zhao, Xin Shi, Haoyi Jiang, Xianda Guo, Liu Liu, Wenyu Liu, Wei Sui, and Xinggang Wang. DreamWAM: Beyond RGB future prediction for world action models. *arXiv preprint arXiv:2608.04996*, 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, Yao Mu, Xiaokang Yang, and Yichao Yan. Enfold: Folding world model imagination into predictive representations for ultra-efficient embodied control. *arXiv preprint arXiv:2607.26657*, 2026.

Mengqi Zhang, Sahil Khose, Yuchen Song, Simar Kareer, Unnat Jain, and Judy Hoffman. Deva: Decoupled video-action model with physical guidance for robot policy learning. *arXiv preprint arXiv:2607.24159*, 2026.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems (RSS)*, 2023.

A Concurrent work

Two concurrent world-action models appeared while this work was in preparation (Table 4). DeVA (Zhang et al., 2026) (July 2026) pairs a Cosmos-Predict2 video expert with a GR00T-style action DiT through a multi-level feature bridge, supervises the video features with affordance and depth decoders (labels from contact annotations and a pretrained depth model), and denoises video and action jointly in one synchronous pass. Flex-π (Yan et al., 2026) (August 2026) is a 6B WAM that jointly denoises RGB, 3D-pointmap, and DINOv3 futures with actions, and uses per-stream dropout to fall back to action-only inference at deployment.

Table 4: **Comparison with concurrent world-action models.** Benchmark numbers as reported by each paper under its own evaluation protocol. The DeVA ablation row removes its affordance and depth decoders, leaving RGB demonstrations only, as in GlanceWAM. Flex-π does not report RoboCasa kitchen; its two LIBERO rows are the reported FLEX-π^* deployment modes.

	RoboCasa kitchen	LIBERO (avg)	Auxiliary supervision beyond RGB demos	World model at test time
DeVA (Zhang et al., 2026)	72.0	99.0	affordance + depth decoders (contact labels, depth model)	joint denoising, every chunk
w/o guidance (their ablation)	66.0	—	none	(same)
Flex-π (Yan et al., 2026), action-only	—	98.7	3D pointmaps + DINOv3 semantic futures	dropped (fast path)
Flex-π, full joint	—	99.2	(same)	joint denoising, every chunk (slow path)
GlanceWAM (ours)	**72.2**	99.0	none	async lookahead frame, $\sim$3 s

B Latency measurement

Protocol. All GlanceWAM latencies are measured at $B{=}1$ on a single NVIDIA A100 (SXM4 40 GB, bf16), $n{=}100$ timed calls after 15 warmup calls, CUDA-synchronized wall clock, on the released LIBERO checkpoint (multi-layer, 99.0% in Table 2). Inputs match the evaluation client: two 256^2 views (agent + wrist) stitched to $224{\times}448$, one observation frame, and a 16-action chunk from the 4-step flow-matching head. Since the lookahead proposer runs asynchronously (§4.5), we report the hold-phase action path: `predict_action` with the current lookahead latent already in hand.

Deployment inference path. The timed path uses the same optimizations as our serving stack: the video DiT is truncated after the last feature layer (blocks 1..27 of 30 for layers $\{5, 12, 19, 26\}$; the velocity head is not needed for actions); the VAE encoder, truncated DiT, and action head are compiled with CUDA graphs (`torch.compile`, reduce-overhead); preprocessing runs on GPU;



Figure 9: **Where the action head reads the lookahead** (extends Figure 6). Two probed contexts, third-person view of the decoded lookahead $\hat{\mathbf{z}}_{\text{la}}$. *Cross-attention:* value-weighted contribution of each lookahead token to the action queries' cross-attention output, i.e. the norm of its attention-weighted value after the output projection. The value-weighted share in §4.4 is $\|\text{lookahead part}\|/(\|\text{obs part}\| + \|\text{lookahead part}\|)$ of the attention output. *Action sensitivity:* action shift when a 2×2-token lookahead patch is overwritten with the observation tokens at the same location. Each map is normalized independently.

and the text embedding, lookahead latent, and attention block mask are cached. This measures 47.95 ms per chunk (std 0.39 ms), which we quote as 48 ms; serving adds a websocket round trip of ≈1–2 ms. The unoptimized path (full 30-block forward, per-block compilation) measures 108.3 ms (std 1.15 ms).

Lookahead refresh. A refresh call (generate the lookahead latent, then act on it) measures 478.6 ms at the trained 10-step sampler budget on the unoptimized path, i.e. ≈37 ms per Euler step on top of the 108.3 ms hold path; at the 1-step budget (§4.5) it is ≈145 ms. In the asynchronous design this cost never blocks the control loop: the proposer runs in the background once per H_f (every 3–4 chunks) while the head keeps acting on the held lookahead, so the per-chunk control cost stays at 48 ms.

Same-hardware baselines. Baseline latencies in Figure 8 are measured on the same GPU class (A100 40 GB). We re-benchmarked Fast-WAM (6B MoT, 32-action chunk, 10-step action head) on our A100: its compiled path (`torch.compile` reduce-overhead with CUDA graphs) takes 91.5 ms per chunk, matching the published 91 ms (634 ms in eager mode; Enfold (Zeng et al., 2026) reports 493 ms). Neither system generates video in the per-chunk action pass; GlanceWAM is ≈2× faster than Fast-WAM (48 vs. 91.5 ms) with higher LIBERO success (99.0% vs. 97.6%) and fewer parameters (1.6B vs. 6B).

C Additional figures

Figure 9 extends the attention probe of §4.4, Figure 10 extends the lookahead gallery of Figure 4, and Figure 11 shows lookaheads across the sampler budgets of Table 3b: fidelity improves with compute, success does not.

D Physical robot deployment details and rollouts

Setup. We use the IMETA Y1 platform in two configurations. The single-arm setup is one 6-DoF Y1 arm with a parallel-jaw gripper, observed by a fixed side camera and a wrist camera. The bimanual setup uses two Y1 arms, observed by a fixed top camera and one wrist camera per arm. All cameras record 640×480 RGB, cropped to the workspace and resized to 224×224 for the backbone. Actions are absolute joint positions plus gripper position for each arm (7-D single-arm, 14-D bimanual), recorded and executed at 20 Hz. GlanceWAM runs fully asynchronously on a single machine, with lookahead generation hidden behind action execution. The deployment stack is built on LeRobot (Cadene et al., 2026), and all three policies execute action chunks with real-time chunking (RTC) (Black et al., 2025).

Tasks and protocol. Three single-arm and two bimanual tasks:

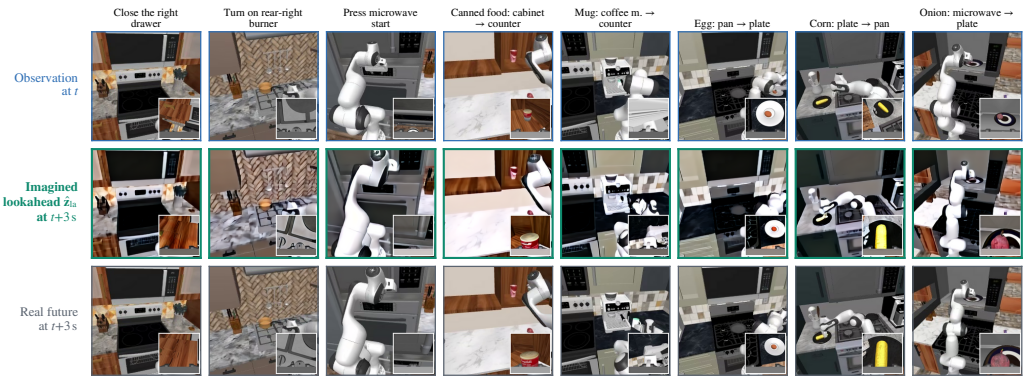

Figure 10: **Lookahead frames on eight kitchen tasks** (extends Figure 4). For eight RoboCasa kitchen tasks (articulation, knobs, buttons, pick-and-place): the observation at t, the GlanceWAM lookahead, and the environment frame at $t{+}3$ s. Lookaheads get the target layout right (closed doors, objects on target surfaces) while coarsening texture; they are decoded to RGB only for display. One third-person view per panel, wrist view inset.

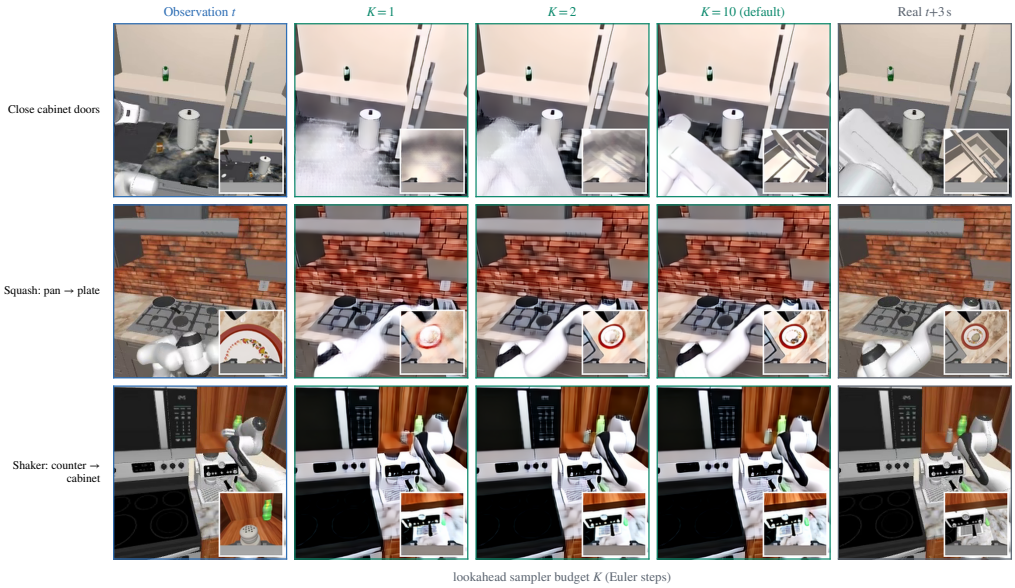

Figure 11: **Generated lookaheads vs. sampler budget** K (RoboCasa kitchen demonstration contexts, matched seeds per row; last column: the demonstration frame at $t{+}H_f$). Fidelity improves from $K{=}1$ to $K{=}10$ (the one-step lookahead smears the arm and the manipulated object, most visibly in the wrist insets), yet success barely changes over $K = 1$–30 (Table 3b). Layout as in Figure 10.

- *Stack cups* (single-arm): grasp the right cup by the rim and nest it in the left cup without toppling either.
- *Pick and place* (single-arm): pick up a $3{\times}3{\times}3$ cm foam cube and drop it into an open cup.
- *Remove cup cap* (single-arm): pull the fitted lid off a closed container and place it on the table.
- *Stack cups* (bimanual): stack the two cups using both arms without toppling them.
- *Bowl* (bimanual): one arm lifts the bowl and the other places a box into it.

Each task has 50 teleoperated demonstrations, and $\pi_{0.5}$, Fast-WAM, and GlanceWAM are fine-tuned on the same data. Each policy is evaluated over 50 trials per task (250 in total), with initial object poses perturbed within ± 8 cm and $\pm 30^{\circ}$. A trial fails if an object is dropped, a container topples, or the task takes longer than 30 s.

Rollouts and failure analysis. Figure 12 shows keyframes from successful GlanceWAM rollouts. On tasks with large object moves, the lookahead gives a clear target for the reach: single-arm *stack cups* and *remove cup cap* reach 94% and 88%, and the bimanual tasks 84% and 72% (vs. 66% and 64% for $\pi_{0.5}$). On *pick and place* (58%), most failures occur at the final grasp. With the VAE's $8\times$ spatial stride and $2\times$ patchification, the cube covers only 1–2 latent tokens, so the lookahead guides the arm toward the cube but cannot resolve sub-centimeter finger alignment; $\pi_{0.5}$'s robot-data pretraining gives it better local precision here.

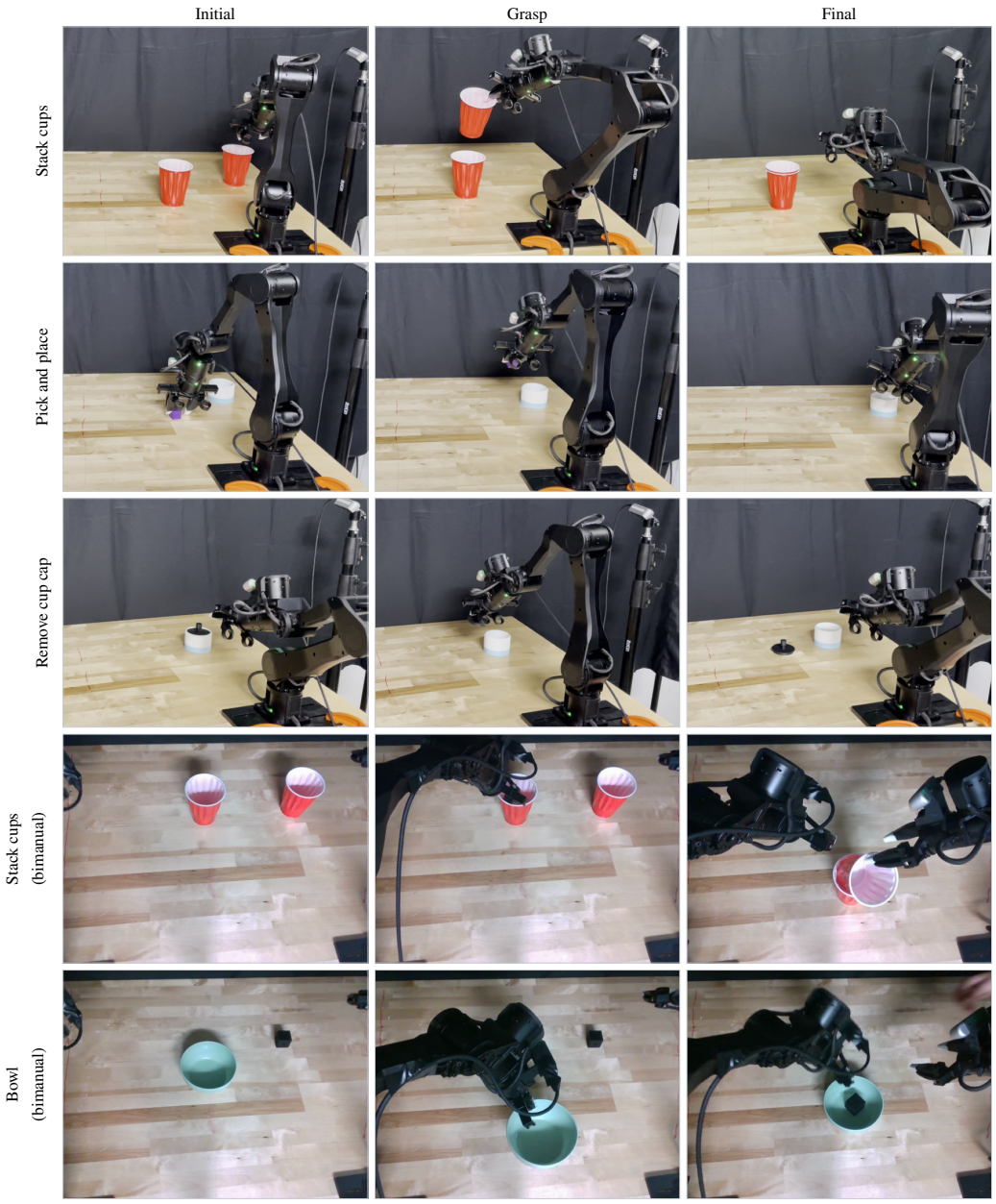

Figure 12: **Real-robot rollout keyframes.** Initial, grasp, and final frames from successful Glance-WAM rollouts. Single-arm rows (*stack cups*, *pick and place*, *remove cup cap*) are side-view recordings cropped to the workspace; bimanual rows (*stack cups*, *bowl*) are frames from the policy's top camera.

E REPRODUCIBILITY AND TRAINING DETAILS

Backbone and latent space. GlanceWAM builds on SkyReels-V2-DF (1.3B) (SkyReels Team, 2025). A raw video sequence $\mathbf{V} \in \mathbb{R}^{T\times H\times W\times 3}$ is compressed into a continuous latent representation $\mathbf{Z} \in \mathbb{R}^{T'\times H'\times W'\times C}$ by the spatiotemporal causal video VAE (Wan Team, 2025; SkyReels Team, 2025) with temporal downsampling ratio $p_t = 4$, spatial downsampling ratio $p_s = 8$, and latent dimension $C = 16$; the latents are then patchified with patch size $p_h = p_w = 2$, yielding $N = (H/16) \times (W/16)$ spatial tokens per frame. Language conditioning enters the DiT via cross-attention, while per-frame noise timesteps τ modulate intermediate features via adaptive layer normalization (adaLN) (Peebles & Xie, 2023).

Hyperparameters and training configuration. All models are trained with PyTorch on $4\times$ NVIDIA H200 (141 GB SXM5) GPUs using `bfloat16` mixed precision. We optimize the model using AdamW ($\beta_1 = 0.9, \beta_2 = 0.95, \epsilon = 10^{-8}$, weight decay 10^{-8}) with gradient norm clipping at 1.0. The video DiT backbone is trained with a base learning rate of 1.0×10^{-5}, while the action head is trained with a learning rate of 1.0×10^{-4}, both scheduled via cosine decay with 5000 warmup steps down to a minimum learning rate of 5.0×10^{-7}. Per-device batch size is 16 (64 total batch size). The action head cross-attends to DiT layers $\{5, 12, 19, 26\}$ with dropout probability $p = 0.2$. The lookahead dropout probability is $p = 0.1$. Flow timesteps are not sampled uniformly: video targets use $\tau = 5u/(1 + 4u)$ with $u = \mathrm{sigmoid}(n)$, $n \sim \mathcal{N}(0, 1)$ (logit-normal with timestep shift 5) (Esser et al., 2024), and actions use $\sigma = 1 - s + s\,x$ with $x \sim \mathrm{Beta}(1.5, 1)$ and $s = 0.999$ (Black et al., 2024a). On RoboCasa, the action loss is additionally reweighted by a bell-shaped function of σ that peaks at 0.5, vanishes at both ends, and has unit mean; LIBERO uses the unweighted loss. Checkpoints are evaluated using exponential moving average (EMA) with decay rate 0.999.