INSTANTWAM: INSTANT WORLD ACTION MODEL VIA ASYMMETRIC ONE-STEP DECODING

Anonymous authors
Paper under double-blind review

ABSTRACT

World Action Models (WAMs) regenerate every action plan from noise, even though much of the next plan was already predicted at the previous replan. This redundancy wastes computation and ignores the temporal structure of closed-loop control. We introduce **InstantWAM**, built on a simple asymmetric principle: generate the first plan from scratch, then iteratively update the already-in-motion future. InstantWAM carries the unexecuted suffix into the next planning window as a persistent generative state, perturbs it back onto the pretrained flow, and corrects it with a single observation-conditioned evaluation. The WAM's own action attention determines how long the inherited future remains useful. Repeated multi-step generation thereby reduces to single-step decoding without any training or modification to the frozen model. InstantWAM reduces average network function evaluations from 10 to 1.62, delivering $3.19\times$ faster planner calls and $2.47\times$ faster episode-level planning. On RoboTwin, it achieves 91.27% success versus 91.83% for 10-step FastWAM; on LIBERO, it reaches 95.75%. The future should not be discarded at every replan, but carried forward and incrementally updated.

"The end is where we start from."
—T. S. Eliot, *Little Gidding*

1 INTRODUCTION

World Action Models (WAMs) are designed to predict the future, yet their action decoders discard that future at every replan. Recent WAMs learn visual and action futures jointly (Zhou et al., 2026; Liu et al., 2026); FastWAM removes the visual rollout at test time while preserving the benefits of world-model pretraining (Yuan et al., 2026). Yet its action decoder still restarts from independent Gaussian noise and spends ten network function evaluations (NFEs) rebuilding a complete plan at every query. Once video generation disappears, repeated action generation becomes the bottleneck.

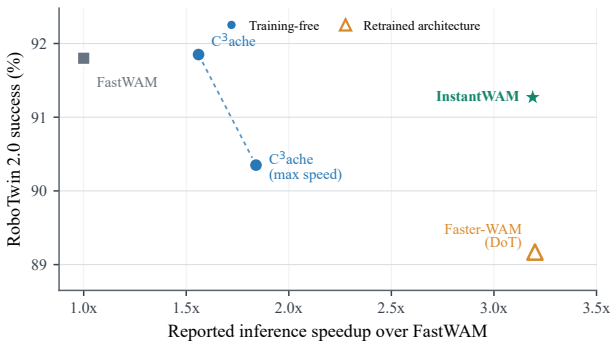

Figure 1: Published RoboTwin 2.0 accuracy–efficiency trade-offs (Zhao et al., 2026b; Ma et al., 2026). Horizontal positions use each source's reported speedup over its FastWAM reference; marker fill distinguishes training-free accelerators from retrained models.

We call this assumption *symmetric decoding*: the first plan and every subsequent plan are treated as identical generation problems. Closed-loop control is not symmetric. Chunking policies predict overlapping plans but execute only a prefix before replanning (Zhao et al., 2023; Chi et al., 2023; Black et al., 2025). After H actions, the unexecuted suffix already predicts part of the next plan. The first plan must be generated; thereafter, the future only needs to be updated. Restarting from noise discards both a valid prediction and the computation that produced it.

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.

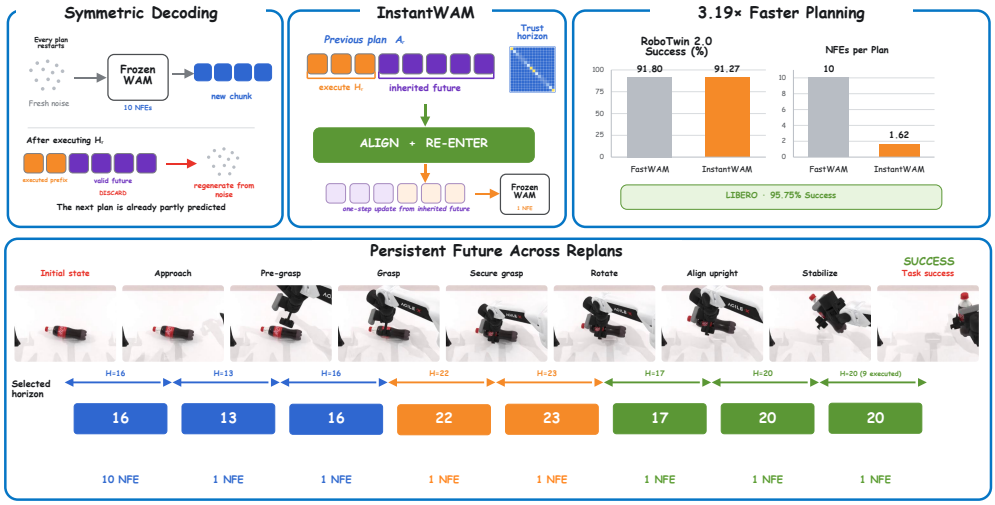

Figure 2: InstantWAM decodes the first plan with the frozen WAM, carries the unexecuted suffix forward as a persistent generative state, and updates it in one NFE at each subsequent replan. Native action attention sets the reuse horizon. This asymmetric schedule reduces average decoding cost from 10 to 1.62 NFEs while retaining FastWAM-level success.

Based on this principle, we introduce **InstantWAM**, which reorganizes WAM inference into an asymmetric process: full decoding once, followed by one-step decoding thereafter. Instead of treating an action chunk as a disposable output, InstantWAM carries its unexecuted suffix across replans as a persistent generative state. The previous prediction becomes the starting point for the subsequent future.

InstantWAM realizes this idea without retraining. It aligns the unused suffix with the new planning window, perturbs it back onto an intermediate state of the pretrained flow, and applies a single correction conditioned on the latest observation. The frozen WAM's own action attention reveals how long the inherited future remains coherent, providing a natural criterion for when to refresh it. Thus, the same model that predicts the plan also determines the persistence of that prediction.

This approach differs from training-based one-step policies, which alter the learned transport (Zhang et al., 2024; Li et al., 2026; Geng et al., 2025), and from learned warm starts or sequential flows, which add a source predictor or posterior transport (Scholz & Turner, 2025; Huang et al., 2026). InstantWAM learns neither: it reuses the future already produced by the frozen WAM. Its speedup comes not from approximating a new plan more aggressively, but from avoiding a generation problem that has already been partly solved.

InstantWAM reduces the average decoding cost from 10 to 1.62 NFEs and delivers 3.19× faster planner calls. On RoboTwin, it achieves 91.27% success, nearly matching the 91.83% of 10-step FastWAM; on LIBERO, it reaches 95.75%. As Figure 1 shows, this places InstantWAM on a stronger accuracy–efficiency frontier than existing FastWAM accelerators while requiring no retraining.

In summary, we make three contributions:

- We identify symmetric decoding as a hidden inefficiency in WAMs and formulate asymmetric decoding: generate the first plan, then update the future already in motion.
- We introduce InstantWAM, which turns the unexecuted plan into a persistent generative state and updates it in one step using a frozen WAM.
- We show that this change yields a 3.19× planning speedup while preserving multi-step-level performance on RoboTwin and LIBERO.

2 Related Work

World action models and inference acceleration. World-model policies connect visual prediction with robot control (Zhou et al., 2024; Hu et al., 2024); WAMs further couple video and action

generation (Zhu et al., 2025; Li et al., 2025; Ye et al., 2026; Zhou et al., 2026; Liu et al., 2026). FastWAM removes test-time video rollout, while newer accelerators cache cross-chunk residuals, sparsify future conditioning, or shorten the action module (Yuan et al., 2026; Zhao et al., 2026b; Oi et al., 2026; Zhao et al., 2026a; Ma et al., 2026). They reduce each decode; InstantWAM instead decodes fully once, then updates the predicted future.

Generative action policies and one-step decoding. Diffusion and flow policies iteratively refine action chunks (Ho et al., 2020; Song et al., 2021; Lipman et al., 2023; Zhao et al., 2023; Chi et al., 2023; Liu et al., 2023b). Fast solvers reduce integration steps (Lu et al., 2022), while consistency, distillation, flow-map, and shortcut objectives learn long updates (Song et al., 2023; Salimans & Ho, 2022; Boffi et al., 2024; Frans et al., 2025). Robot policies adapt these objectives through consistency training or self-distillation (Zhang et al., 2024; Li et al., 2026; Geng et al., 2025; Ahmed et al., 2026). InstantWAM retains the pretrained transport and starts from an aligned prediction, not noise.

Warm starts and sequential generative inference. Warm-start diffusion learns context-conditioned sources; Sequential Flow Matching learns posterior-to-predictive transport for streams (Scholz & Turner, 2025; Huang et al., 2026). InstantWAM learns neither: it aligns the frozen WAM's unexecuted suffix and performs one local correction in the original flow.

Adaptive execution horizons. PACE uses action geometry, dynamic horizon prediction learns a refresh rule, and AutoHorizon derives boundaries from attention (Nie et al., 2026; Zhao et al., 2026c; Wang et al., 2026). InstantWAM ties horizon selection to reuse: attention locates a reliable handover within the inherited suffix and thereby sets the reused state's lifetime.

3 Preliminaries

3.1 Action Chunking and Receding-Horizon Control

Given observation $\mathbf{o}_t$, an action-chunking policy predicts (Zhao et al., 2023; Chi et al., 2023)

$$\mathbf{a}_t = (a_t, a_{t+1}, ..., a_{t+T-1}) \in \mathbb{R}^{T \times d_a}, \tag{1}$$

where T is the prediction horizon and d_a the action dimension. Under receding-horizon control, the robot executes a prefix of $H \leq T$ actions and then replans. Larger H reduces policy queries and preserves open-loop coherence; smaller H admits earlier observation-conditioned correction.

3.2 Flow-Based Action Generation

We consider a WAM action decoder trained by flow matching (Lipman et al., 2023). Conditioned on $\mathbf{o}_t$, its velocity field $v_\theta(\mathbf{x}_\tau, \tau; \mathbf{o}_t)$ transports Gaussian noise at $\tau = 1$ toward actions at $\tau = 0$. With $1 = \tau_K > \cdots > \tau_0 = 0$, the K-step Euler solver evolves noise as follows:

$$\mathbf{x}_1 = \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{2}$$

$$\mathbf{x}_{\tau_{k-1}} = \mathbf{x}_{\tau_k} + (\tau_{k-1} - \tau_k) v_\theta(\mathbf{x}_{\tau_k}, \tau_k; \mathbf{o}_t). \tag{3}$$

Each velocity evaluation is one NFE; a K-step solver therefore spends K NFEs to produce $\widehat{\mathbf{a}}_t = \mathbf{x}_0$. A one-step update from noise must span the whole path although the learned field is local along its transport (Liu et al., 2023b; Bansal et al., 2026).

4 InstantWAM: Asymmetric One-Step Decoding

Standard WAM inference is symmetric across replanning rounds: every action chunk is initialized from noise and decoded with the same multi-step solver, even though consecutive chunks describe largely overlapping futures. InstantWAM breaks this symmetry. It fully decodes only the first plan and treats its unexecuted suffix as a persistent generative state in action space. At each subsequent replan, InstantWAM aligns this state with the new planning window, re-enters the pretrained flow at an intermediate timestep, and updates it under the latest observation with one network evaluation. The WAM's native action attention determines how long the inherited future remains useful.

4.1 PERSISTENT FUTURE STATE

Consecutive replanning rounds do not predict independent trajectories. Let the WAM produce the length-T chunk $\widehat{\mathbf{A}}_r = (\hat{a}_{r,0}, \hat{a}_{r,1}, ..., \hat{a}_{r,T-1})$ at round r. After the robot executes the first H_r actions, the suffix $(\hat{a}_{r,H_r}, ..., \hat{a}_{r,T-1})$ still describes the future of the same unfolding trajectory. InstantWAM therefore carries the predicted future itself across rounds. We align this suffix with the new planning window using the shift-and-pad operator $\mathcal{S}_{H_r}$:

$$\bar{\mathbf{A}}_{r+1} = \mathcal{S}_{H_r}(\widehat{\mathbf{A}}_r) = \left(\hat{a}_{r,H_r}, ..., \hat{a}_{r,T-1}, \underbrace{\hat{a}_{r,T-1}, ..., \hat{a}_{r,T-1}}_{H_r \text{ times}} \right). \tag{4}$$

We call $\bar{\mathbf{A}}_{r+1}$ the *temporal action prior*: the previous future realigned to the present and awaiting correction under new evidence.

Let $\mathbf{A}^\star_r = (a^\star_{r,0}, ..., a^\star_{r,T-1})$ denote the reference chunk produced by the standard full flow under the observation at round r. Over the full T-step chunk, let e_r denote previous-prediction error, $\Delta_r(H_r)$ shifted-reference drift, and D_{r+1} the resulting prior mismatch, all under one RMS norm; formal definitions are given in App. A.

Temporal consistency. Because shifting and padding are linear, the next prior satisfies the deterministic full-chunk bound

$$D_{r+1} \le \kappa_{H_r} e_r + \Delta_r(H_r), \qquad \kappa_{H_r} \le \sqrt{H_r + 1}. \tag{5}$$

Here κ_{H_r} accounts for replicating the final action in the padded tail, while $\Delta_r(H_r)$ includes the complete shifted-reference mismatch. The inherited prior is therefore not an arbitrary warm start, but a state with an explicit temporal correspondence to the new planning window.

4.2 RE-ENTERING THE PRETRAINED FLOW

The inherited future must remain reusable without becoming immutable. Copying $\bar{\mathbf{A}}_{r+1}$ would ignore new evidence, while returning to noise would erase valid temporal structure. InstantWAM instead maps the inherited endpoint back to the pretrained flow at the re-entry timestep σ:

$$\mathbf{X}_{\sigma,r+1} = (1-\sigma)\bar{\mathbf{A}}_{r+1} + \sigma\boldsymbol{\epsilon}_r, \qquad \boldsymbol{\epsilon}_r \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{6}$$

where $\sigma \in (0, 1)$ trades temporal inheritance for correction. Let $\mathbf{Y}^\star_{\sigma,r+1} = (1-\sigma)\mathbf{A}^\star_{r+1} + \sigma\boldsymbol{\epsilon}_r$ be the linear reference constructed with the same Gaussian realization, and let $\mathbf{X}^\star_{\sigma,r+1}$ be the actual intermediate state of the standard full-flow trajectory. We use $\eta_{\sigma,r+1} := \|\mathbf{Y}^\star_{\sigma,r+1} - \mathbf{X}^\star_{\sigma,r+1}\|_{\mathrm{RMS}}$ for the interpolation defect.

Shared-noise attenuation. Over the full chunk, the shared noise cancels exactly; a triangle inequality then relates the warm state to the actual full-flow state:

$$\left\|\mathbf{X}_{\sigma,r+1} - \mathbf{Y}^\star_{\sigma,r+1}\right\|_{\mathrm{RMS}} = (1-\sigma)D_{r+1}, \tag{7}$$

$$\left\|\mathbf{X}_{\sigma,r+1} - \mathbf{X}^\star_{\sigma,r+1}\right\|_{\mathrm{RMS}} \le (1-\sigma)D_{r+1} + \eta_{\sigma,r+1}. \tag{8}$$

The inherited discrepancy is therefore attenuated before the WAM is queried. Temporal inheritance shortens the transport required at the new round: InstantWAM does not ask one evaluation to generate an action chunk from noise, but only to correct a future whose coarse structure has already been produced. It thereby reuses the pretrained flow without learning a new source distribution (Scholz & Turner, 2025; Huang et al., 2026) at test time.

4.3 ONE-STEP DECODING AND ITS STABILITY

Once the inherited future has re-entered the intermediate flow state, only the remaining transport must be evaluated. Conditioned on the new observation $\mathbf{o}_{r+1}$, the pretrained WAM performs one frozen endpoint update,

$$\widehat{\mathbf{A}}_{r+1} = \mathbf{X}_{\sigma,r+1} - \sigma v_\theta(\mathbf{X}_{\sigma,r+1}, \sigma; \mathbf{o}_{r+1}). \tag{9}$$

After one standard K-NFE bootstrap, every later plan applies Eq. (9) with exactly one NFE.

Write the learned endpoint update as $F_{\sigma,\mathbf{o}}(\mathbf{x}) := \mathbf{x} - \sigma v_\theta(\mathbf{x}, \sigma; \mathbf{o})$. We characterize correction over the region traversed by the warm start. Assume $F_{\sigma,\mathbf{o}}$ is locally $K_{\sigma,r+1}$-Lipschitz between the full warm and reference intermediate states, the learned velocity has error at most δ_{r+1} at the reference state, and the material acceleration of the ideal path is bounded by M_{r+1}. Full statements are given in App. A, together with the resulting finite-error bounds.

One-step endpoint stability. Let $c_{\sigma,r+1} := \sigma\delta_{r+1} + \frac{1}{2}M_{r+1}\sigma^2$ collect the local model and one-step integration residuals. The corrected endpoint obeys

$$e_{r+1} \le K_{\sigma,r+1}\big((1-\sigma)D_{r+1} + \eta_{\sigma,r+1}\big) + c_{\sigma,r+1}. \tag{10}$$

The first term is the propagated initialization mismatch; the residual covers model and one-step integration errors. $K_{\sigma,r+1}$ is the local corrective gain along the neighborhood traversed by the warm start under the new observation, rather than a global stability constant.

Sequential replanning stability. Combining temporal carry-over with the local correction gives

$$e_{r+1} \le \alpha_r e_r + \beta_r. \tag{11}$$

Here $\alpha_r := (1-\sigma)K_{\sigma,r+1}\kappa_{H_r}$ and $\beta_r := K_{\sigma,r+1}((1-\sigma)\Delta_r(H_r) + \eta_{\sigma,r+1}) + c_{\sigma,r+1}$ summarize amplification and fresh round-wise disturbance. If $\alpha_r \le \bar{\alpha} < 1$ and $\beta_r \le \bar{\beta}$ uniformly over rounds, then

$$\limsup_{r\to\infty} e_r \le \frac{\bar{\beta}}{1-\bar{\alpha}}. \tag{12}$$

The full finite-round bound and a distributional corollary are deferred to App. A.

These relations trace full-chunk prediction error and shifted-reference drift through shared-noise attenuation and one frozen WAM correction. The asymptotic bound is conditional: finite-sample diagnostics do not establish uniform $\alpha_r < 1$.

4.4 Attention-Guided Lifetime of the Inherited Future

A temporal prior is useful only while aligned with the unfolding task. Replanning too early interrupts a coherent motion and shortens the reusable future; replanning too late carries a stale prediction forward. InstantWAM thus seeks a handoff where the current fragment is complete and the remaining suffix still provides a useful start for one-step decoding.

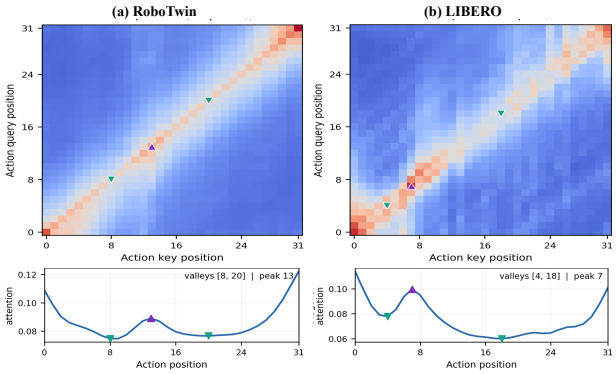

Figure 3: Attention heatmaps and diagonals reveal two candidate handoff valleys on RoboTwin and LIBERO.

The frozen WAM's action attention provides a model-native signal for this handoff. It exposes the temporal dependencies formed during chunk generation: strongly coupled tokens form coherent sub-motions, while weaker connections indicate natural refresh points. The relevant boundary is therefore in the organization of the predicted future, not only in adjacent-action geometry.

Across replans, the attention diagonal exhibits a stable W-shaped, high–low–high–low–high profile. Its two valleys separate an interior coupled fragment from the endpoint anchors. This non-monotonicity makes a first-transition rule brittle: an AutoHorizon-style selector can stop at the first trough around actions 11–12, or skip it and drift toward the maximum horizon under another calibration (Wang et al., 2026). The interior rise persists across RoboTwin profiles conditioned on $H \in \{12, 18, 24\}$, making its neighboring valleys natural handoff candidates (Du et al., 2022; Liu et al., 2023c).

PACE-style action geometry (Nie et al., 2026) captures visible motion changes, but a large change may still belong to one jointly generated sub-motion, while a smooth trajectory may cross an internal temporal boundary. InstantWAM instead selects the reuse horizon from the model that produced the inherited future.

From row-normalized action attention $\mathbf{M}_r \in \mathbb{R}^{T\times T}$, averaged over selected layers and heads, we smooth the main diagonal and select the execution horizon through this restricted local-valley score:

$$d_{r,i} = M_{r,ii}, \qquad s_r(H) = \frac{1}{2\rho+1}\sum_{j=-\rho}^{\rho} d_{r,H-1+j}, \qquad H_r^\star = \arg\min_{H\in\mathcal{H}} s_r(H), \tag{13}$$

using $\rho = 2$. Rather than terminating at the first detected transition, we compare the smoothed valley score inside a reuse-compatible candidate set $\mathcal{H}$. The selected $H_r^\star$ lets the current coupled fragment finish before the remaining actions are aligned, perturbed, and passed to the next one-step decode. Experiments in § 5.5 compare this criterion with fixed horizons, PACE-inspired replanning, and AutoHorizon.

4.5 Complete Inference and Cost

Algorithm 1 InstantWAM Inference

Require: Pretrained WAM v_θ; chunk length T; bootstrap steps K; noise timestep σ; candidate horizons $\mathcal{H}$; radius ρ

Observe $\mathbf{o}_0$; sample $\boldsymbol{\epsilon}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
$(\widehat{\mathbf{A}}_0, \mathbf{M}_0) \leftarrow \mathrm{Solve}_K(v_\theta, \boldsymbol{\epsilon}_0; \mathbf{o}_0)$
$r \leftarrow 0$
while the task is not terminated **do**
  Select $H_r^\star$ from the smoothed attention diagonal using Eq. (13)
  Execute $\widehat{\mathbf{A}}_r[0 : H_r^\star]$; observe $\mathbf{o}_{r+1}$
  if the task is terminated **then**
    break
  end if
  $\bar{\mathbf{A}}_{r+1} \leftarrow \mathcal{S}_{H_r^\star}(\widehat{\mathbf{A}}_r)$; $\boldsymbol{\epsilon}_{r+1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{X}_\sigma \leftarrow (1-\sigma)\bar{\mathbf{A}}_{r+1} + \sigma\boldsymbol{\epsilon}_{r+1}$
  $(\mathbf{u}_{r+1}, \mathbf{M}_{r+1}) \leftarrow \mathrm{WAM}_\theta(\mathbf{X}_\sigma, \sigma; \mathbf{o}_{r+1})$ {one NFE}
  $\widehat{\mathbf{A}}_{r+1} \leftarrow \mathbf{X}_\sigma - \sigma\mathbf{u}_{r+1}$; $r \leftarrow r+1$
end while

Ensure: Executed robot trajectory

An episode begins with one K-NFE bootstrap. Each later round selects $H_r^\star$, executes that prefix, and carries the unused suffix into the next round. The shifted suffix is perturbed at the re-entry timestep and corrected once using Eq. (9); the complete procedure is summarized step by step in Algorithm 1.

This yields an asymmetric decoding budget. If an episode contains R planning rounds, its amortized cost is $[K + (R-1)]/R = 1 + (K-1)/R$ NFEs, approaching one as R grows. Only the initial round performs multi-step integration; every later round performs exactly one frozen WAM evaluation. Operationally, the cross-round state consists only of the previous action chunk and its native attention map. The frozen decoder receives a shifted, perturbed suffix, while attention selects H without an extra NFE. Each update therefore uses one suffix alignment, one Gaussian interpolation, one WAM evaluation, and one attention-diagonal reduction. The acceleration comes from reorganizing the generation process itself: generate the future once, then keep it in motion.

5 Experiments

Our experiments ask three questions: (i) can a temporal prior turn multi-step WAM inference into a training-free one-step correction; (ii) does the gain indeed come from starting closer and locally correcting the inherited plan, as measured in Figures 5 and 6; and (iii) can action attention decide when the inherited future should be replanned without adding a learned horizon controller?

5.1 Experimental Setup

We evaluate FastWAM without retraining on RoboTwin 2.0 (Chen et al., 2025) and LIBERO (Liu et al., 2023a): the first chunk uses its 10-NFE solver and every later chunk uses Algorithm 1. Fixed horizons are $H = 24$ and $H = 10$; complete evaluation and ablation protocols are in App. B.

5.2 Training-Free One-Step Inference

InstantWAM reaches 91.27% on RoboTwin, within 0.56 points of 10-NFE FastWAM; on saturated LIBERO, it reaches 95.75% versus 97.60%.

(a) RoboTwin 2.0

Method	Emb. PT	Clean	Rand.	Avg.
OpenVLA	–	–	–	–
π_0	✓	65.92 −25.96	58.40 −33.38	62.2 −29.6
$\pi_{0.5}$	✓	82.74 −9.14	76.76 −15.02	79.8 −12.0
LingBot-VA	✓	92.90 +1.02	91.50 −0.28	**92.2** +0.4
Motus	✓	88.66 −3.22	87.02 −4.76	87.8 −4.0
FastWAM (10 NFE)	–	91.88	91.78	91.83
Ours, fixed	–	91.20 −0.68	89.60 −2.18	90.40 −1.43
Ours, attention	–	91.48 −0.40	91.06 −0.72	**91.27** −0.56

(b) LIBERO

Method	Spatial	Object	Goal	Long	Avg.
OpenVLA	84.7 −13.5	88.4 −11.6	79.2 −17.8	53.7 −41.5	76.5 −21.1
π_0	96.8 −1.4	98.8 −1.2	95.8 −1.2	85.2 −10.0	94.1 −3.5
$\pi_{0.5}$	98.8 +0.6	98.2 −1.8	98.0 +1.0	92.4 −2.8	96.9 −0.7
LingBot-VA	98.5 +0.3	99.6 −0.4	97.2 +0.2	98.5 +3.3	**98.5** +0.9
Motus	96.8 −1.4	99.8 −0.2	96.6 −0.4	97.6 +2.4	97.7 +0.1
FastWAM (10 NFE)	98.2	100.0	97.0	95.2	97.6
Ours, fixed	96.2 −2.0	99.0 −1.0	95.6 −1.4	92.4 −2.8	95.80 −1.80
Ours, attention	96.0 −2.2	99.0 −1.0	95.2 −1.8	92.8 −2.4	**95.75** −1.85

Table 1: Overall success on RoboTwin 2.0 and LIBERO. Published baselines are from FastWAM; signed values are percentage-point changes from FastWAM. See App. C for setup.

5.3 Reuse the Future, but Re-enter the Flow

We ablate temporal reuse and flow re-entry against FastWAM's published 10-NFE results.

Method	Removed	RoboTwin2.0-15	Spatial	Object	Goal	Long	Avg.
FastWAM (10 NFE)	–	84.80	98.20	100.00	97.00	95.20	97.60
Fresh Noise	temporal reuse	57.83 −26.97	96.60 −1.60	98.80 −1.20	94.60 −2.40	91.40 −3.80	95.35 −2.25
Direct Prior	flow re-entry	44.50 −40.30	95.60 −2.60	98.20 −1.80	94.20 −2.80	92.80 −2.40	95.20 −2.40
Ours, fixed	none	**81.50** −3.30	96.20 −2.00	**99.00** −1.00	**95.60** −1.40	92.40 −2.80	**95.80** −1.80

Table 2: Structural ablations. Signed values are percentage-point changes from FastWAM; each one-step variant removes one component. See App. C for setup.

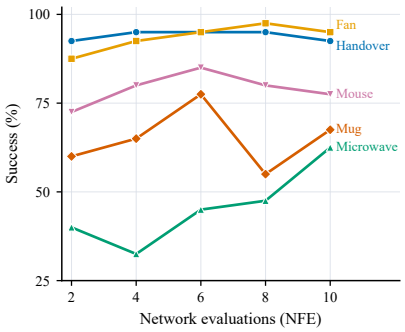

Figure 4: RoboTwin success versus NFE; the best budget is task-dependent.

On RoboTwin, replacing the inherited future with Fresh Noise drops success to 57.83%, while Direct Prior falls to 44.50%. The former removes temporal reuse; the latter keeps continuity but cannot absorb the new observation and execution error. Re-entering the flow restores 81.50%, close to the 84.80% 10-NFE reference. LIBERO stays near saturation, so the structural gap is smaller. Figure 4 adds a complementary view: extra evaluations are not uniformly helpful, and the best budget varies by task. The gain therefore does not come from spending more NFEs. It comes from an asymmetric computation pattern: solve the future once, carry it forward, and spend one model evaluation to correct it. This is the source of the one-step speedup.

Closer source. Matched-reference analysis compares full-chunk intermediate states over 1,374 warm replans. Median discrepancy to the teacher state is 0.0893 for the shifted warm state, versus 0.2233 for the unshifted state and 0.8474 for Gaussian noise (Figure 5).
The unshifted and noise medians are therefore 2.50× and 9.49× larger. The shifted state is closer on 76.1% of shifted–unshifted pairs and every shifted–noise pair. This isolates the benefit of temporal alignment before endpoint correction. Some replans already have a small unshifted discrepancy, while alignment removes the large stale-state tail. Thus, the shifted state provides a closer starting point across the replan population, rather than improving only a few outliers in the evaluation set.

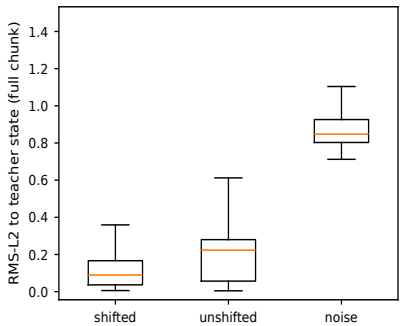

Figure 5: Intermediate-state alignment to the teacher trajectory.

Complete-chunk carry-over. Let $B_{r+1}^{\text{prior}} = \|\mathcal{S}_H(\widehat{\mathbf{A}}_r - \mathbf{A}_r^\star)\|_{\text{RMS}} + \Delta_r(H)$. All 1,374 points satisfy $D_{r+1} \le B_{r+1}^{\text{prior}}$, with median ratio 0.892 (Figure 6(a)). This is the direct data-dependent triangle bound underlying the looser operator bound in Eq. (5).

The scatter uses complete chunks, so inherited, padded, and shifted-reference components all enter before endpoint correction is evaluated.

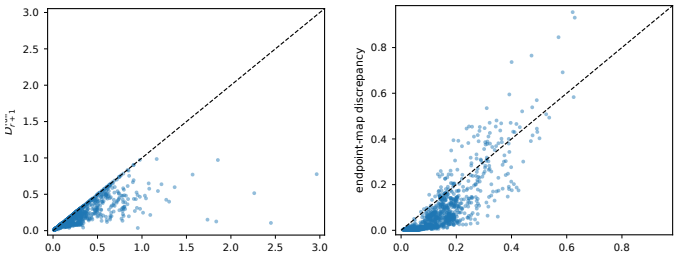

Figure 6: Matched-reference diagnostics. **(a)** All 1,374 replans satisfy $D_{r+1} \leq B_{r+1}^{\text{prior}}$; the median ratio is 0.892. **(b)** Endpoint discrepancy decreases along 92.4% of observed directions; median sampled directional gain $\widehat{K}_r$ is 0.236.

Local correction. At the deployed timestep, endpoint discrepancy decreases along 92.4% of observed directions (Figure 6(b)); the median sampled directional gain is $\widehat{K}_r = 0.236$. Thus, most observed directions improve, but $\widehat{K}_r$ is neither a Lipschitz upper bound nor a stability guarantee. Definitions and quantiles are given in Table 6; the next section describes chained round-to-round errors. This separates one-step correction from round-to-round dynamics.

5.4 Closed-Loop Error Propagation

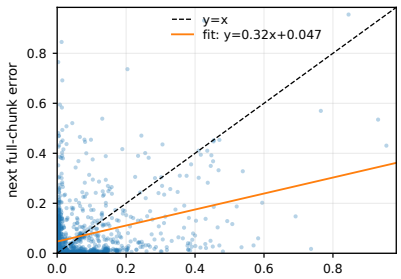

Figure 7: Round-to-round full-chunk error across successive warm-start replans.

Warm starts feed one corrected plan into the next, motivating a round-to-round view of full-chunk error. On the fixed-$H = 18$ teacher subset, the fit across 1,284 successive warm-to-warm pairs is $e_{r+1} = 0.320e_r + 0.0471$.

The sub-unit fitted slope describes a non-explosive average trend over the observed range. The positive intercept and low R^2 reflect fresh round-level disturbance and substantial trajectory heterogeneity, while the scatter records the broad variation hidden by the mean relation. Together, the fit and scatter provide a compact finite-sample summary of how inherited discrepancies evolve after each correction. Most pairs cluster in the low-error region, while sparse high-error transitions contribute much of the dispersion and motivate reading the fit alongside the error distribution.

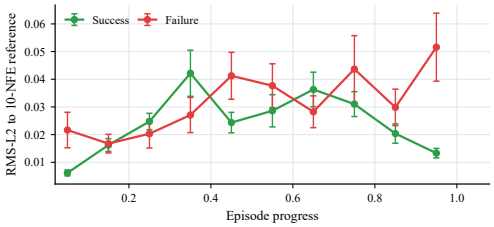

Figure 8: Episode-progress error (mean ± standard error). Successful trajectories remain stable, whereas failures grow more variable after mid-episode and diverge near completion.

Where reuse becomes difficult. Binning the same trajectories by progress reveals a localized failure pattern. Successful episodes show no late-stage rise and finish at their lowest post-bootstrap error, whereas failures become less accurate and more variable near completion. The separation begins only after mid-episode, isolating the regime where a trust or fallback mechanism is most useful. Because the curves separate only late, the failure is phase-local rather than global: discarding the inherited future throughout the episode would forfeit useful reuse. The evidence instead motivates a state-dependent trust check near completion.

5.5 Attention-Guided Replanning

For a length-T chunk, replanning after H actions leaves $T - H$ genuinely predicted actions in the aligned suffix; the remaining H positions require padding. Replanning moderately before the nominal FastWAM boundary therefore offers a longer informative prior. Replanning too early, however, can interrupt a coherent motion and increases policy calls. We use action attention to resolve this trade-off inside the restricted search window.

Across RoboTwin, attention-guided replanning improves success from 90.40% to 91.27% while reducing the average horizon from 24 to 18.21; the gain holds in clean (91.48%) and randomized (91.06%) conditions. On LIBERO, it selects $H \in [8, 10]$ (mean 9.67) while matching saturated fixed-horizon performance (95.75% versus 95.80%).

On the same 15-task subset, fixed $H = 12$, $H = 18$, and $H = 24$ reach 83.73%, 84.57%, and 82.93%; attention reaches 84.83%. The matched-control analysis is separate: among 2,738 shared accepted-seed episodes, attention exceeds fixed $H = 18$ by 0.88 points (task-cluster bootstrap 95% CI $[-1.41, 2.40]$). Thus attention matches $H = 18$ while choosing H online; see App. G for cross-task stability.

Dataset	Replanning	Avg. H	Episodes	Success
RoboTwin2.0-Full	Fixed $H = 24$	24	10,000	90.40
RoboTwin2.0-Full	Attention, 12–24	18.21 −5.79	10,000	**91.27** +0.87
RoboTwin2.0-15	Fixed $H = 24$	24	3,000	82.93
RoboTwin2.0-15	Fixed $H = 18$	18 −6.00	3,000	84.57 +1.64
RoboTwin2.0-15	Fixed $H = 12$	12 −12.00	3,000	83.73 +0.80
RoboTwin2.0-15	Attention, 12–24	18.38 −5.62	3,000	**84.83** +1.90
LIBERO-40	Fixed $H = 10$	10	2,000	**95.80**
LIBERO-40	Attention, 8–10	9.67 −0.33	2,000	95.75 −0.05

Table 3: Fixed versus attention-guided replanning. Small signed values are differences from the fixed baseline at the start of each dataset block; lower H is marked as an efficiency gain. RoboTwin2.0-Full covers all 100 task–condition cells; RoboTwin2.0-15 uses 15 randomly selected tasks, 30 cells, and 100 episodes in every task–condition cell. See App. C for setup.

Under the same frozen checkpoint and $10 \rightarrow 1$ warm-start schedule, Table 4 compares our attention criterion with alternative online replanning rules. Attention performs best, exceeding PACE-inspired (Nie et al., 2026), tuned AutoHorizon, and official AutoHorizon (Wang et al., 2026) by 3.83, 2.50, and 12.33 points. The official selector chooses $H = 32$ on 99.34% of replans, nearly reducing to a fixed maximum horizon. Because seeds are not asserted to be paired across campaigns, these are matched-scale rather than paired comparisons.

Replanning rule	Clean	Rand.	Avg. H	NFE/step	Success
PACE-inspired	83.33	81.67	19.73	0.0808	82.50
AutoHorizon (official)	75.00 −8.33	73.00 −8.67	31.85 +12.12	0.0581 −0.0227	74.00 −8.50
AutoHorizon-tuned	85.00 +1.67	82.67 +1.00	17.36 −2.37	0.0893 +0.0085	83.83 +1.33
Ours, attention	**87.67** +4.34	**85.00** +3.33	18.32 −1.41	0.0873 +0.0065	**86.33** +3.83

Table 4: Comparison of dynamic replanning rules on the fixed 15-task RoboTwin subset. AutoHorizon includes both released and task-adapted tuned variants. See App. C for setup.

5.6 Inference Efficiency

FastWAM uses 10 evaluations per replan; ours uses 10 at bootstrap and one thereafter (1.62 NFEs on average; 83.8% fewer). Table 5 reports call latency of 590.6 → 185.2 ms (3.19×), episode planning of 6.814 → 2.755 s (2.47×), and per-step latency of 25.676 → 10.571 ms (2.43×).

Method	Call (ms)	Plan (s/ep)	Plan (ms/step)
FastWAM native	590.6	6.814	25.676
Ours, fixed $H = 24$	176.5 −414.1	**2.195** −4.619	**7.680** −17.996
Ours, fixed $H = 18$	**169.6** −421.0	2.620 −4.194	9.677 −15.999
Ours, attention	185.2 −405.4	2.755 −4.059	10.571 −15.105

Table 5: End-to-end planning latency. Signed values are reductions from FastWAM; call latency amortizes the 10-NFE bootstrap. Metrics include replan frequency.

Task-wise robustness. Across all 50 RoboTwin tasks, attention-guided replanning improves over fixed $H = 24$ by 0.28 points on clean evaluations and 1.46 points under randomization. The task-level spread in Figure 10 (App. G) shows a larger aggregate gain under distribution shift, although individual tasks remain heterogeneous. Thus attention improves the aggregate result without hiding task-level variation.

6 Conclusion and Limitations

Robot motion does not restart when a policy replans; action generation need not either. We reuse the unexecuted future as a generative state: align it, re-enter the frozen flow, correct once, and let attention set its trust horizon. This turns cross-round overlap into computation while retaining near-multi-step performance at 1.62 average NFEs. Unlike independent-sample warm starts (Scholz & Turner, 2025) and horizon selectors (Zhao et al., 2026c; Wang et al., 2026), ours reuses computation across rounds. Abrupt changes can invalidate the prior, motivating robust rejection and broader transfer. Our evidence is limited to one frozen WAM backbone and simulated benchmarks, so robustness to new embodiments, sensors, and contact-rich physical interaction remains open. A natural next step is an uncertainty-aware gate that detects when the inherited future is no longer trustworthy and selectively returns to full decoding. Testing this mechanism on real robots would clarify when temporal reuse remains reliable under latency, observation noise, and unmodeled disturbances.

AI USE STATEMENT

Generative AI tools were used for language editing, literature organization, and assistance with analysis and typesetting code. The author manually checked the cited papers, quantitative claims, mathematical content, and compiled artifact, and takes responsibility for the complete paper.

REPRODUCIBILITY STATEMENT

The main text specifies the inference equations, evaluation protocol, and all reported hyperparameters, including the complete inference algorithm. The appendix provides detailed evaluation protocols, derivations, per-task results, sensitivity analyses, and attention diagnostics; the supplementary material provides the core inference implementation.

REFERENCES

Riad Ahmed, Sujosh Nag, Moniruzzaman Akash, Mostafa Hussein, and Momotaz Begum. Trajectory-consistent flow matching for robust visuomotor policy learning. *arXiv preprint arXiv:2605.08511*, 2026.

Vansh Bansal, Saptarshi Roy, Alessandro Rinaldo, and Purnamrita Sarkar. On the convergence and straightness of rectified flow. In *International Conference on Artificial Intelligence and Statistics*, 2026.

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

Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. Flow map matching. *arXiv preprint arXiv:2406.07507*, 2024.

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

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*, 2023.

Dazhao Du, Bing Su, Yu Li, Zhongang Qi, Lingyu Si, and Ying Shan. Do we really need temporal convolutions in action segmentation? *arXiv preprint arXiv:2205.13425*, 2022.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *International Conference on Learning Representations*, 2025.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

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.

Yinan Huang, Hans Hao-Hsun Hsu, Junran Wang, Bo Dai, and Pan Li. Accelerated sequential flow matching: A bayesian filtering perspective. *arXiv preprint arXiv:2602.05319*, 2026.

Shaolong Li, Lichao Sun, and Yongchao Chen. One-step flow policy: Self-distillation for fast visuomotor policies. *arXiv preprint arXiv:2603.12480*, 2026.

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

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

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

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023b.

Yushan Liu, Peibo Sun, Shoujie Li, Yifan Xie, Lingfeng Zhang, Xintao Chao, Shiyuan Dong, Fang Chen, Xiao-Ping Zhang, and Wenbo Ding. OA-WAM: Object-addressable world action model for robust robot manipulation. *arXiv preprint arXiv:2605.06481*, 2026.

Zhichao Liu, Leshan Wang, Desen Zhou, Jian Wang, Songyang Zhang, Yang Bai, Errui Ding, and Rui Fan. Temporal segment transformer for action segmentation. *arXiv preprint arXiv:2302.13074*, 2023c.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems*, 2022.

Liheng Ma, Rui Heng Yang, Zhanguang Zhang, Mateo Clemente, Ziwen Hu, Tongtong Cao, and Yingxue Zhang. Faster-WAM: Do world action models need deep action modules? *arXiv preprint arXiv:2608.02365*, 2026.

Junnan Nie, Jiayi Li, Jiachen Zhang, Junyi Lao, Chenghao Liu, Tianle Zhang, Liang Lin, and Songfang Huang. PACE: Phase-aware chunk execution for robot policies with action chunking. *arXiv preprint arXiv:2606.00537*, 2026.

Ryuji Oi, Hikari Otsuka, Kosuke Matsushima, Yuki Ichikawa, Masato Motomura, Tatsuya Kaneko, and Daichi Fujiki. ActionCache: Training-free acceleration for vision-language-action models with action caching and refinement. *arXiv preprint arXiv:2607.06370*, 2026.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

Jonas Scholz and Richard E. Turner. Warm starts accelerate conditional diffusion. *arXiv preprint arXiv:2507.09212*, 2025.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, 2023.

Haoxuan Wang, Gengyu Zhang, Yan Yan, Ramana Rao Kompella, and Gaowen Liu. VLA knows its limits: Adaptive execution horizons for robot policies. *arXiv preprint arXiv:2602.21445*, 2026.

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*, 2026.

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

Qinglun Zhang, Zhen Liu, Haoqiang Fan, Guanghui Liu, Bing Zeng, and Shuaicheng Liu. FlowPolicy: Enabling fast and robust 3d flow-based policy via consistency flow matching for robot manipulation. *arXiv preprint arXiv:2412.04987*, 2024.

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

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

Weisen Zhao, Lam Nguyen, Zhicong Lu, and Yuzhang Shang. C^3ache: Accelerating world action models with cross inference chunk cache. *arXiv preprint arXiv:2606.08962*, 2026b.

Yuchi Zhao, Miroslav Bogdanovic, Arjun Sohal, Liyu Tao, Kourosh Darvish, Alán Aspuru-Guzik, Florian Shkurti, and Animesh Garg. Dynamic execution horizon prediction for chunk-based robot policies. *arXiv preprint arXiv:2606.11408*, 2026c.

Pengfei Zhou, Shengcong Chen, Di Chen, Jiaxu Wang, Rongjun Jin, Bingwen Zhu, Yike Pan, Songen Gu, Shufeng Nan, Xingyu Qiu, et al. τ_0-WM: A unified video-action world model for robotic manipulation. *arXiv preprint arXiv:2606.01027*, 2026.

Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. RoboDreamer: Learning compositional world models for robot imagination. *arXiv preprint arXiv:2404.12377*, 2024.

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.

APPENDIX

A DERIVATION OF THE ERROR CHAIN

We analyze fidelity to the standard full-flow WAM on the complete action chunk. All norms are taken in normalized action space and conditioned on the observation. For a random (or deterministic) matrix $Z \in \mathbb{R}^{m\times d_a}$, define

$$\|Z\|_{\mathrm{RMS},m} \coloneqq \left(\mathbb{E}\frac{\|Z\|_F^2}{md_a}\right)^{1/2}. \tag{14}$$

For deterministic Z, the expectation is omitted equivalently. The main proof always uses $m = T$ and suppresses this subscript. Shorter-block norms appear only in the optional overlap–padding decomposition below. The expectation may subsequently include observations, and no independence between consecutive plans or injected noise is required.

A.1 ASSUMPTIONS AND NOTATION

The three round-wise quantities in the main text are defined on all T actions:

$$e_r \coloneqq \left\|\widehat{\mathbf{A}}_r - \mathbf{A}_r^\star\right\|_{\mathrm{RMS}}, \tag{15}$$

$$\Delta_r(H_r) \coloneqq \left\|\mathcal{S}_{H_r}(\mathbf{A}_r^\star) - \mathbf{A}_{r+1}^\star\right\|_{\mathrm{RMS}}, \tag{16}$$

$$D_{r+1} \coloneqq \left\|\mathcal{S}_{H_r}(\widehat{\mathbf{A}}_r) - \mathbf{A}_{r+1}^\star\right\|_{\mathrm{RMS}}. \tag{17}$$

The derivation uses the following local sufficient conditions.

1. **Shared randomness.** The warm interpolant and linear reference use the same realization $\boldsymbol{\epsilon}_r$ over the complete chunk.
2. **Complete-chunk update regularity.** On the segment joining $\mathbf{X}_{\sigma,r}^\star$ and $\mathbf{X}_{\sigma,r}$ in $\mathbb{R}^{T\times d_a}$, the endpoint map $F_{\sigma,\mathbf{o}_r}$ is $K_{\sigma,r}$-Lipschitz. Thus all cross-token effects, including those from padded inputs through bidirectional attention, are included in one map. If v_θ is locally $L_{\theta,r}$-Lipschitz, then $K_{\sigma,r} \le 1 + \sigma L_{\theta,r}$.
3. **Bounded local model and integration errors.** At the reference intermediate state, $\|v_\theta - v^\star\|_{\mathrm{RMS}} \le \delta_r$. Along the ideal remaining path, $\|\mathcal{D}_t v^\star\|_{\mathrm{RMS}} \le M_r$.
4. **Finite second moments.** Reference chunks, predictions, noise, and velocity evaluations have finite second moments.

These assumptions are local to the states and observations under analysis; consecutive reference chunks may differ.

A.2 COMPLETE-CHUNK TEMPORAL-PRIOR PROPAGATION

Shift-and-pad operator. Fix r, write $H = H_r$, and let $Z = (z_0, ..., z_{T-1})$. By construction,

$$\|\mathcal{S}_H(Z)\|_F^2 = \sum_{i=H}^{T-2}\|z_i\|_2^2 + (H+1)\|z_{T-1}\|_2^2 \le (H+1)\|Z\|_F^2. \tag{18}$$

The input and output have the same Td_a normalization, hence

$$\|\mathcal{S}_H(Z)\|_{\mathrm{RMS}} \le \kappa_H\|Z\|_{\mathrm{RMS}}, \qquad \kappa_H \coloneqq \|\mathcal{S}_H\|_{\mathrm{op}} = \sqrt{H+1}. \tag{19}$$

Equality is attained when only the final input action is nonzero.

Propagation bound. Linearity gives the complete-chunk decomposition

$$\begin{aligned}&\mathcal{S}_H(\widehat{\mathbf{A}}_r) - \mathbf{A}_{r+1}^\star \\ &\quad= \mathcal{S}_H(\widehat{\mathbf{A}}_r - \mathbf{A}_r^\star) + \big(\mathcal{S}_H(\mathbf{A}_r^\star) - \mathbf{A}_{r+1}^\star\big).\end{aligned} \tag{20}$$

Applying the triangle inequality and the operator bound yields

$$D_{r+1} \le \kappa_{H_r} e_r + \Delta_r(H_r), \tag{21}$$

which proves Eq. (5). Both inherited prediction error and the padded tail are included before the Transformer is evaluated.

Overlap–padding accounting. This decomposition is not needed for the bound, but makes the source of $\Delta_r(H_r)$ explicit. Let Δ_r^{ov} and Δ_r^{pad} be the separately normalized RMS discrepancies on the first $T - H_r$ and final H_r coordinates of $\mathcal{S}_{H_r}(\mathbf{A}_r^\star) - \mathbf{A}_{r+1}^\star$. Then

$$\Delta_r(H_r)^2 = \frac{T - H_r}{T}\big(\Delta_r^{\mathrm{ov}}\big)^2 + \frac{H_r}{T}\big(\Delta_r^{\mathrm{pad}}\big)^2. \tag{22}$$

This is an exact RMS identity used only for attribution; the propagation and endpoint bounds continue to operate on the complete chunk.

A.3 Shared-Noise Attenuation

Derivation. Suppressing round indices, the coupled complete-chunk interpolants satisfy

$$\mathbf{X}_\sigma - \mathbf{Y}_\sigma^\star = (1-\sigma)\bar{\mathbf{A}} + \sigma\boldsymbol{\epsilon} - (1-\sigma)\mathbf{A}^\star - \sigma\boldsymbol{\epsilon} \tag{23}$$

$$= (1-\sigma)(\bar{\mathbf{A}} - \mathbf{A}^\star). \tag{24}$$

Therefore

$$\|\mathbf{X}_\sigma - \mathbf{Y}_\sigma^\star\|_{\mathrm{RMS}} = (1-\sigma)D. \tag{25}$$

Inserting the actual full-flow intermediate state gives

$$\|\mathbf{X}_\sigma - \mathbf{X}_\sigma^\star\|_{\mathrm{RMS}} \le \|\mathbf{X}_\sigma - \mathbf{Y}_\sigma^\star\|_{\mathrm{RMS}} + \|\mathbf{Y}_\sigma^\star - \mathbf{X}_\sigma^\star\|_{\mathrm{RMS}} \tag{26}$$

$$\le (1-\sigma)D + \eta_\sigma, \tag{27}$$

which proves Eq. (8). The defect η_σ is retained because the linear training interpolant need not coincide sample-by-sample with the learned full-flow trajectory.

A.4 One-Step Corrective Stability

Derivation. Suppress round and observation indices. Let $\{\mathbf{X}_t^\star : t \in [0,\sigma]\}$ be the complete-chunk ideal remaining trajectory with $\mathbf{X}_0^\star = \mathbf{A}^\star$. Under our time convention,

$$\frac{d}{dt}\mathbf{X}_t^\star = v^\star(\mathbf{X}_t^\star, t), \qquad \mathbf{A}^\star = \mathbf{X}_\sigma^\star - \int_0^\sigma v^\star(\mathbf{X}_t^\star, t)\, dt. \tag{28}$$

Define the ideal one-step Euler endpoint

$$\mathbf{A}_{\mathrm{E}}^\star \coloneqq \mathbf{X}_\sigma^\star - \sigma v^\star(\mathbf{X}_\sigma^\star, \sigma). \tag{29}$$

Adding and subtracting $F_{\sigma,\mathbf{o}}(\mathbf{X}_\sigma^\star)$ and $\mathbf{A}_{\mathrm{E}}^\star$ gives

$$\|\widehat{\mathbf{A}} - \mathbf{A}^\star\|_{\mathrm{RMS}} \le \|F_{\sigma,\mathbf{o}}(\mathbf{X}_\sigma) - F_{\sigma,\mathbf{o}}(\mathbf{X}_\sigma^\star)\|_{\mathrm{RMS}} + \|F_{\sigma,\mathbf{o}}(\mathbf{X}_\sigma^\star) - \mathbf{A}_{\mathrm{E}}^\star\|_{\mathrm{RMS}} + \|\mathbf{A}_{\mathrm{E}}^\star - \mathbf{A}^\star\|_{\mathrm{RMS}}. \tag{30}$$

Complete-chunk Lipschitz continuity bounds the first term by $K_\sigma\|\mathbf{X}_\sigma - \mathbf{X}_\sigma^\star\|_{\mathrm{RMS}}$. The velocity error assumption gives

$$\|F_{\sigma,\mathbf{o}}(\mathbf{X}_\sigma^\star) - \mathbf{A}_{\mathrm{E}}^\star\|_{\mathrm{RMS}} \le \sigma\delta. \tag{31}$$

For the last term, define $\mathcal{D}_t v^\star = \frac{d}{dt} v^\star(\mathbf{X}_t^\star, t)$. Then

$$\|\mathbf{A}_{\mathrm{E}}^\star - \mathbf{A}^\star\|_{\mathrm{RMS}} \le \int_0^\sigma \int_t^\sigma \|\mathcal{D}_s v^\star\|_{\mathrm{RMS}}\, ds\, dt \tag{32}$$

$$\le \frac{M}{2}\sigma^2. \tag{33}$$

Substitution into Eq. (30) followed by Eq. (8) yields

$$e_{r+1} \le K_{\sigma,r+1}\big((1-\sigma)D_{r+1} + \eta_{\sigma,r+1}\big) + c_{\sigma,r+1}, \tag{34}$$

where $c_{\sigma,r+1} = \sigma\delta_{r+1} + \frac{1}{2}M_{r+1}\sigma^2$. This proves Eq. (10) on the complete chunk.

Distributional corollary. Let $\widehat{\mu}_A$ and $\mu_A^\star$ be the laws of the one-step and full-flow chunks, and let $W_{2,\mathrm{RMS}}$ use the coordinate-normalized RMS ground metric. Their construction under the same observation and base noise defines an admissible coupling, so

$$W_{2,\mathrm{RMS}}(\widehat{\mu}_A, \mu_A^\star) \le \|\widehat{\mathbf{A}} - \mathbf{A}^\star\|_{\mathrm{RMS}}, \tag{35}$$

whose right-hand side is bounded above by Eq. (10).

A.5 Sequential Replanning Stability

Finite-round recursion. Combining Eqs. (5) and (10) gives

$$\begin{aligned} e_{r+1} &\le K_{\sigma,r+1}\left[(1-\sigma)\big(\kappa_{H_r} e_r + \Delta_r(H_r)\big) + \eta_{\sigma,r+1}\right] + c_{\sigma,r+1} && (36)\\ &= \alpha_r e_r + \beta_r, && (37)\end{aligned}$$

where

$$\alpha_r \coloneqq (1-\sigma) K_{\sigma,r+1}\kappa_{H_r}, \tag{38}$$

$$\beta_r \coloneqq K_{\sigma,r+1}\big((1-\sigma)\Delta_r(H_r) + \eta_{\sigma,r+1}\big) + c_{\sigma,r+1}. \tag{39}$$

This proves Eq. (11). Iteration gives the nonuniform finite-round bound

$$e_r \le \left(\prod_{j=0}^{r-1} \alpha_j\right) e_0 + \sum_{i=0}^{r-1} \left(\prod_{j=i+1}^{r-1} \alpha_j\right) \beta_i, \tag{40}$$

where an empty product equals one. This expression allows the measured round-wise coefficients to vary.

Conditional uniform bound. If $\alpha_j \le \bar{\alpha} < 1$ and $\beta_j \le \bar{\beta}$ uniformly over rounds, then

$$\begin{aligned} e_r &\le \bar{\alpha}^r e_0 + \bar{\beta}\sum_{k=0}^{r-1} \bar{\alpha}^k && (41)\\ &= \bar{\alpha}^r e_0 + \frac{1-\bar{\alpha}^r}{1-\bar{\alpha}}\bar{\beta}. && (42)\end{aligned}$$

Taking $\limsup$ proves Eq. (12). Thus the finite-round expansion is general under the local bounds above, while the asymptotic result uses the additional uniform condition.

A.6 Sanity Checks

- If $e_r = \Delta_r(H_r) = 0$, the propagated prior mismatch is zero.
- If $\eta_{\sigma,r} = \delta_r = M_r = 0$, endpoint error is controlled by $K_{\sigma,r}(1-\sigma)D_r$.
- At $\sigma = 0$, no observation-conditioned correction is made.
- At $\sigma = 1$, temporal inheritance disappears from the input.
- For $H_r = 0$, $\kappa_{H_r} = 1$; repeating the final action for larger H_r produces the worst-case factor $\sqrt{H_r + 1}$.

The directional diagnostics and measurement protocols for $\widehat{K}_r$ and $\widehat{\alpha}_r = (1-\sigma)\kappa_{H_r}\widehat{K}_r$ are reported in Table 6.

A.7 Empirical Validation of the Error Chain

The error analysis requires temporal alignment to shorten the remaining path to the full-flow trajectory. We test this directly against a 10-NFE teacher under the same observation using full-chunk intermediate-state discrepancies,

$$\begin{aligned} I_{\mathrm{shift}} &= \|\mathbf{X}_{\sigma,r}^{\mathrm{shift}} - \mathbf{X}_{\sigma,r}^\star\|_{\mathrm{RMS}},\\ I_{\mathrm{unshift}} &= \|\mathbf{X}_{\sigma,r}^{\mathrm{unshift}} - \mathbf{X}_{\sigma,r}^\star\|_{\mathrm{RMS}},\\ I_{\mathrm{noise}} &= \|\boldsymbol{\epsilon} - \mathbf{X}_{\sigma,r}^\star\|_{\mathrm{RMS}}. \end{aligned}$$

Here I_{unshift} retains the previous chunk but ignores that the planning window has advanced, separating temporal content from temporal alignment in this diagnostic.

Quantity	Value
I_{shift} median	**0.0893**
I_{unshift} median	0.2233
I_{noise} median	0.8474
$I_{\text{shift}} < I_{\text{noise}}$	100.0%
$I_{\text{shift}} < I_{\text{unshift}}$	76.1%
$\widehat{K}$: median / P90 / P95	.236/.904/1.114
$\widehat{K} < 1$	92.4%
$\widehat{\alpha}$: median / P90 / P95	.661/2.533/3.120
$\widehat{\alpha} < 1$	63.6%
$D_{r+1} \le B_{r+1}^{\text{prior}}$	100.0%
Median $D_{r+1}/B_{r+1}^{\text{prior}}$	0.892
e_{r+1} versus e_r fit	$0.320e_r + 0.0471$
Round-to-round fit R^2	0.105

Table 6: Complete-chunk diagnostics on 1,374 warm replans from 90 fixed-$H = 18$ episodes; the round-to-round fit uses 1,284 warm-to-warm pairs. A 10-NFE teacher supplies the exact intermediate state at $\sigma = 0.357421875$. Here $\widehat{K}$ is the sampled directional endpoint gain, $\widehat{\alpha} = (1-\sigma)\kappa_H\widehat{K}$, and $B_{r+1}^{\text{prior}} = \|\mathcal{S}_H(\widehat{\mathbf{A}}_r - \mathbf{A}_r^\star)\|_{\text{RMS}} + \Delta_r(H)$. These finite-sample directional summaries are distinct from the uniform Lipschitz quantity and round-wise condition in the conditional bound.

Temporal alignment gives the smallest intermediate discrepancy: median 0.0893 versus 0.2233 for the unshifted state and 0.8474 for noise. The sampled endpoint gain satisfies $\widehat{K} < 1$ on 92.4% of replans, while the induced complete-chunk coefficient satisfies $\widehat{\alpha} < 1$ on 63.6%. The exact prior bound contains every point with median ratio 0.892. Across consecutive replans, $e_{r+1} = 0.320e_r + 0.0471$ with $R^2 = 0.105$, summarizing a non-explosive average trend amid substantial trajectory heterogeneity.

A.8 Why We Use the Last FastWAM Timestep

FastWAM's 10-NFE schedule ends with the nonzero timestep $\sigma = 0.357421875 \approx 5/14$. We inherit this value directly: under our premise, the time-aligned prior already lies near the next action distribution and should therefore traverse only the final segment of the pretrained flow. The value is fixed before downstream evaluation and is not selected by a task-level search.

To characterize sensitivity, we evaluate fixed-$H = 24$ warm-start one-step inference on RoboTwin `open_microwave` under the clean condition, using 20 closed-loop episodes per timestep. The sweep in Table 7 is broad and non-monotonic: too little perturbation leaves insufficient room for observation-conditioned correction, whereas too much progressively erases the temporal prior.

σ	Success rate
0.10	5% -70
0.35	75%
0.50	85% +10
0.75	85% +10
0.90	55% -20

Table 7: Closed-loop perturbation-timestep sensitivity on RoboTwin `open_microwave`/clean with fixed $H = 24$ and warm-start one-step inference. Each setting contains 20 episodes; small signed values are percentage-point differences from the deployed-schedule approximation $\sigma = 0.35$.

The best observed settings in this sweep are $\sigma = 0.50$ and 0.75, rather than the deployed 0.357421875 (approximated by the 0.35 row). Thus, the main configuration reflects the pretrained FastWAM schedule, not a perturbation strength optimized for this benchmark.

B Complete Experimental Protocol

Benchmarks and accepted seeds. We evaluate 100 episodes for each of 50 RoboTwin tasks under clean and randomized conditions (10,000 total), and 50 episodes for each of 40 LIBERO tasks across its four suites (2,000 total). RoboTwin filters initializations through an expert solvability check, so forcing identical raw seeds across independently evaluated policies can invalidate an episode; we use accepted-seed pairing where matching episodes are available.

Inference and timestep. We use the released pretrained FastWAM without changing its architecture, objective, or parameters. We directly adopt the last nonzero timestep of its original 10-NFE trajectory, fixed across all tasks and episodes, rather than tuning perturbation strength downstream. Its exact value and closed-loop sensitivity are reported in App. A.8.

Attention search and comparisons. We search positions 12–24 on RoboTwin and 8–10 on LIBERO, smooth diagonal attention with radius $\rho = 2$, and select its local minimum using Eq. (13). Besides standard FastWAM, *Fresh Noise + 1 NFE* removes temporal reuse, *Direct Prior Reuse* removes noise-perturbed flow re-entry, and the full method is evaluated with fixed and attention-guided horizons on both benchmarks under the same inference settings.

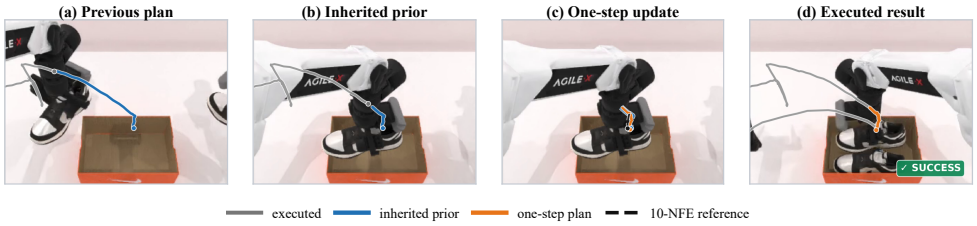

Figure 9: RoboTwin rollout across consecutive replans. After the current prefix is executed, the remaining future is aligned and corrected in one NFE while staying close to the same-seed 10-NFE reference; the corrected plan is then carried into the successful continuation.

Because a shorter horizon creates more planning calls while the bootstrap is amortized over the episode, we report per-plan NFE, realized horizon, and synchronized latency together. These expose the closed-loop cost that the nominal one-step update alone does not. The measured speedups therefore reflect the complete online planning path rather than an isolated decoder call.

C Table-by-Table Experimental Configurations

This section disambiguates evaluation campaigns that share a task subset but use different episode budgets. Unless stated otherwise, RoboTwin uses both clean and randomized conditions, the frozen FastWAM checkpoint, policy seed 42, accepted-environment seed search beginning at 4,300,000, action chunks of length 32, and $\sigma = 0.357421875$. Our warm-start variants use 10 NFEs for the first plan and one NFE thereafter. Published comparison rows are copied from their cited sources and are not treated as episode-paired with our runs.

Exact RoboTwin subsets. The 15-task subset used in Tables 2 to 4 and 6 contains `move_stapler_pad`, `blocks_ranking_size`, `place_dual_shoes`, `place_mouse_pad`, `place_a2b_right`, `place_object_basket`, `place_object_stand`, `turn_switch`, `place_fan`, `dump_bin_bigbin`, `adjust_bottle`, `hanging_mug`, `handover_block`, `open_microwave`, and `beat_block_hammer`. The latency subset used in Tables 5 and 8 is the first seven tasks in this list plus `place_fan`.

Table 1: main benchmark results. Our RoboTwin rows cover 50 tasks, two conditions, and 100 episodes per task–condition cell (10,000 episodes per method). Fixed replanning uses $H = 24$; attention searches $H \in [12, 24]$ with radius $\rho = 2$. Our LIBERO rows cover all 40 tasks across Spatial, Object, Goal, and Long, with 50 episodes per task (2,000 per method); fixed replanning uses $H = 10$ and attention searches $H \in [8, 10]$. All other rows are published reference values.

Table 2: structural ablations. The RoboTwin portion is a small-sample ablation campaign on the 15-task subset: 15 tasks $\times$ 2 conditions $\times$ 20 episodes, or 600 episodes per variant. The full method and Direct Prior use a $10 \rightarrow 1$ schedule with $H = 24$; Fresh Noise discards temporal reuse and uses one NFE at every plan. The LIBERO portion uses 40 tasks $\times$ 50 episodes, or 2,000 episodes per variant, with $H = 10$. FastWAM values are published references rather than rows from these 20-episode RoboTwin campaigns.

Table 3: fixed versus attention-guided replanning. The full RoboTwin block contains 10,000 episodes per row. The RoboTwin2.0-15 block contains 100 episodes in each of 30 task–condition cells, or 3,000 per row, for fixed $H \in \{12, 18, 24\}$ and attention $H \in [12, 24]$. These aggregate rows come from completed campaigns; accepted-seed pairing is available only for the separately reported 2,738 shared episodes between fixed $H = 18$ and attention. The attention aggregate contains 2,545/3,000 successful episodes (84.83%). The LIBERO block uses 2,000 episodes per row.

Table 4: dynamic-rule ablation. This is a separate small-sample campaign on the same 15 tasks: 20 episodes per task–condition cell and 600 episodes per rule. All rules use the same frozen checkpoint and first-plan 10-NFE bootstrap followed by one-NFE warm updates. PACE-inspired searches $H \in [10, 24]$; tuned AutoHorizon and our attention rule search $H \in [12, 24]$; the official AutoHorizon row retains its released threshold. Campaign seeds are not asserted to be episode-paired, so the table supports a matched-scale ablation comparison, not a paired significance claim.

Tables 5 and 8: latency. Both tables summarize the same balanced latency campaign: eight tasks, two conditions, five episodes per task–condition–method cell, 80 episodes per method, and 480 episodes across six methods. Runs use batch size one, bf16, and three NVIDIA A800-SXM4-80GB GPUs with one worker per GPU. All 7,041 planner calls are CUDA-synchronized. Table 5 shows four central rows; Table 8 reports all six methods and the component breakdown.

Table 6: teacher diagnostics. The diagnostic uses 90 fixed-$H = 18$ episodes from the 15-task subset and all 1,374 resulting warm replans. At each replan, a same-state 10-NFE teacher is evaluated and its exact intermediate state at $\sigma = 0.357421875$ is saved. Reported distances use normalized RMS over the complete $T = 32$ action chunk; the round-to-round fit uses the 1,284 warm-to-warm pairs. These are replan-level diagnostics, not additional success-rate episodes.

Table 7: timestep sensitivity. This ablation contains five independent settings on RoboTwin `open_microwave/clean`, each with 20 closed-loop episodes, fixed $H = 24$, and one-NFE warm updates after bootstrap (100 episodes total). The $\sigma = 0.35$ row approximates the deployed 0.357421875 schedule. Because it is a one-task, 20-episode-per-setting sensitivity study, it is not used as a main benchmark estimate.

Table 9: complete task breakdown. The Ours columns are the same attention-guided full RoboTwin campaign used in Table 1: 50 tasks, clean and randomized conditions, and 100 episodes per cell (10,000 total). FastWAM and the remaining baseline columns are reproduced from the cited FastWAM comparison table; signed differences are computed task-by-task against the corresponding published FastWAM cell.

D Latency Protocol and Full Results

We measure the complete inference path with CUDA synchronization, including observation preprocessing, model execution, temporal-prior construction, attention extraction when used, horizon selection, and action postprocessing. Environment stepping, video and artifact logging, and evaluation I/O are excluded. The balanced subset contains eight RoboTwin tasks under clean and randomized conditions, with five episodes per task–condition–method cell: 80 episodes per method and 480 episodes in total. All runs use batch size one and bf16 on three NVIDIA A800-SXM4-80GB GPUs, with exactly one worker assigned to each GPU. Every one of the 7,041 recorded planner calls passes the schedule and sample-count validation.

Method	Schedule	Bootstrap (ms)	Update (ms)	Amortized (ms)	Calls/ep	Plan (s/ep)	Plan (ms/step)	Memory (GiB)
FastWAM native	$10 \to 10$	–	590.6	590.6	11.54	6.814	25.676	24.801
Fresh Noise	$1 \to 1$	–	144.1 −446.5	144.1 −446.5	13.93 +2.39	2.007 −4.807	6.152 −19.524	24.801
Direct Prior	$10 \to 1$	542.6	144.0 −446.6	164.2 −426.4	19.79 +8.25	3.249 −3.565	6.993 −18.683	24.802 +0.001
Ours, fixed $H = 24$	$10 \to 1$	544.1	144.4 −446.2	176.5 −414.1	12.44 +0.90	2.195 −4.619	7.680 −17.996	24.797 −0.004
Ours, fixed $H = 18$	$10 \to 1$	541.0	143.8 −446.8	169.6 −421.0	15.45 +3.91	2.620 −4.194	9.677 −15.999	24.802 +0.001
Ours, attention	$10 \to 1$	564.4	157.9 −432.7	185.2 −405.4	14.88 +3.34	2.755 −4.059	10.571 −15.105	24.802 +0.001

Table 8: Full latency breakdown. Small signed values are differences from FastWAM in the units of each column; lower is better except for memory, where the differences are negligible. Bootstrap has no like-for-like FastWAM value. Bootstrap and update report the first planner call and subsequent calls separately; amortized latency pools both. Planning time per episode and per environment step additionally expose the cost of each method's realized replanning frequency.

For attention-guided inference, stratified-bootstrap 95% confidence intervals are $[183.28, 187.34]$ ms for amortized call latency, $[2.582, 2.943]$ seconds for planning time per episode, and $[10.358, 10.805]$ ms per environment step. The corresponding FastWAM intervals are $[589.10, 592.06]$ ms, $[6.115, 7.554]$ seconds, and $[25.510, 25.877]$ ms. Within an attention-guided call, temporal-prior construction, attention extraction, horizon selection, preprocessing, and postprocessing average 0.153, 10.152, 0.068, 4.974, and 0.208 ms, respectively.

E Statistical and Baseline Details

Attention implementation. We reconstruct action-to-action attention from the final denoising model call, average over all ActionDiT layers and heads, retain the normalized 32×32 action block, and normalize each row. The selector evaluates the centered five-token diagonal score for every integer horizon in the stated search interval and returns its global minimum; equal scores choose the smaller horizon. Exact attention hooks, index checks, and warm-start state management are provided in the supplementary implementation.

PACE-inspired baseline. For each arm, we compute the RMS velocity over its six joints; gripper coordinates are excluded. We apply edge-replicated Gaussian smoothing with standard deviation 2 and window length 9, then detect valleys with minimum separation 10. For each task, the acceptance threshold is the fifth percentile of positive finite valley prominences collected from 550 training demonstrations (50 clean and 500 randomized), without using evaluation success. At test time, the last accepted valley in $H \in [10, 24]$ sets the execution horizon, with $H = 24$ as fallback. The first accepted valley after H sets the end of the inherited segment; if none exists, the segment ends at action 32.

Adapted AutoHorizon baseline. The tuned variant uses the same final-call ActionDiT attention and $10 \to 1$ warm-start schedule as our method. Its bidirectional soft pointer is restricted to $H \in [12, 24]$, with `hold_thr`$= 0.5$, `max_entropy_q`$= 0.9$, and `run_len`$= 1$. An unreliable plateau, an out-of-range pointer, or full bidirectional coverage falls back to $H = 24$. The separately reported official variant retains the released threshold `hold_thr`$= 0.3$.

Uncertainty and error summaries. Latency intervals use 4,000 episode-bootstrap replicates with random seed 4,300,000. Episodes are resampled within each task–condition stratum, and the 2.5th and 97.5th percentiles form the reported 95% interval. The round-to-round diagnostic in Figure 7 contains 1,284 matched consecutive-replan pairs and reports the fitted affine relation together with its intercept. For Figure 8, normalized episode progress is divided into ten equal-width bins. The plot contains 869 replan observations from successful episodes and 505 from failed episodes; each point is the bin mean and each error bar is the sample standard error $s/\sqrt{n}$.

F Claim Scope

Taken together, the results establish a three-part claim about the frozen WAM: temporal alignment supplies a closer source state, flow re-entry usually reduces its endpoint discrepancy under the latest observation, and repeated correction shows a non-explosive average trend over the mea-

sured trajectories. Structural ablations and matched-teacher diagnostics isolate these components rather than attributing the gain to one-step inference alone.

The evidence does not establish a universal temporal boundary. In particular, the paired confidence interval includes zero; the attention result is therefore narrower: online selection recovers the strong earlier-replanning regime and reduces the realized horizon without a detected task-level loss. Whether the same signal transfers across WAM backbones or physical robots remains open.

G Cross-Horizon and Cross-Task Attention Patterns

This appendix provides additional measurements underlying Figures 3 and 11. Across replanning rounds, the effective execution horizon changes and the diagonal profiles often retain elevated endpoint anchors together with state-dependent interior structure. The prevalence and location of the two-valley structure vary across tasks, especially on LIBERO-Long. This variation motivates online boundary selection instead of a fixed global attention template.

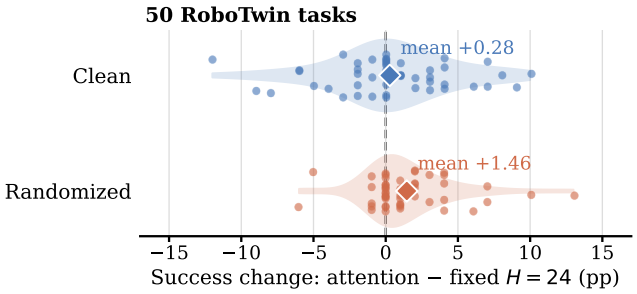

Figure 10: Per-task success change of attention replanning over fixed $H = 24$; diamonds show means.

For the matched fixed-$H = 18$ comparison, 2,738 episodes share identical accepted environment seeds with attention-guided inference. Attention improves success by 0.88 points on this subset; a task-cluster bootstrap gives a 95% confidence interval of $[-1.41, 2.40]$ points, indicating comparable task-level performance while selecting the execution horizon online.

Among warm-start replans, an exact two-valley W is detected in 3,888/8,249 RoboTwin replans (47.1%) and 950/3,640 LIBERO-Long replans (26.1%). RoboTwin extrema are comparatively concentrated (8.83 ± 1.29, 13.86 ± 1.27, and 21.10 ± 2.17 for the left valley, middle peak, and right valley), whereas LIBERO-Long exhibits broader variation. Thus, the W is a recurring, state-dependent cue. The horizon-conditioned curves group online replans by their selected H and characterize the resulting attention structure across both benchmarks. For this descriptive statistic, we smooth each row-normalized attention diagonal with a Gaussian kernel of standard deviation 1.2 and require exactly two valleys at action positions 4–29, each with prominence at least 10% of the within-profile range and separated by at least four positions; the intervening maximum defines the middle peak.

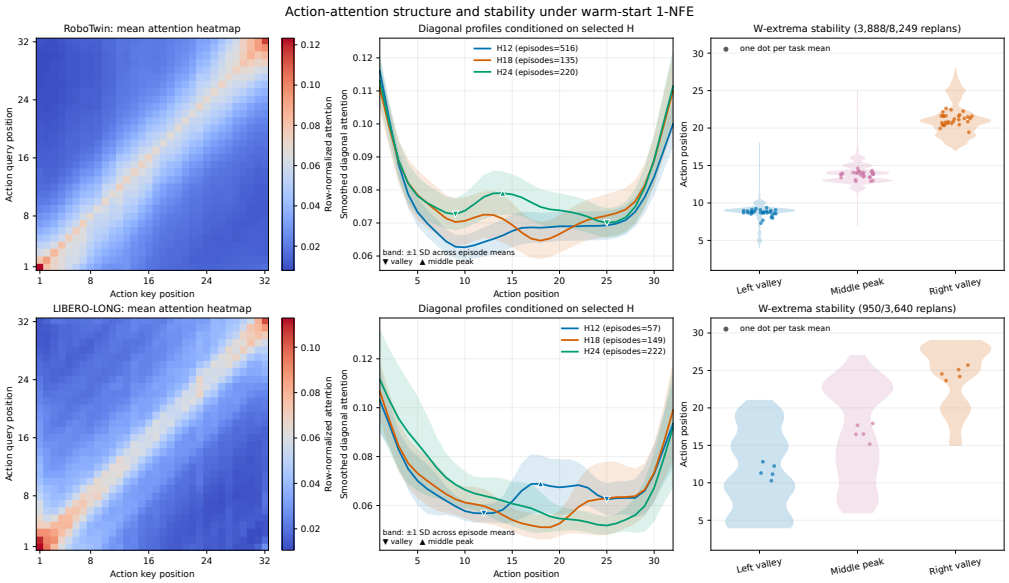

Figure 11: Attention stability across RoboTwin (top) and LIBERO-Long (bottom). From left to right: episode-balanced mean action-attention, diagonal profiles conditioned on the online selected horizon (bands show one standard deviation across episode means), and task-level locations of detected W extrema. Only warm-start 1-NFE replans are included; bootstrap plans are excluded.

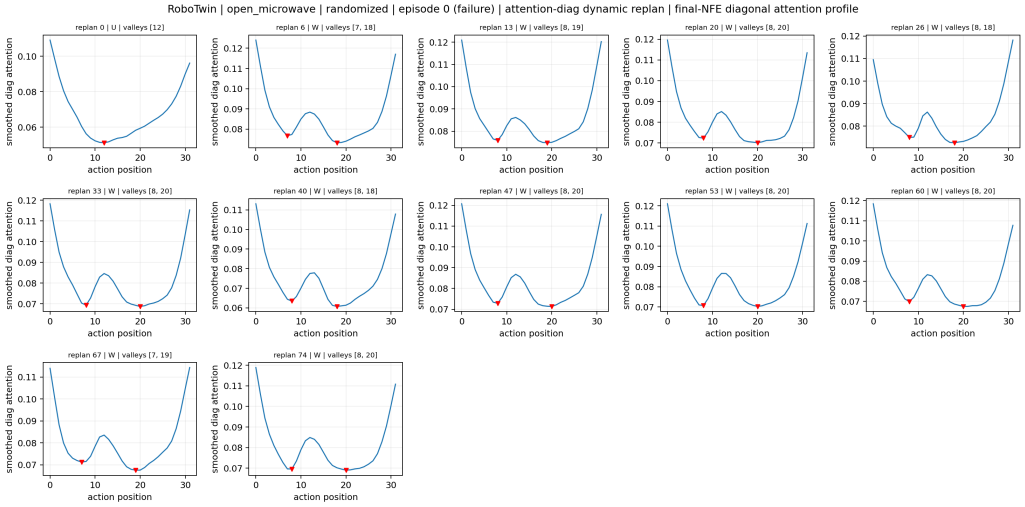

Figure 12: Diagonal action-attention profiles across replanning rounds on RoboTwin `open_microwave`. The effective horizon varies across the episode, while several rounds exhibit the W-like high–low–high–low–high profile. Red markers identify its two valleys.

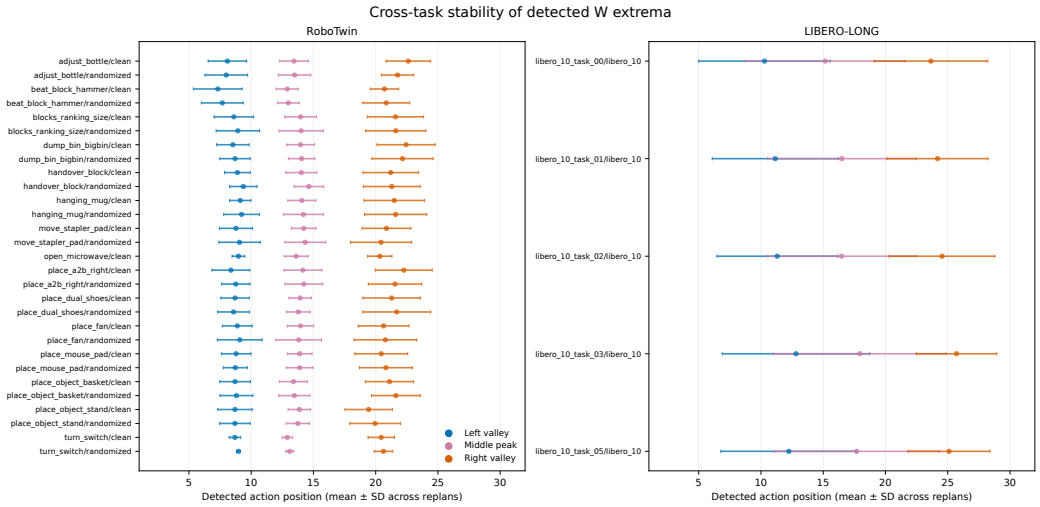

Figure 14: Cross-task stability of detected W extrema. Points and error bars show the mean and standard deviation across warm-start replans within each task–condition cell. RoboTwin exhibits more concentrated extrema, whereas LIBERO-Long shows substantially greater task-dependent variation.

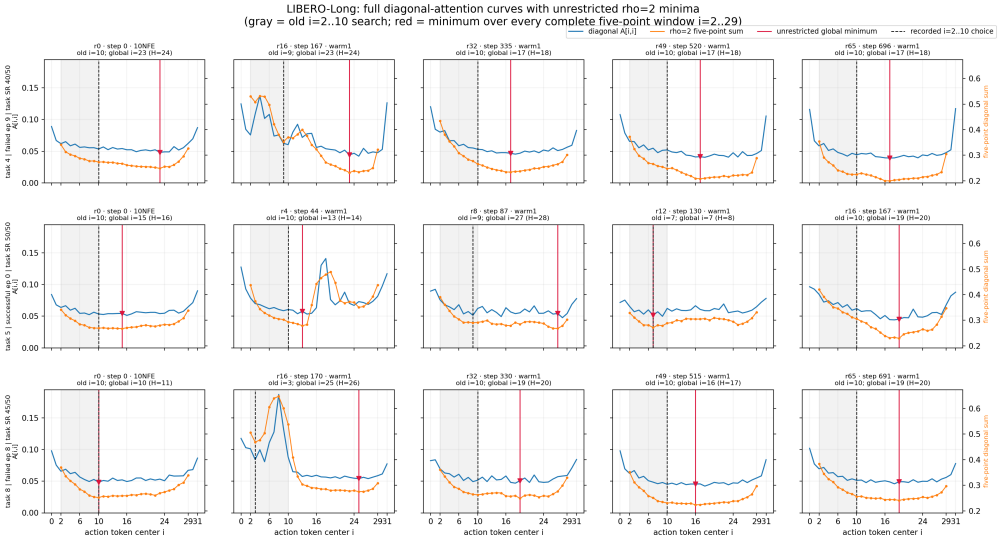

Figure 13: Additional full-range diagonal-attention traces from distinct LIBERO-Long tasks and replanning horizons. Endpoint elevation recurs across tasks; state-dependent interior structure determines whether and where a second valley emerges. This variation motivates restricted, online search rather than a globally fixed attention threshold shared across tasks.

H COMPLETE PER-TASK ROBOTWIN COMPARISON

Table 9 reports all 50 RoboTwin tasks under clean (C) and randomized (R) evaluation. Published comparison results are taken from FastWAM (Yuan et al., 2026); our attention-guided results contain 100 episodes per task and condition. The resulting averages are 91.48% under clean conditions, 91.06% under randomization, and 91.27% overall.

	Ours		FastWAM		Joint		IDM		No co-train		LingBot-VA		$\pi_{0.5}$		Motus	
Task	C	R	C	R	C	R	C	R	C	R	C	R	C	R	C	R
Adjust Bottle	100$_{0}$	100$_{0}$	100	100	98$_{-2}$	99$_{-1}$	94$_{-6}$	99$_{-1}$	98$_{-2}$	100$_{0}$	90$_{-10}$	94$_{-6}$	100$_{0}$	99$_{-1}$	89$_{-11}$	93$_{-7}$
Beat Block Hammer	100$_{+1}$	98$_{+1}$	99	97	100$_{+1}$	98$_{+1}$	98$_{-1}$	98$_{+1}$	80$_{-19}$	92$_{-5}$	96$_{-3}$	98$_{+1}$	96$_{-3}$	93$_{-4}$	95$_{-4}$	88$_{-9}$
Blocks Ranking RGB	100$_{0}$	97$_{-3}$	100	100	100$_{0}$	100$_{0}$	100$_{0}$	99$_{-1}$	88$_{-12}$	86$_{-14}$	99$_{-1}$	98$_{-2}$	92$_{-8}$	85$_{-15}$	99$_{-1}$	97$_{-3}$
Blocks Ranking Size	97$_{+3}$	89$_{-9}$	94	98	83$_{-11}$	91$_{-7}$	79$_{-15}$	90$_{-8}$	56$_{-38}$	62$_{-36}$	94$_{0}$	96$_{-2}$	49$_{-45}$	26$_{-72}$	75$_{-19}$	63$_{-35}$
Click Alarmclock	100$_{0}$	100$_{0}$	100	100	100$_{0}$	100$_{0}$	98$_{-2}$	100$_{0}$	100$_{0}$	98$_{-2}$	99$_{-1}$	100$_{0}$	98$_{-2}$	89$_{-11}$	100$_{0}$	100$_{0}$
Click Bell	100$_{0}$	99$_{-1}$	100	100	100$_{0}$	98$_{-2}$	100$_{0}$	96$_{-4}$	100$_{0}$	100$_{0}$	100$_{0}$	100$_{0}$	99$_{-1}$	66$_{-34}$	100$_{0}$	100$_{0}$
Dump Bin Bigbin	95$_{-2}$	93$_{-3}$	97	96	95$_{-2}$	95$_{-1}$	93$_{-4}$	98$_{+2}$	92$_{-5}$	94$_{-2}$	89$_{-8}$	96$_{0}$	92$_{-5}$	97$_{+1}$	95$_{-2}$	91$_{-5}$
Grab Roller	100$_{0}$	100$_{0}$	100	100	100$_{0}$	100$_{0}$	100$_{0}$	100$_{0}$	100$_{0}$	100$_{0}$	100$_{0}$	100$_{0}$	100$_{0}$	100$_{0}$	100$_{0}$	100$_{0}$
Handover Block	94$_{-1}$	82$_{-1}$	95	81	93$_{-2}$	91$_{+10}$	97$_{+2}$	94$_{+13}$	58$_{-37}$	46$_{-35}$	99$_{+4}$	78$_{-3}$	66$_{-29}$	57$_{-24}$	86$_{-9}$	73$_{-8}$
Handover Mic	99$_{0}$	97$_{-3}$	99	100	100$_{+1}$	100$_{0}$	98$_{-1}$	99$_{-1}$	100$_{+1}$	100$_{0}$	94$_{-5}$	96$_{-4}$	98$_{-1}$	97$_{-3}$	78$_{-21}$	63$_{-37}$
Hanging Mug	69$_{+11}$	74$_{+12}$	58	62	71$_{+13}$	56$_{-6}$	66$_{+8}$	62$_{0}$	28$_{-30}$	40$_{-22}$	40$_{-18}$	28$_{-34}$	18$_{-40}$	17$_{-45}$	38$_{-20}$	38$_{-24}$
Lift Pot	98$_{-2}$	100$_{0}$	100	100	100$_{0}$	100$_{0}$	100$_{0}$	100$_{0}$	92$_{-8}$	90$_{-10}$	100$_{0}$	99$_{-1}$	96$_{-4}$	85$_{-15}$	96$_{-4}$	99$_{-1}$
Move Can Pot	94$_{+4}$	93$_{+5}$	90	88	97$_{+7}$	99$_{+11}$	97$_{+7}$	100$_{+12}$	80$_{-10}$	68$_{-20}$	94$_{+4}$	97$_{+9}$	51$_{-39}$	55$_{-33}$	34$_{-56}$	74$_{-14}$
Move Pillbottle Pad	100$_{0}$	99$_{0}$	100	99	99$_{-1}$	100$_{+1}$	98$_{-2}$	100$_{+1}$	88$_{-12}$	96$_{-3}$	99$_{-1}$	99$_{0}$	84$_{-16}$	61$_{-38}$	93$_{-7}$	96$_{-3}$
Move Playingcard Away	100$_{0}$	100$_{0}$	100	100	100$_{0}$	100$_{0}$	99$_{-1}$	100$_{0}$	94$_{-6}$	96$_{-4}$	100$_{0}$	99$_{-1}$	96$_{-4}$	84$_{-16}$	100$_{0}$	96$_{-4}$
Move Stapler Pad	68$_{-9}$	63$_{-1}$	77	64	85$_{+8}$	81$_{+17}$	89$_{+12}$	85$_{+21}$	64$_{-13}$	78$_{+14}$	91$_{+14}$	79$_{+15}$	56$_{-21}$	42$_{-22}$	83$_{+6}$	85$_{+21}$
Open Laptop	96$_{-2}$	100$_{0}$	98	100	89$_{-9}$	92$_{-8}$	92$_{-6}$	92$_{-8}$	100$_{+2}$	98$_{-2}$	92$_{-6}$	94$_{-6}$	90$_{-8}$	96$_{-4}$	95$_{-3}$	91$_{-9}$
Open Microwave	70$_{+8}$	**81**$_{+6}$	62	45	3$_{-59}$	14$_{-31}$	54$_{-8}$	53$_{+8}$	46$_{-16}$	52$_{+7}$	82$_{+20}$	86$_{+41}$	34$_{-28}$	77$_{+32}$	95$_{+33}$	91$_{+46}$
Pick Diverse Bottles	77$_{-3}$	86$_{+1}$	80	85	86$_{+6}$	87$_{+2}$	87$_{+7}$	89$_{+4}$	58$_{-22}$	62$_{-23}$	89$_{+9}$	82$_{-3}$	81$_{+1}$	71$_{-14}$	90$_{+10}$	91$_{+6}$
Pick Dual Bottles	99$_{-1}$	96$_{0}$	100	96	98$_{-2}$	99$_{+3}$	100$_{0}$	98$_{+2}$	80$_{-20}$	74$_{-22}$	100$_{0}$	99$_{+3}$	93$_{-7}$	63$_{-33}$	96$_{-4}$	90$_{-6}$
Place A2B Left	97$_{+2}$	96$_{+3}$	95	93	96$_{+1}$	96$_{+3}$	97$_{+2}$	96$_{+3}$	84$_{-11}$	92$_{-1}$	97$_{+2}$	93$_{0}$	87$_{-8}$	82$_{-11}$	88$_{-7}$	79$_{-14}$
Place A2B Right	99$_{+6}$	95$_{-4}$	93	99	95$_{+2}$	95$_{-4}$	94$_{+1}$	98$_{-1}$	88$_{-5}$	84$_{-15}$	97$_{+4}$	95$_{-4}$	87$_{-6}$	84$_{-15}$	91$_{-2}$	87$_{-12}$
Place Bread Basket	87$_{-4}$	97$_{+4}$	91	93	89$_{-2}$	94$_{+1}$	91$_{0}$	97$_{+4}$	74$_{-17}$	76$_{-17}$	97$_{+6}$	95$_{+2}$	77$_{-14}$	64$_{-29}$	91$_{0}$	94$_{+1}$
Place Bread Skillet	91$_{+1}$	91$_{-2}$	90	93	90$_{0}$	93$_{0}$	90$_{0}$	95$_{+2}$	98$_{+8}$	84$_{-9}$	95$_{+5}$	90$_{-3}$	85$_{-5}$	66$_{-27}$	86$_{-4}$	83$_{-10}$
Place Burger Fries	93$_{-3}$	99$_{0}$	96	99	100$_{+4}$	100$_{+1}$	97$_{+1}$	99$_{0}$	94$_{-2}$	96$_{-3}$	97$_{+1}$	95$_{-4}$	94$_{-2}$	87$_{-12}$	98$_{+2}$	98$_{-1}$
Place Can Basket	68$_{-3}$	54$_{-15}$	71	69	50$_{-21}$	23$_{-46}$	37$_{-34}$	28$_{-41}$	72$_{+1}$	72$_{+3}$	81$_{+10}$	84$_{+15}$	62$_{-9}$	62$_{-7}$	81$_{+10}$	76$_{+7}$
Place Cans Plasticbox	99$_{0}$	94$_{-2}$	99	96	98$_{-1}$	98$_{+2}$	98$_{-1}$	96$_{0}$	98$_{-1}$	96$_{0}$	100$_{+1}$	99$_{+3}$	94$_{-5}$	84$_{-12}$	98$_{-1}$	94$_{-2}$
Place Container Plate	99$_{+3}$	99$_{-1}$	96	100	99$_{+3}$	98$_{-2}$	100$_{+4}$	96$_{-4}$	94$_{-2}$	98$_{-2}$	99$_{+3}$	97$_{-3}$	99$_{+3}$	95$_{-5}$	98$_{+2}$	99$_{-1}$
Place Dual Shoes	89$_{-5}$	88$_{0}$	94	88	93$_{-1}$	89$_{+1}$	85$_{-9}$	87$_{-1}$	80$_{-14}$	74$_{-14}$	94$_{0}$	89$_{+1}$	75$_{-19}$	75$_{-13}$	93$_{-1}$	87$_{-1}$
Place Empty Cup	100$_{0}$	100$_{0}$	100	100	100$_{0}$	100$_{0}$	100$_{0}$	100$_{0}$	100$_{0}$	100$_{0}$	100$_{0}$	100$_{0}$	100$_{0}$	99$_{-1}$	99$_{-1}$	98$_{-2}$
Place Fan	96$_{0}$	93$_{-3}$	96	96	99$_{+3}$	96$_{0}$	97$_{+1}$	95$_{-1}$	80$_{-16}$	88$_{-8}$	99$_{+3}$	93$_{-3}$	87$_{-9}$	85$_{-11}$	91$_{-5}$	87$_{-9}$
Place Mouse Pad	88$_{+5}$	84$_{-5}$	83	89	96$_{+13}$	91$_{+2}$	97$_{+14}$	93$_{+4}$	64$_{-19}$	76$_{-13}$	93$_{+10}$	96$_{+7}$	60$_{-23}$	39$_{-50}$	66$_{-17}$	68$_{-21}$
Place Object Basket	81$_{-8}$	80$_{-8}$	89	88	86$_{-3}$	81$_{-7}$	87$_{-2}$	82$_{-6}$	82$_{-7}$	90$_{+2}$	91$_{+2}$	88$_{0}$	80$_{-9}$	76$_{-12}$	81$_{-8}$	87$_{-1}$
Place Object Scale	96$_{+6}$	94$_{-3}$	90	97	96$_{+6}$	99$_{+2}$	99$_{+9}$	99$_{+2}$	86$_{-4}$	80$_{-17}$	96$_{+6}$	95$_{-2}$	86$_{-4}$	80$_{-17}$	88$_{-2}$	85$_{-12}$
Place Object Stand	95$_{+5}$	93$_{-1}$	90	94	92$_{+2}$	98$_{+4}$	96$_{+6}$	100$_{+6}$	82$_{-8}$	92$_{-2}$	99$_{+9}$	96$_{+2}$	91$_{+1}$	85$_{-9}$	98$_{+8}$	97$_{+3}$
Place Phone Stand	95$_{-2}$	99$_{0}$	97	99	100$_{+3}$	100$_{+1}$	99$_{+2}$	99$_{0}$	90$_{-7}$	92$_{-7}$	97$_{0}$	97$_{-2}$	81$_{-16}$	81$_{-18}$	87$_{-10}$	86$_{-13}$
Place Shoe	95$_{-1}$	98$_{-1}$	96	99	95$_{-1}$	97$_{-2}$	95$_{-1}$	98$_{-1}$	92$_{-4}$	90$_{-9}$	98$_{+2}$	98$_{-1}$	92$_{-4}$	93$_{-6}$	99$_{+3}$	97$_{-2}$
Press Stapler	94$_{+4}$	96$_{-1}$	90	97	52$_{-38}$	50$_{-47}$	50$_{-40}$	57$_{-40}$	80$_{-10}$	80$_{-17}$	85$_{-5}$	82$_{-15}$	87$_{-3}$	83$_{-14}$	93$_{+3}$	98$_{+1}$
Put Bottles Dustbin	91$_{-4}$	91$_{+1}$	95	90	93$_{-2}$	95$_{+5}$	97$_{+2}$	92$_{+2}$	78$_{-17}$	88$_{-2}$	87$_{-8}$	91$_{+1}$	84$_{-11}$	79$_{-11}$	81$_{-14}$	79$_{-11}$
Put Object Cabinet	84$_{-10}$	87$_{-2}$	94	89	95$_{+1}$	90$_{+1}$	93$_{-1}$	90$_{+1}$	88$_{-6}$	84$_{-5}$	85$_{-9}$	87$_{-2}$	80$_{-14}$	79$_{-10}$	88$_{-6}$	71$_{-18}$
Rotate QRcode	86$_{-7}$	88$_{-1}$	93	89	91$_{-2}$	92$_{+3}$	91$_{-2}$	86$_{-3}$	82$_{-11}$	78$_{-11}$	96$_{+3}$	91$_{+2}$	89$_{-4}$	87$_{-2}$	89$_{-4}$	73$_{-16}$
Scan Object	90$_{+1}$	91$_{-1}$	89	92	92$_{+3}$	92$_{0}$	93$_{+4}$	90$_{-2}$	78$_{-11}$	86$_{-6}$	96$_{+7}$	91$_{-1}$	72$_{-17}$	65$_{-27}$	67$_{-22}$	66$_{-26}$
Shake Bottle	100$_{0}$	100$_{0}$	100	100	100$_{0}$	100$_{0}$	100$_{0}$	100$_{0}$	100$_{0}$	100$_{0}$	100$_{0}$	97$_{-3}$	99$_{-1}$	97$_{-3}$	100$_{0}$	97$_{-3}$
Shake Bottle Horizontally	100$_{0}$	100$_{0}$	100	100	100$_{0}$	100$_{0}$	100$_{0}$	100$_{0}$	100$_{0}$	100$_{0}$	100$_{0}$	99$_{-1}$	99$_{-1}$	99$_{-1}$	100$_{0}$	98$_{-2}$
Stack Blocks Three	96$_{+1}$	95$_{-2}$	95	97	98$_{+3}$	97$_{0}$	99$_{+4}$	95$_{-2}$	90$_{-5}$	90$_{-7}$	99$_{+4}$	98$_{+1}$	91$_{-4}$	76$_{-21}$	91$_{-4}$	95$_{-2}$
Stack Blocks Two	100$_{0}$	100$_{0}$	100	100	100$_{0}$	100$_{0}$	100$_{0}$	100$_{0}$	100$_{0}$	98$_{-2}$	100$_{0}$	98$_{-2}$	97$_{-3}$	100$_{0}$	100$_{0}$	98$_{-2}$
Stack Bowls Three	72$_{-8}$	82$_{+1}$	80	81	84$_{+4}$	86$_{+5}$	85$_{+5}$	83$_{+2}$	66$_{-14}$	82$_{+1}$	86$_{+6}$	83$_{+2}$	77$_{-3}$	71$_{-10}$	79$_{-1}$	87$_{+6}$
Stack Bowls Two	92$_{0}$	98$_{0}$	92	98	97$_{+5}$	95$_{-3}$	94$_{+2}$	96$_{-2}$	90$_{-2}$	98$_{0}$	94$_{+2}$	98$_{0}$	95$_{+3}$	96$_{-2}$	98$_{+6}$	98$_{0}$
Stamp Seal	88$_{-2}$	90$_{-4}$	90	94	96$_{+6}$	99$_{+5}$	99$_{+9}$	94$_{0}$	60$_{-30}$	78$_{-16}$	96$_{+6}$	97$_{+3}$	79$_{-11}$	55$_{-39}$	93$_{+3}$	92$_{-2}$
Turn Switch	57$_{-4}$	64$_{+5}$	61	59	73$_{+12}$	72$_{+13}$	59$_{-2}$	74$_{+15}$	66$_{+5}$	66$_{+7}$	44$_{-17}$	45$_{-14}$	62$_{+1}$	54$_{-5}$	84$_{+23}$	78$_{+19}$
Average	**91.48**$_{-0.40}$	**91.06**$_{-0.72}$	91.88	91.78	90.84$_{-1.04}$	90.32$_{-1.46}$	91.16$_{-0.72}$	91.34$_{-0.44}$	82.76$_{-9.12}$	84.80$_{-6.98}$	92.90$_{+1.02}$	91.50$_{-0.28}$	82.74$_{-9.14}$	76.76$_{-15.02}$	88.66$_{-3.22}$	87.02$_{-4.76}$

Table 9: Complete per-task RoboTwin comparison. All entries are success rates (%). Small signed values are percentage-point differences from FastWAM in the same task and condition (blue: better; red: worse; gray: tied). C and R denote clean and randomized evaluation. FastWAM variants and published baselines are reproduced from Table 3 of FastWAM (Yuan et al., 2026).