DISPECWAM: DISTILLATION AND SPECULATIVE INFERENCE FOR WORLD ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Current World Action Models (WAMs) face a tension between future prediction and inference latency. Multi-step video–action generation is expensive, while aggressive step reduction can change both predicted futures and actions. We propose **DiSpecWAM** to retain explicit future prediction and correct unreliable few-step actions. F2 Distill trains low- and higher-budget Student Models using Teacher-transition consistency and video distribution calibration on Student rollouts. At inference, R2 Spec uses the Students as Draft and Target Models. The Target verifies Draft proposals, locally repairs selected rejections, and generates a complete action chunk when recovery fails. On RoboTwin 2.0, DiSpecWAM achieves 92.65% average success with 205 ms mean latency per inference call, an arithmetic 16.61× latency ratio to the evaluated full-budget LingBot-VA configuration.

1 INTRODUCTION

Embodied intelligence aims to endow robots with human-like abilities to perceive, reason, and interact with the physical world. Within this field, World Action Models (WAMs), including PAD (Guo et al., 2024) and UWM (Zhu et al., 2025), have emerged as a predictive control paradigm that couples future visual prediction with action generation. Unlike policies that map current observations directly to actions, WAMs jointly model visual dynamics and action trajectories, enabling robots to anticipate action consequences and learn future-aware representations of physical interactions. This aids long-horizon control.

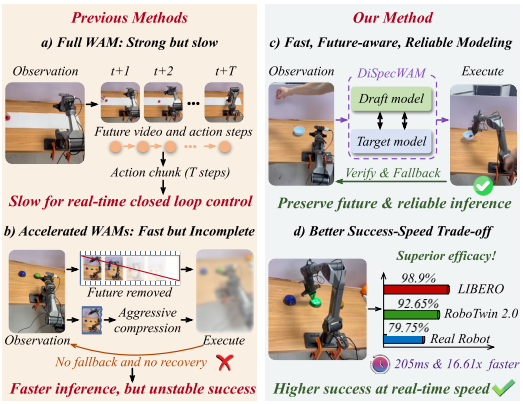


Figure 1: DiSpecWAM retains future prediction while combining few-step generation with action verification and recovery. (a) Full-budget WAM generation. (b) Faster WAM designs that omit or compress explicit future prediction. (c) Draft–Target control. (d) Success on LIBERO, RoboTwin 2.0, and four real-robot tasks. The 205 ms value is the RoboTwin 2.0 inference-call latency. The 16.61× annotation is its arithmetic ratio to the displayed full-budget LingBot-VA latency, without hardware normalization.

However, iterative video and action denoising can delay responses to new observations. For our synchronous full-budget evaluation of LingBot-VA (Li et al., 2026b), we use 25 video and 50 action denoising steps per 32-action proposal. We measure 3.405 seconds of model-only latency on a single NVIDIA RTX 4090 D GPU. This configuration limits responsive closed-loop replanning. Other schedules and asynchronous deployment can change latency.

To mitigate this bottleneck, existing approaches can be broadly grouped into two categories. The first avoids explicit future-video generation at inference. Methods such as UVA (Li et al., 2025), Fast-WAM (Yuan et al., 2026), and GigaWorld-Policy (Ye et al., 2026a) incorporate video modeling as a training objective but perform control without generating future frames at test time. The second retains future prediction while reducing its computational cost. Flash-WAM (Akbari et al., 2026) distills video and action diffusion to one and two denoising steps, respectively, whereas Efficient-

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.

WAM (Li et al., 2026a) represents future dynamics using sparse, control-relevant representations rather than high-fidelity videos.

These approaches leave open how to retain explicit future-video generation at low latency while correcting unreliable few-step actions online. Omitting generated future frames does not imply losing learned dynamics or closed-loop control. Conversely, a fixed-budget video–action generator can still produce action errors under strong compression. We study a future-predictive WAM whose low-budget proposals are checked by a higher-budget Student, with local repair and complete Target generation when needed.

We propose DiSpecWAM, which couples few-step distillation with adaptive speculative inference (Leviathan et al., 2023). F2 Distill trains low- and high-budget Students using video–action consistency and video distribution calibration. At inference, R2 Spec assigns them as Draft and Target. Draft proposes actions, while Target verifies, repairs, and supplies fallback actions.

Our contributions are:

- **We introduce DiSpecWAM,** an end-to-end WAM acceleration framework combining few-step video–action distillation with adaptive speculative inference. It preserves future prediction and offers flexible speed–performance trade-offs.
- **We propose F2 Distill,** which builds on modality-specific consistency distillation and adds rollout-based video distribution calibration for two Student budgets.
- **We develop R2 Spec,** which extends Draft–Target verification with local action repair, independent reverification, and complete Target generation after failed recovery.

2 Related Work

World Action Models. World Action Models jointly model future dynamics and robot actions. PAD (Guo et al., 2024) and UWM (Zhu et al., 2025) couple video and action diffusion within shared Transformer backbones. WorldVLA (Cen et al., 2025) autoregressively unifies image and action tokens. UniVLA (Wang et al., 2026) models vision, language, and action as discrete token sequences. DreamVLA (Zhang et al., 2026b) predicts compact dynamic, spatial, and semantic future cues for action generation, and DreamZero (Ye et al., 2026b) adapts pretrained video diffusion to robot control. LingBot-VA (Li et al., 2026b) preserves temporal causality through interleaved video–action generation, whereas Motus (Bi et al., 2026) coordinates perception, future prediction, and control with a Mixture-of-Transformers architecture. These models provide predictive control but remain computationally demanding.

World Action Model Acceleration. Existing approaches either omit explicit future generation or compress predictive modeling. UVA (Li et al., 2025), Fast-WAM (Yuan et al., 2026), and GigaWorld-Policy (Ye et al., 2026a) use future modeling during training but bypass video generation at inference. Flash-WAM (Akbari et al., 2026) introduced modality-specific consistency parameterizations for few-step video and action generation. Light-WAM (Li et al., 2026c) combines a compact video backbone with direct action decoding. LaWAM (Chen et al., 2026) predicts latent visual subgoals, while Efficient-WAM (Li et al., 2026a) uses sparse control-oriented future representations. Realtime-VLA FLASH (Niu et al., 2026) already verifies speculative action proposals and falls back to full inference. Our setting differs in retaining explicit few-step WAM video prediction and studying local action repair followed by an independent recheck.

3 Method

3.1 Overview

As illustrated in Figure 2, DiSpecWAM combines F2 Distill with R2 Spec. During training, F2 Distill transfers the full-capacity Teacher Model into two few-step Student Models through consistency and distribution distillation, without assigning control roles. At inference, R2 Spec assigns the lower-budget and higher-budget Students as Draft and Target, respectively. Draft generates candidate video–action trajectories, and Target verifies, repairs, or replaces their action components.

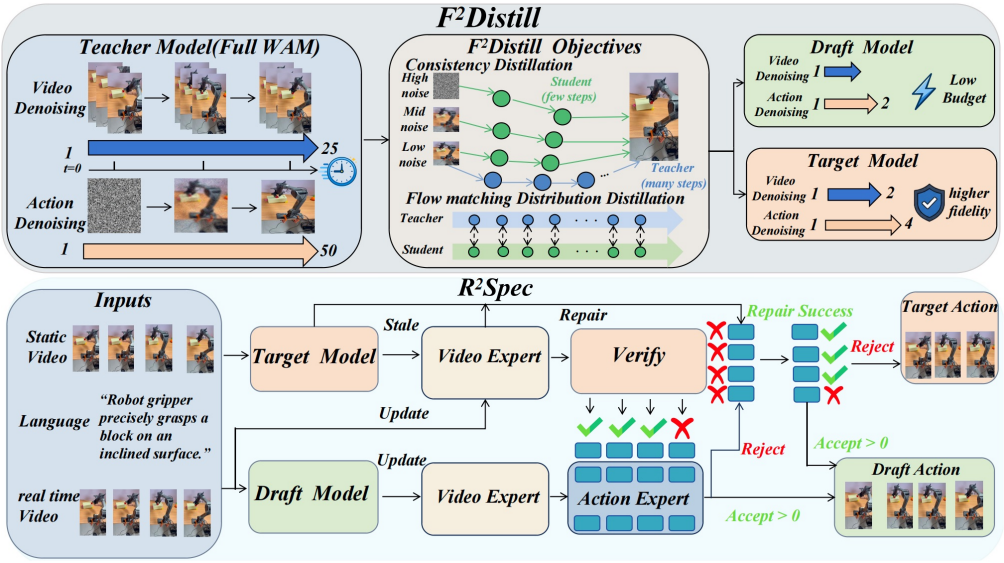


Figure 2: DiSpecWAM pipeline. F2 Distill combines Teacher-transition consistency with Student-rollout distribution calibration to train lower- and higher-budget Students. R2 Spec assigns them as Draft and Target. Target video processing is shown schematically. Initial verification uses cached Target features, while repair and fallback compute a fresh Target context. Only zero-prefix rejections with agreeing discrete decisions enter repair. Accepted repairs execute a corrected prefix. Failed recovery invokes complete higher-budget Target action generation.

3.2 Few-Step Fidelity Distillation

F2 Distill trains two denoising budgets from a full Teacher. We adopt modality-specific consistency parameterizations from Flash-WAM (Akbari et al., 2026) and consistency models (Song & Dhariwal, 2024). We adapt Distribution Matching Distillation (DMD) (Yin et al., 2024) to velocity-implied video endpoints, calling this adaptation Flow Matching Distribution Distillation (FMDD). Both objectives train each Student.

Few-Step Consistency Distillation. Let $\mu \in \mathcal{U} = \{v, a\}$ index video and action, and let x_0^μ and $\epsilon^\mu \sim \mathcal{N}(0, I)$ denote a clean sample and Gaussian noise. The condition c includes observations and the other modality's current noisy state and noise level. Within each paired Teacher transition and its consistency target, that other-modality state is held fixed, so both sides of Eq. (4) use the same c. The flow-matching path at $\sigma \in [0, 1]$ is

$$x_\sigma^\mu = (1 - \sigma)x_0^\mu + \sigma\epsilon^\mu. \tag{1}$$

Along this path, the velocity convention is $\partial_\sigma x_\sigma^\mu = \epsilon^\mu - x_0^\mu$. For adjacent levels $\sigma_s > \sigma_e$, the Teacher Model provides a one-step Euler transition

$$\widetilde{x}_{\sigma_e}^\mu = x_{\sigma_s}^\mu + (\sigma_e - \sigma_s)v_F^\mu(x_{\sigma_s}^\mu, \sigma_s, c). \tag{2}$$

Shorter Teacher transitions limit displacement from velocity deviations along the sampled path, as detailed in Appendix A.

The Student first predicts a clean endpoint and then forms a consistency output:

$$\widehat{x}_{0,\theta}^\mu = x_\sigma^\mu - \sigma v_\theta^\mu(x_\sigma^\mu, \sigma, c), \qquad f_\theta^\mu = c_{\text{skip}}^\mu(\sigma)x_\sigma^\mu + c_{\text{out}}^\mu(\sigma)\widehat{x}_{0,\theta}^\mu. \tag{3}$$

For video, $c_{\text{skip}}^v(\sigma) = \sigma_{\text{data}}^2/(\sigma^2 + \sigma_{\text{data}}^2)$ and $c_{\text{out}}^v(\sigma) = \sigma\sigma_{\text{data}}/\sqrt{\sigma^2 + \sigma_{\text{data}}^2}$, with fixed $\sigma_{\text{data}} > 0$. For action, $c_{\text{skip}}^a = 0$ and $c_{\text{out}}^a = 1$, so $f_\theta^a = \widehat{x}_{0,\theta}^a$. Both parameterizations satisfy the boundary identity $f_\theta^\mu(x_0^\mu, 0, c) = x_0^\mu$ exactly. Thus the loss need only enforce agreement away from zero noise. Let $\langle u \rangle_M = \sum_i M_i u_i / \sum_i M_i$ for a nonempty binary mask M. With positive modality weights λ_μ, masks M_μ, stop-gradient sg, and an elementwise penalty $\rho_{\delta_{\text{CD}}}$, define

$$L_{\text{CD}} = \sum_{\mu \in \mathcal{U}} \lambda_\mu \mathbb{E}\left[\left\langle \rho_{\delta_{\text{CD}}}\left(f_\theta^\mu(x_{\sigma_s}^\mu, \sigma_s, c) - \text{sg}[f_{\bar{\theta}}^\mu(\widetilde{x}_{\sigma_e}^\mu, \sigma_e, c)]\right)\right\rangle_{M_\mu}\right]. \tag{4}$$

Here $\bar{\theta}$ is the exponential-moving-average Student. We use the Huber penalty $\rho_\delta(u) = u^2/2$ for $|u| \le \delta$ and $\delta(|u| - \delta/2)$ otherwise. The expectation covers training samples, noise, and sampled levels. Appendix A discusses its bounded output derivative.

At inference, modality μ uses $\sigma_i^\mu = 1 - i/N^\mu$. Intermediate steps use the Euler update in Eq. (2) with v_F^μ replaced by v_θ^μ. The last query outputs $f_\theta^\mu(x^\mu_{\sigma_{N^\mu-1}}, \sigma_{N^\mu-1}, c)$. Queries are interleaved in descending noise order using the latest other-modality state. Thus one-step video generation uses $f_\theta^v(x_1^v, 1, c)$, and its output is y_θ^v below.

Flow Matching Distribution Distillation. Consistency constrains local trajectory transitions but does not fully match the Student rollout distribution. We apply distribution distillation to video while retaining consistency supervision for both modalities. Let y_θ^v be a Student-generated video endpoint. With independent $\epsilon_{\mathrm{FMDD}}^v \sim \mathcal{N}(0, I)$, we re-noise its detached value:

$$\widetilde{y}_\sigma^v = (1-\sigma)\bar{y}_\theta^v + \sigma\epsilon_{\mathrm{FMDD}}^v, \qquad \bar{y}_\theta^v = \mathrm{sg}[y_\theta^v]. \tag{5}$$

For $\psi \in \{F, \phi\}$, define the velocity-implied endpoint $d_\psi^v = \widetilde{y}_\sigma^v - \sigma v_\psi^v(\widetilde{y}_\sigma^v, \sigma, c)$. The Teacher Model is F. The auxiliary fake-score model ϕ fits Student rollouts with masked mean-squared velocity error against $\epsilon_{\mathrm{FMDD}}^v - \bar{y}_\theta^v$ on $\widetilde{y}_\sigma^v$. Their normalized endpoint difference is

$$g^v = \mathrm{clip}\left(\alpha^v(d_\phi^v - d_F^v), -\gamma, \gamma\right), \quad \alpha^v = \min\left(\frac{1}{\langle|\bar{y}_\theta^v - d_F^v|\rangle_{M_v} + \varepsilon_{\mathrm{num}}}, \gamma\right). \tag{6}$$

Here clip acts elementwise, $\gamma > 0$ bounds the scale and correction, and $\varepsilon_{\mathrm{num}} > 0$ stabilizes normalization. With video weight $\omega_v > 0$ and ramp $R(s) = \min\{1, s/s_{\mathrm{warm}}\}$ at training step s, the FMDD loss and total objective are

$$L_{\mathrm{FMDD}} = \frac{\omega_v}{2}\mathbb{E}\left[\left\langle (y_\theta^v - \mathrm{sg}[y_\theta^v - g^v])^2 \right\rangle_{M_v}\right], \qquad L_{\mathrm{F2Distill}} = L_{\mathrm{CD}} + R(s)L_{\mathrm{FMDD}}. \tag{7}$$

The two losses act on different objects: L_{CD} aligns Teacher transitions, while L_{FMDD} corrects video endpoints from Student rollouts. The latter supplies a clipped video signal, and L_{CD} retains direct action supervision. Appendix A gives the local gradient interpretation and its limits.

3.3 Repairable and Reliable Speculative Inference

Speculative Inference Procedure. Algorithm 1 separates cached-context verification from the fresh Target computation used for repair and fallback. After executing a returned prefix, the controller updates history with those actions and the next observation.

Draft and Target Models. R2 Spec assigns the lower-budget and higher-fidelity Students as Draft D_{θ_D} and Target T_{θ_T}, with denoising budgets (N_D^v, N_D^a) and (N_T^v, N_T^a). An action proposal has horizon H, and $1 \le H_r \le H$ is the partial-execution or repair prefix. Let o_t contain the current camera observations, robot state, language instruction, and executed-action history. Draft constructs its own current context c_t^D and produces

$$(\widehat{V}_t^d, \widehat{A}_t^d) = D_{\theta_D}(c_t^D, N_D^v, N_D^a). \tag{8}$$

The Target never consumes Draft-internal features or KV tensors. Initial verification pairs current observed inputs with cached Target video features c_t^- from the last refresh.

A fresh $c_t^f = \mathrm{TargetContext}(o_t)$ runs the Target video branch on current observations when the cache is absent or a rejected draft enters repair or fallback. These Target features replace the cache for the next call, while an accepted Draft prefix leaves it unchanged. Only observed frames and executed actions update history. Unexecuted Draft actions and Draft video are discarded rather than committed as observations. $\mathrm{FullTarget}(c_t^f)$ is complete generation by the *higher-budget Student*, not the Teacher.

Let C_D, C_V, C_X, C_R, C_F denote Draft generation, initial verification, fresh Target-context refresh, repair with recheck, and *incremental* complete Target generation. For event frequencies p_X, p_R, p_F, approximate accounting gives $\mathbb{E}[C] = C_D + C_V + p_X C_X + p_R C_R + p_F C_F$. Events can overlap. Standalone Target-only latency is not C_F, and the reported mean is measured independently.

Algorithm 1 Draft–Verify–Repair–Reverify

Require: Observation o_t, cached Target context c_t^-, models $D_{\theta_D}, T_{\theta_T}$, and horizons H, H_r
Ensure: Draft, repaired, or Target Model action prefix

$c_t^D \leftarrow \mathrm{DraftContext}(o_t)$, $(\widehat{V}_t^d, \widehat{A}_t^d) \leftarrow D_{\theta_D}(c_t^D, N_D^v, N_D^a)$
if c_t^- is unavailable **then**
  $c_t^f \leftarrow \mathrm{TargetContext}(o_t)$, $c_t^- \leftarrow c_t^f$
end if
$(L_{\mathrm{init}}, q_{H_r}, q_H) \leftarrow \mathrm{Verify}(\widehat{A}_t^d, c_t^-, \mathcal{S}_{\mathrm{init}}, \delta_{\mathrm{ver}}, \epsilon_{\mathrm{init}}^a)$
if $L_{\mathrm{init}} = H$ and $q_H = 1$ **then**
  return $\widehat{A}_{t,1:H}^d$
else if $L_{\mathrm{init}} \geq H_r$ and $q_{H_r} = 1$ **then**
  return $\widehat{A}_{t,1:H_r}^d$ for execution, then reobserve and replan
else if $L_{\mathrm{init}} = 0$ and $q_{H_r} = 1$ **then**
  $c_t^f \leftarrow \mathrm{TargetContext}(o_t)$ if not already computed
  Compute $c_t^\star$ and $\widehat{A}_t^r$ by Eqs. (12)–(13)
  $(L_{\mathrm{re}}, q_{\mathrm{re}}) \leftarrow \mathrm{Verify}_{H_r}(\widehat{A}_t^r, c_t^f, \{\sigma_{\mathrm{re}}\}, \delta_{\mathrm{ver}}, \epsilon_{\mathrm{re}}^a)$
  $q_{\mathrm{ok}} \leftarrow \mathbb{I}[L_{\mathrm{re}} \geq H_r]\mathbb{I}[\|M_r \odot (\widehat{A}_t^r - \widehat{A}_t^d)\|_F \leq \kappa] q_{\mathrm{re}}$
  if $q_{\mathrm{ok}} = 1$ **then**
    return $\widehat{A}_{t,1:H_r}^r$
  end if
end if
$c_t^f \leftarrow \mathrm{TargetContext}(o_t)$ if not already computed
return $\mathrm{FullTarget}(c_t^f)$

Multi-Level Flow-Consistency Verification. To avoid full sequential denoising, initial verification evaluates the draft at levels $\mathcal{S}_{\mathrm{init}} = \{\sigma_1, \ldots, \sigma_K\}$ in parallel. With shared Gaussian action noise $\epsilon_{\mathrm{init}}^a$, the Target endpoint at level k is

$$\begin{aligned} \widetilde{A}_{t,\sigma_k} &= (1-\sigma_k)\widehat{A}_t^d + \sigma_k \epsilon_{\mathrm{init}}^a, \\ \widehat{A}_{T,t}^{(k)} &= \widetilde{A}_{t,\sigma_k} - \sigma_k v_T^a(\widetilde{A}_{t,\sigma_k}, \sigma_k, c_t^-). \end{aligned} \tag{9}$$

Let $\widehat{a}_{T,t+h-1}^{(k)}$ and $\widehat{a}_{t+h-1}^d$ be the corresponding action steps of $\widehat{A}_{T,t}^{(k)}$ and $\widehat{A}_t^d$. Their continuous-channel discrepancy is

$$d_h^{(k)} = \mathrm{Dist}_{\mathrm{cont}}\left(\widehat{a}_{T,t+h-1}^{(k)}, \widehat{a}_{t+h-1}^d\right), \quad \mathrm{Dist}_{\mathrm{cont}}(A,B) = \frac{\|M_{\mathrm{cont}} \odot (A-B)\|_2}{\sqrt{\|M_{\mathrm{cont}}\|_1}}. \tag{10}$$

M_{cont} selects continuous channels. For a prefix it broadcasts over time, and $\mathrm{Dist}_{\mathrm{cont}}$ uses its Frobenius norm divided by the square root of the active-entry count. Discrete gripper decisions, when present, are checked separately. At tolerance δ_{ver}, the verified prefix is

$$L_{\mathrm{init}} = \min_{k \in \{1,\ldots,K\}} \sum_{h=1}^{H} \prod_{j=1}^{h} \mathbb{I}[d_j^{(k)} \leq \delta_{\mathrm{ver}}]. \tag{11}$$

If $L_{\mathrm{init}} \geq H_r$, every checked level meets the continuous-action tolerance through step H_r. Appendix B gives the resulting prefix norm bound. Verify returns $q_\ell = 1$ when discrete decisions agree through step ℓ at all checked levels. Execute all H Draft actions only if $L_{\mathrm{init}} = H$ and $q_H = 1$, or the first H_r if $L_{\mathrm{init}} \geq H_r$ and $q_{H_r} = 1$. A zero prefix enters repair only when $q_{H_r} = 1$. Other cases use fresh complete Target generation.

Fresh-Target Re-anchoring and Repair. At repair level σ_h, the Target Model reconstructs endpoints under cached and fresh contexts:

$$z_h = (1-\sigma_h)\widehat{A}_t^d + \sigma_h \epsilon_{\mathrm{init}}^a, \qquad \widehat{A}_{T,t}^{\xi} = z_h - \sigma_h v_T^a(z_h, \sigma_h, c_t^{\xi}), \quad \xi \in \{-, f\}. \tag{12}$$

Let $e_{\mathrm{fresh}} = \mathrm{Dist}_{\mathrm{cont}}(\widehat{A}_{T,t,1:H_r}^-, \widehat{A}_{T,t,1:H_r}^f)$. Both endpoints use the same z_h and noise level, so their difference measures the Target's response to a context change on the candidate execution prefix.

Using threshold δ_{fresh}, repair sets $c_t^\star = c_t^f$ if $e_{\text{fresh}} > \delta_{\text{fresh}}$ and c_t^- otherwise. For $\sigma_l < \sigma_h$, take one Target Euler step:

$$z_l' = z_h + (\sigma_l - \sigma_h) v_T^a(z_h, \sigma_h, c_t^\star), \qquad \widetilde{A}_t^r = z_l' - \sigma_l v_T^a(z_l', \sigma_l, c_t^\star). \tag{13}$$

Set $\widehat{A}_t^r = \widehat{A}_t^d + M_r \odot (\widetilde{A}_t^r - \widehat{A}_t^d)$, where M_r masks normalized continuous channels in the first H_r actions. All other channels remain unchanged. With independent noise ϵ_{re}^a, a single-level Verify_{H_r} rechecks the repaired prefix under c_t^f at σ_{re}. Execute it only if $L_{\text{re}} \geq H_r$, its discrete gate $q_{\text{re}} = 1$, and the masked Frobenius norm $\|M_r \odot (\widehat{A}_t^r - \widehat{A}_t^d)\|_F \leq \kappa$. Otherwise, use $\text{FullTarget}(c_t^f)$. The independent, single-level recheck limits computation but is not stronger than the initial multilevel test merely by being independent.

An accepted repair changes the Draft by at most κ on its mask and meets the Target tolerance under independent noise. This constrains agreement with the Target, not correctness in the environment.

4 Experiments

4.1 Implementation Details

Model Architecture. DiSpecWAM uses LingBot-VA's 30-layer video–action Transformer (Li et al., 2026b). The $(25, 50)$ WAM is the Teacher. Students with $(N_D^v, N_D^a) = (1, 2)$ and $(N_T^v, N_T^a) = (2, 4)$ serve as Draft and Target, respectively. All process three RGB views, two future frames, and 32-action chunks.

Training Data and Setup. RoboTwin 2.0 (Chen et al., 2025) covers 50 bimanual tasks. Our training subset contains 25,000 clean and augmented trajectories with 5.53 million transitions. We train for 3,000 steps on eight NVIDIA A800 80GB GPUs with bfloat16 and FSDP (Zhao et al., 2023). Global batch sizes are 32 for consistency and 8 for FMDD. AdamW (Loshchilov & Hutter, 2017) uses a 5×10^{-6} learning rate, 100 warmup steps, 2.0 gradient clipping, and 0.995 EMA. We use $\delta_{\text{CD}} = 0.1$, $s_{\text{warm}} = 100$, $H = 32$, and $H_r = 8$. Verification uses $\mathcal{S}_{\text{init}} = \{0.25, 0.5\}$. Repair uses $(\sigma_h, \sigma_l, \sigma_{\text{re}}) = (0.5, 0.25, 0.375)$, $\delta_{\text{fresh}} = \delta_{\text{ver}} = 0.15$, and $\kappa = 0.75$.

Benchmarks and Baselines. For our RoboTwin 2.0 runs, each of 50 tasks has 100 trials per Clean or Random condition per seed, so a single-seed rate changes in 0.02-point increments. Table 4 averages multiple seeds. Baseline rows may use other denominators. LIBERO (Liu et al., 2023) includes Spatial, Object, Goal, and Long.

4.2 Main Results

Table 1: **RoboTwin 2.0 and LIBERO results.** Latency values may come from different hardware and evaluation protocols. Speedup is the arithmetic ratio to the displayed LingBot-VA latency, not a hardware-normalized comparison. LIBERO averages use the four suites.

Method	RoboTwin 2.0 SR (%) ↑			LIBERO SR (%) ↑					Efficiency	
	Clean	Random	Avg.	Spatial	Object	Goal	Long	Avg.	Lat. (ms) ↓	Speedup ↑
Motus (Bi et al., 2026)	88.66	87.02	87.84	96.8	99.8	96.6	97.6	97.7	3215.00	1.06×
Flash-WAM (Akbari et al., 2026)	88.42	82.66	85.54	97.0	92.8	96.4	98.0	96.1	210.29	16.19×
Fast-WAM (Yuan et al., 2026)	91.88	91.78	91.83	98.2	100.0	97.0	95.2	97.6	232.20	14.66×
PFD (Fang et al., 2026)	93.11	92.69	92.90	98.6	99.2	99.2	95.4	98.1	244.16	13.95×
WLA-0 (Yang et al., 2026)	92.94	90.02	91.48	99.0	100.0	97.8	97.6	98.6	256.34	13.28×
Light-WAM (Li et al., 2026c)	76.40	76.30	76.35	98.2	99.6	97.8	93.0	97.2	378.56	8.99×
LaWAM (Chen et al., 2026)	92.64	89.80	91.22	99.4	99.6	98.4	97.0	98.6	303.78	11.21×
MECo-WAM (Zhang et al., 2026a)	93.26	91.98	92.62	98.8	100.0	98.2	95.8	98.2	232.29	14.66×
LingBot-VA (Li et al., 2026b)	92.90	91.50	92.20	98.5	99.6	97.2	98.5	98.5	3404.85	1.00×
DiSpecWAM (Ours)	**93.50**	**91.80**	**92.65**	**99.0**	**100.0**	**98.8**	**97.8**	**98.9**	**205.00**	16.61×

Overall Performance. DiSpecWAM obtains 93.50% Clean, 91.80% Random, and 92.65% average RoboTwin 2.0 success, respectively 0.60, 0.30, and 0.45 points above the displayed LingBot-VA configuration. PFD has a higher RoboTwin average (92.90%). On LIBERO, DiSpecWAM averages 98.90% versus LingBot-VA's 98.50%.

Latency Analysis. The mean inference call takes 205.00 ms per 32-action proposal versus 3404.85 ms for the evaluated full-budget LingBot-VA configuration, an arithmetic 16.61× ratio. Because R2 Spec may execute only H_r actions before replanning, this is not compute time per executed action. Their RoboTwin averages are 92.65% and 92.20%.

Video Fidelity. Table 2 compares matched v1/a2 and v2/a4 budgets. LPIPS (Zhang et al., 2018) and SSIM (Wang et al., 2004) assess frame fidelity. FVD (Unterthiner et al., 2019) compares video-feature distributions. Clean SR reports task success alongside visual quality. At both budgets, DiSpecWAM improves all four measures over Flash-WAM (Akbari et al., 2026). In R2 Spec, v1/a2 supplies Draft proposals, and v2/a4 verifies them as Target.

Method / Budget	LPIPS ↓	SSIM ↑	FVD ↓	Clean SR ↑
Full (v25/a50)	**0.0971**	**0.8902**	**39.81**	**92.90**
Flash-WAM v1/a2	0.1424	0.8884	95.53	88.42
Flash-WAM v2/a4	0.1248	0.8889	66.70	90.24
DiSpecWAM v1/a2	**0.1287**	**0.8892**	**73.82**	**91.48**
DiSpecWAM v2/a4	**0.1096**	**0.8897**	**49.64**	**92.50**

Table 2: Future-video fidelity and RoboTwin 2.0 Clean SR. Full denotes LingBot-VA.

Internal Feature Diagnostic. We compare final-block video features for one RoboTwin 2.0 handover block input. Full LingBot-VA, its v2/a4 schedule, and Flash-WAM receive the same initial camera crops, instruction, and random seed. Each yields 240 paired tokens of dimension 3072. Shared PCA and t-SNE (van der Maaten & Hinton, 2008) give the visualization in Figure 3.
In the original feature space, median paired-token distances to Full are 4.03 for v2/a4 and 9.42 for Flash-WAM, with cosine similarities of 0.987 and 0.936. The plotted v2/a4 schedule reuses Full's checkpoint, so it is distinct from the distilled Target in Table 2 and cannot isolate F2 Distill. This single input and two-dimensional projection do not establish distribution alignment across tasks.

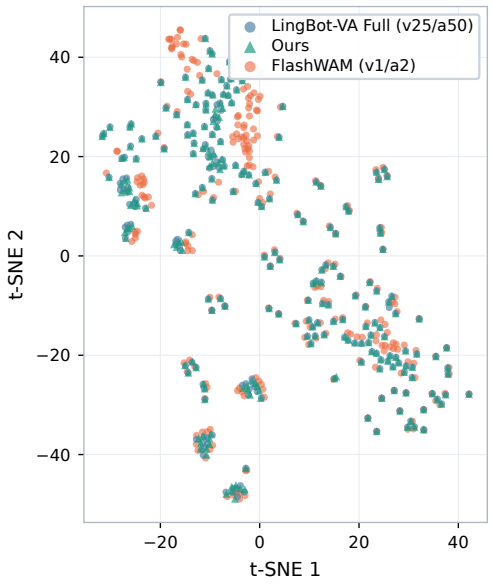


Figure 3: Joint t-SNE of final-block video tokens for one RoboTwin 2.0 handover block input.

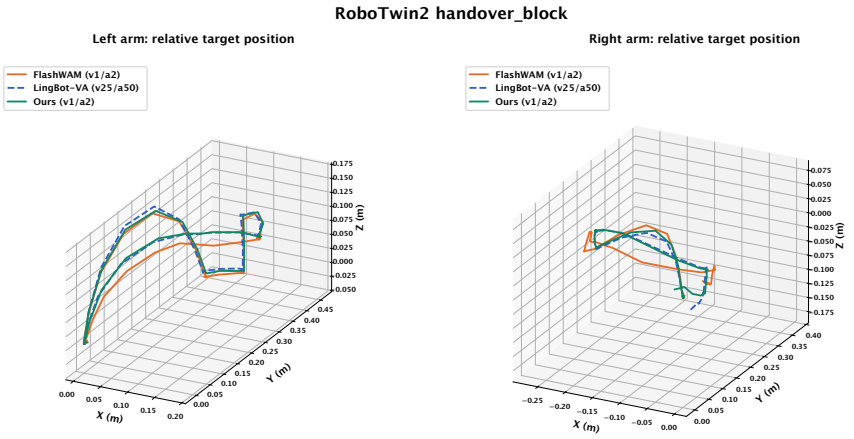


Figure 4: Left- and right-arm relative target positions (m) in one RoboTwin 2.0 handover_block episode. At v1/a2, DiSpecWAM follows the full LingBot-VA (v25/a50) trajectory more closely than Flash-WAM.

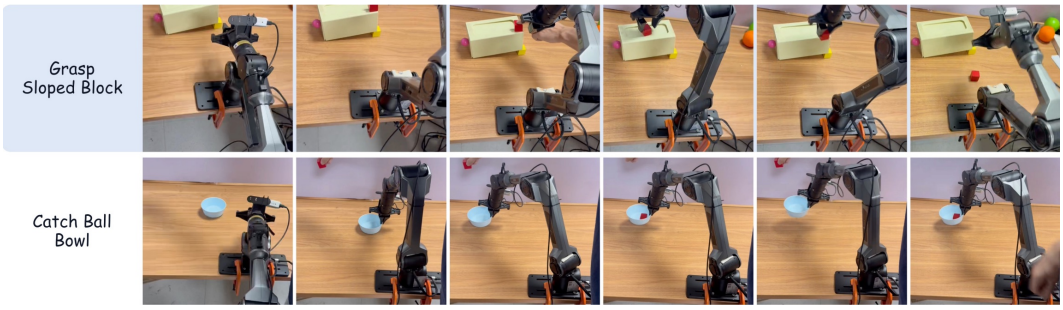


Figure 5: AgileX Piper task sequences. Top, Grasp Sloped Block: the arm catches a block sliding off a box. Bottom, Catch Ball Bowl: the gripper moves a bowl to catch the falling object.

4.3 Real-World Evaluation

Hardware and Task Setup. On an AgileX Piper arm, we evaluate Grasp Sloped Block, Catch Ball Bowl, Grasp Rolling Ball, and Press Bell Timed. Each method has 100 trials per task, or 400 trials across the four tasks.

Performance Analysis. DiSpecWAM averages 79.75% success, 7.25 points above Fast-WAM (72.50%), the strongest baseline. Figure 6 gives task success. Its largest gains over the strongest per-task baseline are 12 points on Grasp Sloped Block and 9 points on Grasp Rolling Ball. The gains on Catch Ball Bowl and Press Bell Timed are one point each. Figures 5 and 7 show executions.

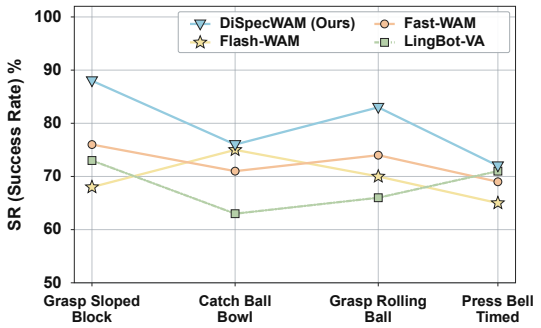


Figure 6: AgileX Piper success rates (100 trials per task and method). The arm catches a block sliding off a box, moves a bowl under a falling object, grasps a ball rolling across a white sheet, or presses a bell before a deadline.

4.4 Ablation Studies

Impact of Components. Draft-only omits Target checks and recovery, while Target-only generates every chunk with the higher-budget model. Their displayed Clean SR values trail DiSpecWAM by 2.02 and 1.00 points, although the systems may reobserve at different frequencies. Without repair, success drops by 0.70 points and latency rises from 205 to 256 ms. The w/o Consistency variant removes both video and action consistency supervision, whereas w/o Distribution Alignment removes video FMDD alone. Their 21.04- and 9.33-point drops therefore do not isolate complementarity between the two video losses.

Table 3: Denoising-budget trade-off.

N^v	N^a	Clean SR ↑	Lat. (ms) ↓	Speedup ↑
1	1	88.70	156	21.8×
1	**2**	91.48	183	18.6×
2	**4**	**92.50**	**376**	**9.1**×
4	8	92.74	762	4.5×
25	50	92.90	3405	1.0×

Table 4: Component ablation (RoboTwin 2.0 Clean).

Method Variant	Clean SR ↑	Latency ↓
DiSpecWAM (Full)	**93.50**	**205 ms**
w/o Speculative Sampling	–	–
Draft Model Only	91.48	183 ms
Target Model Only	92.50	376 ms
w/o Repair	92.80	256 ms
w/o Consistency	72.46	216 ms
w/o Distribution Alignment	84.17	205 ms

Denoising-Budget Analysis. Draft $(1, 2)$ gains 2.78 points over $(1, 1)$ for 27 ms more. Target $(2, 4)$ trails Full by 0.40 points at 9.1× speedup.

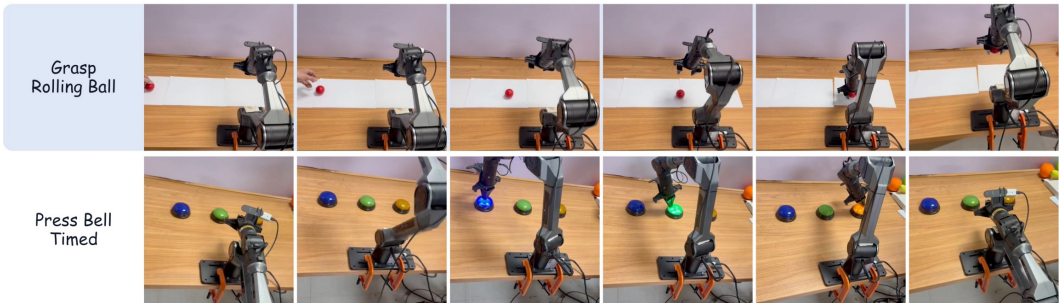


Figure 7: AgileX Piper task sequences. Top, Grasp Rolling Ball: the gripper catches a ball rolling across a white sheet. Bottom, Press Bell Timed: the arm presses a bell before the time limit expires.

Table 5: RoboTwin 2.0 Clean: verification threshold. Draft, Repair, and Fallback report final return routes.

δ_{ver}	Clean SR ↑	Draft	Repair	Fallback/Lat. ↓
0.20	92.94	81%	16%	3% / 196 ms
0.15	**93.50**	**74%**	**22%**	**4% / 205 ms**
0.10	93.18	66%	26%	8% / 218 ms

Table 6: Repair bound (RoboTwin 2.0 Clean).

κ	Clean SR ↑	Accept ↑	Fallback/Lat. ↓
0.50	93.06	72.4%	6.5% / 229 ms
0.75	**93.50**	**84.6%**	**4.0% / 205 ms**
1.00	93.14	88.9%	2.8% / 195 ms

Verification and Repair Settings. At $\delta_{\text{ver}} = 0.15$ and $\kappa = 0.75$, Clean SR reaches 93.50% at 205 ms. The 84.6% figure is the fraction of repair attempts accepted by independent reverification, not the fraction of failed tasks rescued in the environment. Final fallback accounts for 4.0% of calls.

Task-Level Behavior. Figure 8 shows DiSpecWAM outperforming the full WAM and reduced-budget variants on four RoboTwin 2.0 Clean tasks, especially Hanging Mug and Turn Switch. The bars do not isolate Draft, verification, or repair effects, which Table 4 tests.

4.5 Discussion and Limitations

Table 4 shows 2.02 more Clean SR points than Draft-only for 22 ms more per call and 1.00 more point than Target-only while saving 171 ms. Different execution-prefix lengths and replanning frequencies prevent attributing these gains solely to repair. Matched-prefix controls and stagewise timing are needed.
Appendix A gives a conditional endpoint analysis. Shared Draft–Target errors can evade verification, and stale context can foil repair or fallback.

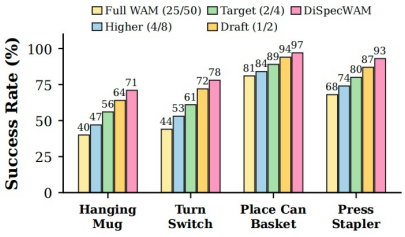


Figure 8: Task-level success on RoboTwin 2.0 Clean, with 100 trials per task.

5 Conclusion

We present **DiSpecWAM**, a unified WAM acceleration framework that integrates high-fidelity few-step video–action distillation with adaptive, repairable speculative inference. On RoboTwin 2.0, DiSpecWAM achieves 93.50% Clean and 91.80% Random success with 205.00 ms mean latency per 32-action inference call, compared with 92.90%, 91.50%, and 3404.85 ms for the evaluated full-budget LingBot-VA configuration. It also achieves 98.90% average success on LIBERO and 79.75% mean success across four AgileX Piper tasks, exceeding Fast-WAM by 7.25 points on the latter. These results indicate a favorable measured success–latency trade-off for future-aware control through training-time compression and inference-time recovery. Future work will evaluate verifier generalization and repair coverage across more real-world tasks and platforms, with particular attention to verification errors and fallback failures in deployment.

REFERENCES

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

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

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

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

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

Pengcheng Fang, Hongli Chen, and Xiaohao Cai. Privileged foresight distillation: Zero-cost future correction for world action models. *arXiv preprint arXiv:2604.25859*, 2026.

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

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International conference on machine learning*, pp. 19274–19286. PMLR, 2023.

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

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

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified Video Action Model. In *Proceedings of Robotics: Science and Systems*, June 2025. doi: 10.15607/RSS.2025.XXI.074. URL https://www.roboticsproceedings.org/rss21/p074.html.

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*, 2026c.

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*, volume 36, 2023. doi: 10.52202/075280-1939. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8c3c666820ea055a77726d66fc7d447f-Abstract-Datasets_and_Benchmarks.html.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Jiahui Niu, Kefan Gu, Yucheng Zhao, Shengwen Liang, Tiancai Wang, Xing Hu, Ying Wang, and Huawei Li. Realtime-vla flash: Speculative inference framework for diffusion-based vlas. *arXiv preprint arXiv:2605.13778*, 2026.

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *International Conference on Learning Representations*, volume 2024, pp. 15078–15097, 2024.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. FVD: A New Metric for Video Generation. In *ICLR 2019 Workshops: DeepGenStruct*, 2019. URL https://mlanthology.org/iclrw/2019/unterthiner2019iclrw-fvd/.

Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL https://www.jmlr.org/papers/v9/vandermaaten08a.html.

Yuqi Wang, Xinghang Li, Wenxuan Wang, Junbo Zhang, Yingyan Li, Yuntao Chen, Xinlong Wang, and Zhaoxiang Zhang. Unified vision-language-action model. In *International Conference on Learning Representations*, volume 2026, pp. 80929–80944, 2026.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

Yi Yang, Zhihong Liu, Siqi Kou, Yiyang Chen, Yanzhe Hu, Jianbo Zhou, Boyuan Zhao, Zhijie Wei, Xiao Xia, Xueqi Li, et al. World-language-action model for unified world modeling, language reasoning, and action synthesis. *arXiv preprint arXiv:2606.05979*, 2026.

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

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

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6613–6623. IEEE, 2024.

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

Jianjun Zhang, Jian Zhu, Taiyi Su, Chong Ma, Zitai Huang, Yi Xu, and Hanli Wang. Learning 4d geometric priors for inference-efficient world action models. *arXiv preprint arXiv:2607.05468*, 2026a.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_The_Unreasonable_Effectiveness_CVPR_2018_paper.html.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, et al. Dreamvla: a vision-language-action model dreamed with comprehensive world knowledge. *Advances in Neural Information Processing Systems*, 38:24195–24228, 2026b.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, et al. Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*, 2023.

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

A THEORY OF FLOW-MATCHING DISTRIBUTION DISTILLATION

Let $p^c_{T,v}$ and $p^c_{S,v}$ be the Teacher and Student video-endpoint distributions conditioned on c. The consistency loss aligns sampled transitions, but does not establish $p^c_{S,v} = p^c_{T,v}$. Let $d_T = d^v_F$, $d_S = d^v_S$, and $d_\phi = d^v_\phi$ denote endpoint estimates at the same re-noised input. These estimates differ from the video consistency output f^v_θ in Eq. (3).

A.1 TEACHER-TRANSITION AND HUBER BOUNDS

For a sampled conditional path, let $e^\mu_F = v^\mu_F - (\epsilon^\mu - x^\mu_0)$. Equation (2) gives $\widetilde{x}^\mu_{\sigma_e} - x^\mu_{\sigma_e} = (\sigma_e - \sigma_s) e^\mu_F$. Thus shortening a transition reduces its displacement for a fixed residual. The residual need not be Teacher model error because a marginal velocity can differ from the sampled conditional velocity. The Huber derivative is $\rho'_\delta(u) = \mathrm{clip}(u, -\delta, \delta)$, which bounds each masked residual's influence on the consistency output. It does not bound the full parameter gradient.

A.2 LOCAL ENDPOINT CONTRACTION

At a fixed re-noised input, write $\Delta = d_S - d_T$. Assume $d_\phi = d_S$, the Teacher endpoint and scale α^v remain fixed, and clipping is inactive. An idealized endpoint step of size $\eta > 0$ gives $\Delta^+ = (1-r)\Delta$ for $r = \eta\omega_v\alpha^v$. For $0 < r < 1$ and $D = \|\Delta\|_2^2/2$,

$$D^+ = (1-r)^2 D < D \qquad \text{when } \Delta \neq 0. \tag{A.1}$$

It does not cover parameter-gradient steps, clipping, or distributional convergence.

A.3 FAKE-SCORE ERROR AND OBJECTIVE GRADIENTS

Write $e = d_\phi - d_S$ for the fake-score endpoint error. Under the same fixed-target assumptions, the update gives $\Delta^+ = (1-r)\Delta - re$. The triangle inequality yields

$$\|\Delta^+\|_2 \leq (1-r)\|\Delta\|_2 + r\|e\|_2, \qquad 0 < r < 1. \tag{A.2}$$

If the error is larger than the current discrepancy or points adversely, the local correction need not contract. This is a conditional endpoint statement rather than a distributional convergence theorem.

The gradient of the complete objective has the form $\nabla_\theta L_{\mathrm{CD}} + R(s)\nabla_\theta L_{\mathrm{FMDD}}$. The FMDD term backpropagates through the Student video endpoint, while the action endpoint has no direct FMDD residual. Shared backbone parameters can still affect both outputs, so this design does not mathematically isolate action updates. It does preserve an explicit action-consistency objective while adding a bounded video-distribution signal.

For a fixed detached target, the per-sample video-endpoint gradient of Eq. (7) is $\omega_v M_v \odot g^v / \|M_v\|_1$. Elementwise clipping bounds each coordinate by $\omega_v\gamma/\|M_v\|_1$. This is an endpoint-gradient bound, not a guarantee of optimization stability.

A.4 INTERPRETATION OF COMPONENT ABLATIONS

The w/o Consistency ablation removes L_{CD} for both video and action, while retaining FMDD on video. Conversely, w/o Distribution Alignment removes FMDD while retaining video and action consistency. Their comparison cannot isolate the interaction of the two *video* objectives because the first ablation also removes direct action supervision. A controlled video-objective study would retain action consistency in every variant.

B VERIFICATION AND REPAIR BOUNDS

B.1 VERIFIED PREFIX AND STATE DEVIATIONS

If $L_{\text{init}} \geq H_r$, Eq. (11) makes every checked level satisfy $d_h^{(k)} \leq \delta_{\text{ver}}$ for $h \leq H_r$. With $m_{\text{cont}} = \|M_{\text{cont}}\|_1$ active continuous channels per step, the verified prefix obeys

$$\max_{1\leq k\leq K} \left\| M_{\text{cont}} \odot \left(\widehat{A}^{(k)}_{T,t,1:H_r} - \widehat{A}^{d}_{t,1:H_r} \right) \right\|_F \leq \delta_{\text{ver}} \sqrt{H_r m_{\text{cont}}}. \tag{B.1}$$

For the first H_r steps, Eq. (11) bounds each checked continuous-action discrepancy by $\delta_{\text{ver}}\sqrt{m_{\text{cont}}}$. Let the environment transition $s_{h+1} = \mathcal{T}(s_h, a_h)$ be L_s-Lipschitz in state and L_a-Lipschitz in decoded action. If Draft and Target start from the same state and their discrete decisions agree, their state discrepancy at each checked level k is bounded by

$$\|s^d_{H_r} - s^{T,k}_{H_r}\|_2 \leq L_a \delta_{\text{ver}} \sqrt{m_{\text{cont}}} \sum_{j=0}^{H_r-1} L_s^j. \tag{B.2}$$

Indeed, the Lipschitz assumption gives $\|\Delta s_{h+1}\|_2 \leq L_s \|\Delta s_h\|_2 + L_a \delta_{\text{ver}} \sqrt{m_{\text{cont}}}$, and iterating from $\Delta s_0 = 0$ proves the bound. It measures agreement with the Target rollout, not task correctness, and may grow quickly when $L_s > 1$.

B.2 SENSITIVITY TO TARGET CONTEXT

The fresh-context comparison also admits a conditional sensitivity bound. For fixed z_h, suppose the masked Target velocity is L_c-Lipschitz under $\text{Dist}_{\text{cont}}$ with respect to a norm on fixed-size context embeddings. Equation (12) then gives $e_{\text{fresh}} \leq \sigma_h L_c \|c_t^- - c_t^f\|$. A small context change therefore limits the endpoint change under this assumption. The converse does not follow, so e_{fresh} is a Target-response test rather than a certificate that the observation is unchanged.