CL-WAM: CLOSING THE OBSERVATION LOOP IN WORLD–ACTION MODELS FOR HUMANOID MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

World-action models (WAMs) predict video and generate actions, but refreshing an image context does not explicitly correct a persistent predicted state when contact outcomes diverge from a rollout. We introduce the *Closed-Loop World–Action Model* (CL-WAM), which closes this prediction–observation–action loop inside the world model. At each policy call, the previous belief, memory, and executed action produce a prior; the new observation provides an innovation that corrects the belief and updates memory before the next action chunk and, when scheduled, the next candidate rollout. A port-Hamiltonian-inspired latent field separates a skew/dissipative autonomous term from action-conditioned and predicted-contact terms, while scheduled rollout training feeds real and self-generated visual features through the same belief-update path. At deployment, slow candidate planning periodically refreshes a token that conditions a faster action head. Across four LIBERO suites, CL-WAM achieves 99.2% average success versus 98.6% for the officially released DiT4DiT checkpoint. On 24 RoboCasa-GR1 tasks, it reaches 67.50% $(2430/3600)$ versus 56.7% $(680/1200)$ for official DiT4DiT Run 1, a descriptive 10.8-point difference. Removing belief correction or scheduled rollout training lowers RoboCasa-GR1 success by 3.7 or 4.1 points, respectively. On seven real-world Unitree G1 tasks, observed success is $109/140$ (77.9%) versus $101/140$ (72.1%) for a locally matched DiT4DiT baseline.

1 INTRODUCTION

World-Action Models (WAMs) (Ma et al., 2026; Ye et al., 2026; Guo et al., 2025; Routray et al., 2025; Kim et al., 2026; Liao et al., 2025; Zhang et al., 2025b), built on the success of generative video models, have rapidly become a leading paradigm for generalizable robot manipulation. By coupling a video Diffusion Transformer with an action Diffusion Transformer, methods such as DiT4DiT (Ma et al., 2026) demonstrate that the rich spatiotemporal structure learned during video pre-training directly transfers to low-level control. Across this line of work, modelling video dynamics has emerged as a more data-efficient scaling proxy than the static image-text priors inherited by Vision-Language-Action backbones (Kim et al., 2024; Black et al., 2024; Bjorck et al., 2025).

Despite this progress, contact-rich, long-horizon humanoid manipulation remains difficult. The officially released DiT4DiT checkpoint achieves 56.7% average success in its published Run 1 on the 24-task RoboCasa-GR1 suite, leaving substantial room for improvement on the same benchmark. At the same time, success rates on standard LIBERO suites are already high, making aggregate LIBERO scores less informative about failures during extended interaction. These observations motivate a more specific question: can a video-action model use each new observation to correct a persistent internal state, retain task-relevant information across steps, and train on the predictions it will consume during deployment?

Prior work addresses parts of the feedback problem: World-VLA-Loop (Liu et al., 2026a) uses closed-loop world-model training, recurrent state-space models (Hafner et al., 2020; 2023) retain latent state, and memory-augmented policies (Shi et al., 2025; Cherepanov et al., 2025b) store long-horizon task information. We study the combination of an observation-corrected persistent belief, writable memory, a port-Hamiltonian latent flow, and closed-loop rollout training in one

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

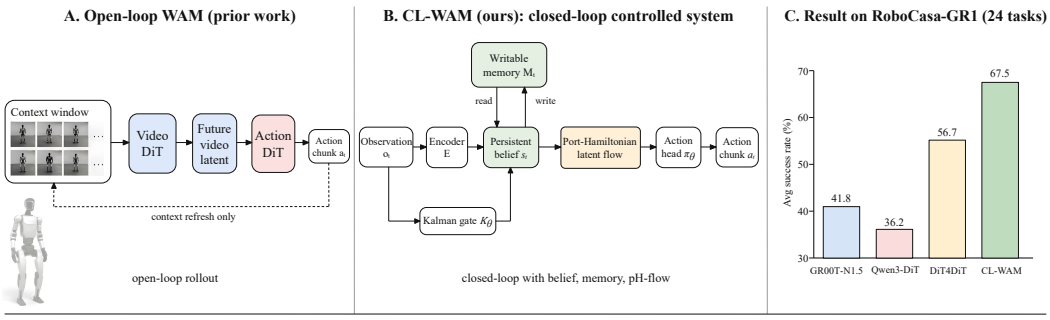


Figure 1: **From open-loop WAMs to a closed-loop controlled system.** Left: existing WAMs roll out a future from a context window; observations only re-enter through context refresh. Centre: CL-WAM maintains a persistent belief s_t updated by a learned innovation gate K_θ, a writable memory M_t for task-level facts, and a port-Hamiltonian latent flow that admits dissipation, control, and a predicted-contact term. Right: on the same 24-task RoboCasa-GR1 benchmark, CL-WAM obtains $\mathbf{67.50}\%$ $(2430/3600)$, compared with 56.7% $(680/1200)$ for Run 1 of the officially released DiT4DiT checkpoint.

video-action backbone. These mechanisms target partial observability, dissipative contact dynamics, and error accumulation during self-generated rollouts (Kalman, 1960; van der Schaft & Jeltsema, 2014; Ross et al., 2011).

We introduce CL-WAM to close the feedback loop around a WAM's persistent predictive state. At each policy call, the previous belief, memory, and executed action form a prior; the newest observation yields an innovation that corrects the belief and updates memory before the next action chunk is generated and before any scheduled plan refresh. It instantiates four tightly coupled primitives: (1) a **persistent belief state** updated at every observation through a learned Kalman-like innovation gate, supervised by an offline smoother; (2) a **writable external memory** with read/write/rewrite operators that stores task-level facts the rollout would otherwise have to re-derive; (3) a **port-Hamiltonian-inspired latent field** $v = (J-R)\nabla H + v_{\mathrm{ctrl}} + v_{\mathrm{contact}} + v_{\mathrm{corr}}$ that separates a dissipative autonomous component from action-conditioned and predicted-contact terms, without equating the learned latent coordinates with measured robot states; and (4) a **closed-loop rollout training** scheme that mixes real and self-generated observations under a Bernoulli schedule, jointly with the belief update. The four primitives are tied together by a dual flow-matching objective and executed at deployment through a two-rate loop in which a heavier world-model planner is refreshed every 24 executed action steps, while a lightweight action pathway generates executable chunks conditioned on the committed plan token between replans.

Across the four LIBERO suites, CL-WAM obtains 99.2% average success versus 98.6% for the officially released DiT4DiT checkpoint. On 24 RoboCasa-GR1 tasks it obtains 67.50% $(2430/3600)$, versus 56.7% $(680/1200)$ for official DiT4DiT Run 1. On seven Unitree G1 tasks, observed success is $109/140$ (77.9%), versus $101/140$ (72.1%) for the local matched baseline. We report task-level shifts and controlled ablations in Section 4.3 and Appendices J and M. Code and reproducibility materials are available in our anonymous repository.

2 Related Work

Video-grounded generative world-action models. DiT4DiT (Ma et al., 2026) jointly trains video and action DiTs. DreamZero, Ctrl-World, ViPRA, Cosmos-Policy, Genie Envisioner, and DreamVLA develop video-grounded control in different forms (Ye et al., 2026; Guo et al., 2025; Routray et al., 2025; Kim et al., 2026; Liao et al., 2025; Zhang et al., 2025b). Action Images represents actions as video frames (Zhen et al., 2026), while Being-H0.7 studies compact latent world-action models (Luo et al., 2026). These works motivate our joint video-action backbone.

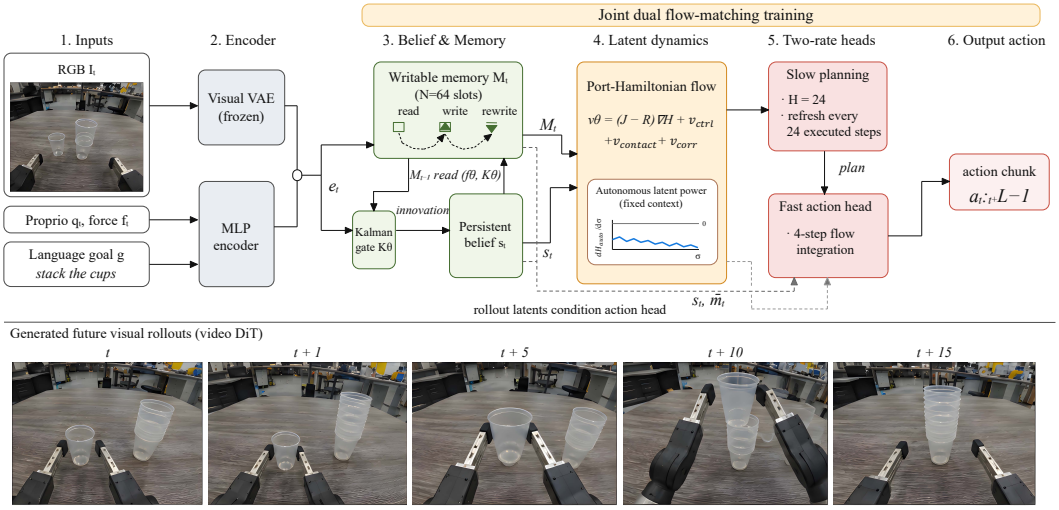


Figure 2: **CL-WAM architecture.** The encoder produces e_t; a memory read conditions the belief prior, the innovation gate corrects the belief using the current observation, and memory is then updated. The shared pH field integrates the corrected belief mean; a post-integration projection forms a token that conditions action-chunk generation. The yellow banner marks components jointly trained under dual flow matching. The inset schematically depicts only the non-positive instantaneous power of the isolated autonomous term under fixed context and continuous pH integration; it does not imply convergence of the complete candidate rollout or the executed robot.

Closed-loop control, memory, and structured dynamics. World-In-World and World-VLA-Loop already study closed-loop world-model training (Zhang et al., 2025a; Liu et al., 2026a); recurrent world models maintain latent state for prediction and control (Hafner et al., 2020; 2023). MemoryVLA, ELMUR, and MIKASA study memory for long-horizon manipulation (Shi et al., 2025; Cherepanov et al., 2025b;a). Hamiltonian and Lagrangian neural models introduce structured inductive biases for learned dynamics (Greydanus et al., 2019; Cranmer et al., 2020). LaDi-WM predicts visual features with latent diffusion and refines actions using imagined states (Huang et al., 2025). Port-Hamiltonian systems provide a separate structural template with dissipation and control inputs (van der Schaft & Jeltsema, 2014). We evaluate observation correction, writable memory, structured latent flow, and closed-loop training together in a video-action backbone with two-rate inference.

3 CL-WAM: Closing the Loop on Video and Action Models

This section details CL-WAM, a Video-Action Model that behaves as a closed-loop controlled system rather than an open-loop future generator. We first set up flow matching and formalise the problem, then describe the architecture and the joint training objective, and close with the two-rate inference loop. An end-to-end overview is given in Figure 2.

3.1 Preliminaries

Flow matching. Flow matching (Lipman et al., 2023) learns a time-dependent velocity field $v(x_\tau, \tau)$ that transports samples between $p_1 = \mathcal{N}(0, I)$ and a data distribution p_0. Under the optimal-transport interpolation $x_\tau = (1-\tau)x_0 + \tau z$ with $\tau \sim \mathcal{U}[0,1]$ and $z \sim \mathcal{N}(0, I)$, the target velocity is $v^\star = z - x_0$, and training minimises $\mathcal{L}_{\text{FM}} = \mathbb{E}\|v(x_\tau, \tau) - (z - x_0)\|_2^2$. Inference solves $\dot{x}_\tau = v(x_\tau, \tau)$ from $x_1 \sim \mathcal{N}(0, I)$ to x_0 using N Euler steps.

Problem statement. At each step t the robot receives an observation $o_t = (I_t, q_t, f_t, g)$ comprising RGB I_t, proprioception q_t, force f_t, and language goal g, and emits an action chunk a_t. A feed-forward WAM can refresh its observation context at later policy calls, yet it need not maintain a

predicted state that the next observation explicitly corrects. We instead define a closed-loop WAM as a tuple $(E, f_\theta, K_\theta, h_\theta, M, v_\theta, \pi_\theta)$ whose per-step computation is

$$e_t = E(o_t), \quad \hat{s}_{t|t-1} = s_{t-1} + \Delta t \cdot f_\theta(s_{t-1}, a_{t-1}, M_{t-1}), \tag{1}$$

$$s_t = \hat{s}_{t|t-1} + K_\theta(\hat{s}_{t|t-1}, M_{t-1})\big(e_t - h_\theta(\hat{s}_{t|t-1})\big), \quad M_t = \mathrm{Mem}_\theta(M_{t-1}, e_t, s_t), \tag{2}$$

$$\frac{dx_\sigma^{\mathrm{pH}}}{d\sigma} = v_\theta(x_\sigma^{\mathrm{pH}}; c_t, a_t^{\mathrm{tok}}), \qquad x_0^{\mathrm{pH}} = s_t,$$

$$a_{t:t+L-1} \sim \pi_\theta\Big(\cdot \mid \mathrm{PHToken}_\theta(x_1^{\mathrm{pH}}), s_t, M_t, \eta_j, g\Big), \quad \sigma \in [0,1]. \tag{3}$$

Here s_t is the corrected belief mean, c_t is the pH conditioning context formed from that belief and the current memory read, and $\mathrm{PHToken}_\theta$ projects the state *after* pH integration into the action-head token space. Equation (1) is a one-step dynamics prior, Equation (2) is a learned innovation update with simultaneous memory rewrite, and Equation (3) is the latent flow producing the next action chunk. The distinguishing feedback path in Eqs. (1)–(2) revises a persistent predictive state before the action-generation step in Eq. (3).

3.2 Architecture

CL-WAM is built on the Cosmos-Predict2.5-2B (NVIDIA, 2025) video backbone and the GR00T-N1.5 (Bjorck et al., 2025) action DiT, mirroring the DiT4DiT (Ma et al., 2026) configuration so comparisons isolate the four primitives we add.

Belief state. The encoder E maps o_t to $e_t \in \mathbb{R}^d$ via the frozen visual VAE and a small MLP on (q_t, f_t, g). The corrected belief mean s_t summarizes pooled visual features, object slots (Locatello et al., 2020), proprioception, and task information; its low-rank-plus-diagonal covariance P_t is maintained separately. Both pH call paths use s_t directly as their initial state. The dynamics prior f_θ is a residual GRU–Transformer with a learned process noise inflated near contact. The innovation gate $K_\theta = U_\theta V_\theta^\top$ is low-rank (r=16) and supervised by an offline smoother E_{post} with K=8 future observations (Rangapuram et al., 2018): $\mathcal{L}_{\text{belief}} = \sum_t \| s_t - \mathrm{sg}[E_{\text{post}}(o_{\le t}, o_{t+1:t+K})] \|_2^2$.

Writable memory. A bank $M_t = \{m_t^{(i)}\}_{i=1}^N \subset \mathbb{R}^{d_m}$ (N=64, d_m=256) stores task-level facts. Reads use slot attention $\bar{m}_t = \sum_i \alpha_{t,i} m_{t-1}^{(i)}$. Writes use a sigmoid gate

$$\beta_t = \sigma(W_\beta[e_t; s_t; \bar{m}_t] - b_\beta), \qquad m_t^{(i)} = (1-\beta_{t,i}) m_{t-1}^{(i)} + \beta_{t,i} \Phi_\theta(e_t, s_t, i), \tag{4}$$

with bias b_β=2 initialising the gate closed. Memory is regularised by InfoNCE retrieval (van den Oord et al., 2018), sparsity, and stability ($\mathcal{L}_{\text{mem}} = \mathcal{L}_{\text{InfoNCE}} + \lambda_{\text{sparse}} \|\beta\|_1 + \lambda_{\text{stable}} \|M_t - M_{t-1}\|_F^2$).

Port-Hamiltonian latent flow. We use a port-Hamiltonian-inspired decomposition as an inductive bias for the learned latent vector field, without treating latent coordinates as calibrated physical state:

$$v_\theta(x; c, a) = (J_\theta - R_\theta(c)) \nabla_x H_\theta(x; c) + v_{\mathrm{ctrl},\theta}(a, c) + v_{\mathrm{contact}}(x; \mathcal{C}_t) + v_{\mathrm{corr}}(x). \tag{5}$$

Here $J_\theta = A_\theta - A_\theta^\top$ is skew-symmetric and $R_\theta = L_\theta D_\theta L_\theta^\top + d_\theta I \succeq 10^{-4} I$, with softplus-gated low-rank and diagonal terms (full parameterization in Appendix C). The context-gated control output $v_{\mathrm{ctrl},\theta}(a, c)$ is added directly to the latent field. The observation-grounded belief prior f_θ is trained with future visual targets through $\mathcal{L}_{\mathrm{CL}}$; the pH field matches a latent interpolation velocity. Its separate training loss uses demonstration actions. The pH token path uses zero action during training and the last executed action (zero after reset) at deployment; candidate scoring uses hypothetical actions. These integrations supply latent conditioning and scoring features. Contact is predicted from observations, and v_{corr} has a soft penalty.

Proposition 1 (Conditional latent energy balance)**.** *Fix the conditioning context c and action input a during one forward pH latent integration, and assume that $H_\theta(x; c)$ has no explicit dependence on the integration coordinate σ. Write $p_\sigma = \nabla_x H_\theta(x_\sigma^{\mathrm{pH}}; c)$. Every differentiable solution of Eq.* (5) *satisfies*

$$\frac{d}{d\sigma} H_\theta(x_\sigma^{\mathrm{pH}}; c) = -p_\sigma^\top R_\theta(c) p_\sigma + p_\sigma^\top v_{\mathrm{ctrl},\theta}(a, c) + p_\sigma^\top \big(v_{\mathrm{contact}} + v_{\mathrm{corr}}\big). \tag{6}$$

Consequently, the isolated autonomous structural term $(J_\theta - R_\theta)\nabla H_\theta$ satisfies $dH_\theta/d\sigma \leq -10^{-4}\|p_\sigma\|_2^2$. This conditional statement does not guarantee decreasing energy, passivity, or stability for the complete learned field or the executed robot rollout. Proof and scope are in Appendix C.

Action head. The action DiT conditions on corrected belief, memory read, a pH token formed with four Euler substeps, and the retained plan token. The fast head starts from fresh Gaussian noise and uses four action-flow steps to generate an executable $L = 12$ action chunk; the selected slow candidate contributes only a conditioning token (Section 3.4).

3.3 Joint training under closed-loop rollouts

Closed-loop schedule. At optimizer update u of U, the probability of retaining a real visual input at future rollout step $k = 1, \ldots, 24$ is

$$\gamma(u) = 0.60 + 0.195[1 + \cos(\pi \,\mathrm{clip}(u/U, 0, 1))], \tag{7}$$

$$\rho_{u,k} = \rho_0 \gamma(u)^{k-1}, \quad \rho_0 = 1, \quad b_{u,k} \sim \mathrm{Bernoulli}(\rho_{u,k}), \tag{8}$$

$$\tilde{z}_{t+k} = b_{u,k} z^{\mathrm{real}}_{t+k} + (1 - b_{u,k}) \hat{z}^{\mathrm{mix}}_{t+k|t+k-1}. \tag{9}$$

Thus $\gamma(u)$ anneals from 0.99 to 0.60, and $\rho_{u,1} = 1$. The belief prior is $\hat{s}^{\mathrm{mix}}_{t+k|t+k-1} = \mathrm{BeliefPred}(s^{\mathrm{mix}}_{t+k-1}, a_{t+k-1}, \bar{m}^{\mathrm{mix}}_{t+k-1}, p^{\mathrm{contact}}_{t+k-1})$; the separate observation model predicts $\hat{z}^{\mathrm{mix}}_{t+k|t+k-1} = \mathrm{VisPred}(\hat{s}^{\mathrm{mix}}_{t+k|t+k-1})$ *before* observation correction. The same draw mixes visual features, object slots, and centres, while proprioceptive and semantic features remain real; the mixed observation then corrects belief and updates memory.

Starting from the same initial corrected belief and memory, a second path recursively uses generated observations without correction. With z^{pool}_{t+k} the temporally aligned, spatially pooled frozen-VAE target, the implemented losses are

$$\mathcal{L}_{\mathrm{CL}} = \frac{1}{24} \sum_{k=1}^{24} \mathrm{MSE}\Big(\hat{z}^{\mathrm{mix}}_{t+k|t+k-1}, \mathrm{sg}\Big[z^{\mathrm{pool}}_{t+k} \Big] \Big), \tag{10}$$

$$\mathcal{L}_{\mathrm{roll}} = \frac{1}{24} \sum_{k=1}^{24} \mathrm{MSE}\Big(\hat{z}^{\mathrm{free}}_{t+k|t+k-1}, \mathrm{sg}\Big[z^{\mathrm{pool}}_{t+k} \Big] \Big) + 0.1 \mathcal{L}_{\mathrm{progress}}. \tag{11}$$

The 24 transitions use 25 observations; $\mathcal{L}_{\mathrm{progress}}$ is the smooth-L_1 loss against demonstration remaining progress from 1 to 0. Appendix D gives the full update order.

Tri-timestep design. Following DiT4DiT (Ma et al., 2026), we decouple three time variables: a uniform video flow timestep $\tau_v \sim \mathcal{U}[0, 1]$, a fixed feature-extraction timestep τ_f for a stable visual condition, and an action flow timestep $\tau_a = 1 - \tau$ with $\tau \sim \mathrm{Beta}(\alpha, \beta)$ that allocates capacity to critical control phases. Figure 6 of the appendix visualises the closed-loop mixing and the tri-timestep densities. The pH interpolation/integration coordinate σ is sampled separately during training; it is distinct from (τ_v, τ_f, τ_a) and is not calibrated to physical time.

Joint objective. We optimize video/action flow matching jointly with eight auxiliary terms:

$$\mathcal{L} = \mathcal{L}_{\mathrm{FM}} + \lambda_1 \mathcal{L}_{\mathrm{pH}} + \lambda_2 \mathcal{L}_{\mathrm{geo}} + \lambda_3 \mathcal{L}_{\mathrm{contact}} + \lambda_4 \mathcal{L}_{\mathrm{belief}} + \lambda_5 \mathcal{L}_{\mathrm{roll}} + \lambda_6 \mathcal{L}_{\mathrm{CL}} + \lambda_7 \mathcal{L}_{\mathrm{mem}} + \lambda_8 \mathcal{L}_{\mathrm{obs}}. \tag{12}$$

Here $\mathcal{L}_{\mathrm{pH}}$ includes separate latent-state flow matching, $\mathcal{L}_{\mathrm{geo}}$ uses encoder-derived slot targets, and $\mathcal{L}_{\mathrm{contact}}$ is a weak contact prior in the reported experiments. The visual MSE in $\mathcal{L}_{\mathrm{obs}}$ is the same term as $\mathcal{L}_{\mathrm{CL}}$. The weights are $(0.5, 0.2, 0.3, 0.5, 0.5, 0.5, 0.2, 0.1)$; Appendix D.2 specifies the four auxiliary losses and their reductions.

3.4 Two-rate inference

At each policy call j, CL-WAM first encodes the current observation o_t, corrects the belief s_t, and updates memory M_t according to Eqs. (1)–(2). Only after this observation update does the slow pathway refresh the plan, if a refresh is due. The pH conditioning path starts directly from the

corrected belief mean, uses the preceding executed chunk's last action (zero after reset) for four Euler substeps, and projects the integrated state into a conditioning token. At a slow refresh, $N = 8$ candidate chunks are sampled with 32 action-flow steps. Candidate n is repeated or truncated to $H = 24$ actions $\bar{a}^{(n)}_{1:24}$. From $b^{(n)}_0 = s_t$, each planning step reads $r^{(n)}_k = \mathrm{Read}(M_t, b^{(n)}_{k-1})$, evaluates the pre-update energy gradient $q^{(n)}_k = \nabla_b H_\theta(b^{(n)}_{k-1})$, and advances $b^{(n)}_k$ with one pH Euler substep conditioned on $\bar{a}^{(n)}_k$. The deployed score is

$$\begin{aligned} J_n = \frac{1}{24}\sum_{k=1}^{24} \mathrm{softplus}\, f_{\mathrm{task}}([b^{(n)}_k; r^{(n)}_k]) \\ + \frac{0.2}{24}\sum_{k=1}^{24} \mathrm{ReLU}\Big((q^{(n)}_k)^\top (J^{(n)}_{\theta,k} - R^{(n)}_{\theta,k}) q^{(n)}_k\Big) \\ + \frac{0.2}{24\cdot 8}\sum_{k=1}^{24}\sum_{m=1}^{8} p^{\mathrm{contact},(n)}_{k,m} + \frac{0.01}{24D}\sum_{k=1}^{24} \|\bar{a}^{(n)}_k\|_2^2. \end{aligned} \tag{13}$$

Here D is the action dimension. The task cost is trained against demonstration remaining progress, not realized success. The implemented autonomous-power ReLU term is zero in exact arithmetic because $J_\theta^\top = -J_\theta$ and $R_\theta \succeq 0$; the other auxiliary component is predicted contact probability. Appendix D specifies the scorer recursion.

The minimum-score candidate provides a plan-conditioning token:

$$n^\star_j = \arg\min_{1\le n\le 8} J_n, \qquad \eta_j = \mathrm{sigmoid}(\gamma_\theta)\ \mathrm{Linear}_\theta\Big(\bar{a}^{(n^\star_j)}_{1:12}\Big). \tag{14}$$

The linear map converts the selected candidate's first 12 actions into one token, and γ_θ is its learned gate parameter. On calls without a slow refresh, the previous η_j is retained. The fast action head then draws *new Gaussian noise* and uses four flow steps, conditioned on the latest belief, memory, pH token, and η_j, to produce the executable action chunk. Neither the selected candidate nor its noise sample initializes this fast sampling.

For RoboCasa, the slow plan is refreshed on the first policy call and every $P = 2$ calls thereafter, with $L = 12$ executed actions per call. On LIBERO the reported cadence is $P = 3$ and $L = 8$, also giving $PL = 24$ executed actions between refreshes.

4 Experiments

We evaluate CL-WAM to address three questions: (1) how CL-WAM compares with state-of-the-art VLA and WAM policies in simulation and on real hardware; (2) whether the closed-loop design remains effective under distribution shift; and (3) which components drive the gain, and whether the improvement can be explained by additional model capacity or deployment-time planning compute.

4.1 Experiment setup

Benchmarks. We evaluate the four LIBERO suites, 24 RoboCasa-GR1 tabletop tasks, and seven Unitree G1 tasks (Figure 3). In the controlled RoboCasa comparison, three independent training seeds and 50 trials per task and seed give 3,600 episodes per method. Official DiT4DiT Run 1 uses one released checkpoint and 1,200 trials on the same 24 tasks; G1 uses 20 trials per task. Appendix E gives task, camera, and training details.

4.2 Comparison against state-of-the-art policies

LIBERO. Table 1 compares the four suites. Under identical downstream training and evaluation, CL-WAM reaches $99.15 \pm 0.14\%$ across three seeds versus $98.65 \pm 0.20\%$ for DiT4DiT (matched), displayed as 99.2% and 98.7%. On Long the rates are $98.0 \pm 0.23\%$ and $96.8 \pm 0.46\%$. The official DiT4DiT (98.6%) and G0.5 (98.9%) rows are external published references. Task counts appear in Appendix J.

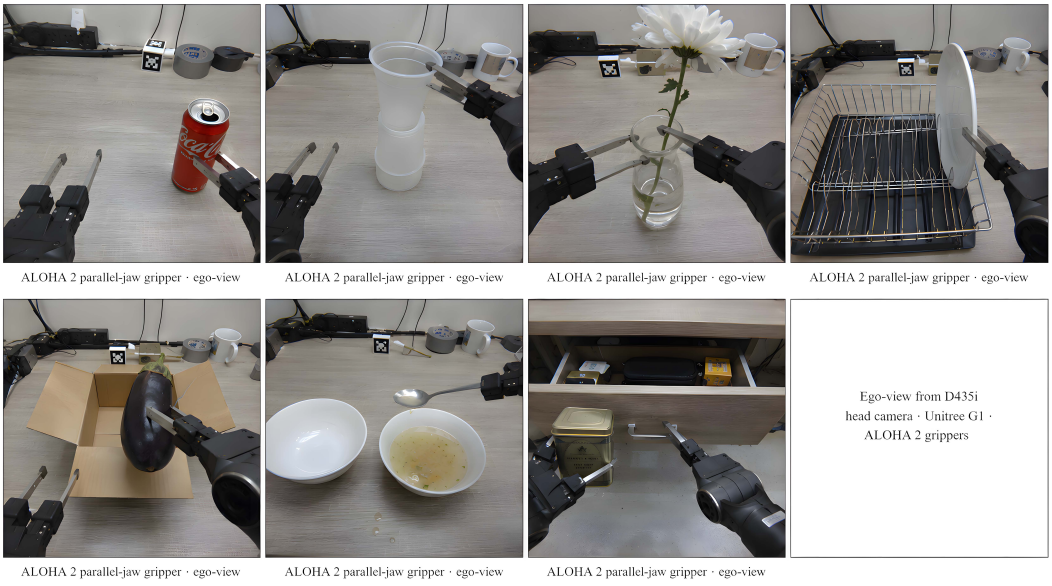


Figure 3: **Seven Unitree G1 evaluation tasks.** Egocentric D435i views of the seven tasks, executed with ALOHA 2 grippers.

Table 1: **LIBERO success rates (%).** Dark/light shading marks best/second best among shown rows; ties share shading. CL-WAM and DiT4DiT (matched) each use three training seeds and 50 trials per task. Subscripts are across-seed standard errors; Avg is computed from each seed's four-suite average. Other rows are published references with potentially different budgets.

Method	Spatial ↑	Object ↑	Goal ↑	Long ↑	**Avg** ↑
Diffusion Policy (Chi et al., 2023)	78.3	92.5	68.3	50.5	72.4
π_0 (Black et al., 2024)	96.8	98.8	95.8	85.2	94.2
$\pi_{0.5}$ (Physical Intelligence et al., 2025)	98.8	98.2	98.0	92.4	96.9
OpenVLA-OFT (Kim et al., 2025)	97.6	98.4	97.9	94.5	97.1
CogVLA (Li et al., 2025)	98.6	98.8	96.6	95.4	97.4
GR00T-N1.5 (Bjorck et al., 2025)	96.2	94.0	96.0	90.0	94.1
Cosmos Policy (Kim et al., 2026)	98.1	100.0	98.2	97.6	98.5
FoMoVLA (Li et al., 2026)	98.4	99.6	99.4	97.6	98.8
G0.5 (Liu et al., 2026b)	98.4	100.0	98.6	98.6	98.9
DiT4DiT (Ma et al., 2026)	98.6	100.0	99.2	96.6	98.6
DiT4DiT (matched) (Ma et al., 2026)	$98.6_{\pm 0.23}$	$100.0_{\pm 0.00}$	$99.2_{\pm 0.12}$	$96.8_{\pm 0.46}$	$98.7_{\pm 0.20}$
CL-WAM (ours)	$\mathbf{99.2}_{\pm 0.12}$	$\mathbf{99.8}_{\pm 0.12}$	$\mathbf{99.6}_{\pm 0.12}$	$\mathbf{98.0}_{\pm 0.23}$	$\mathbf{99.2}_{\pm 0.14}$

RoboCasa-GR1 and real-world G1. On RoboCasa-GR1, CL-WAM obtains $2430/3600$ (67.50%), versus $680/1200$ (56.7%) for official DiT4DiT Run 1, a descriptive 10.8-point difference. The locally matched and $+58$M controls obtain 55.06% and 55.08%. On seven G1 tasks, CL-WAM records $109/140$ (77.9%), versus $101/140$ (72.1%) for matched DiT4DiT; each task has 20 trials. Complete task results are in Appendices J and M. Under protocol-matched three-seed training and evaluation, CL-WAM achieves $2430/3600$ (67.50%) on RoboCasa-GR1, versus $1982/3600$ (55.06%) for DiT4DiT (matched), an observed 12.44-point difference. When the matched baseline is given the eight-candidate slow-loop planning intervention, it reaches 60.6%, leaving an observed 6.90-point gap. The released DiT4DiT checkpoint's Run 1 obtains $680/1200$ (56.7%) and is an external descriptive reference. On the seven G1 tasks, CL-WAM records $109/140$ successes versus $101/140$ for the matched baseline: eight additional successes in this small hardware sample.

Table 2: **RoboCasa-GR1 and G1 success rates (%).** RoboCasa DiT4DiT is official Run 1 (1,200 trials); locally controlled rows use 3,600 trials each. G1 uses 20 trials per task. Dashes denote unavailable results; dark/light shading marks best/second best reported values, with ties shared.

	RoboCasa-GR1	Unitree G1 real-world			
Method	Avg SR	7-task Avg	Stack Cup	Box Packing	Move Spoon
GR00T-N1.5 (Bjorck et al., 2025)	41.8	—	25	0	15
GR00T-N1.6 (Llontop et al., 2026)	40.8	—	—	—	—
Qwen3-DiT (Bai et al., 2025)	36.2	—	5	0	5
DiT4DiT (paper) (Ma et al., 2026)	50.8	—	—	—	—
DiT4DiT (Ma et al., 2026)	56.7	—	—	—	—
DiT4DiT (matched) (Ma et al., 2026)	55.06	72.1	60	50	45
DiT4DiT +58M	55.08	73.6	65	55	45
CL-WAM (ours)	**67.50**	**77.9**	**70**	**60**	**50**

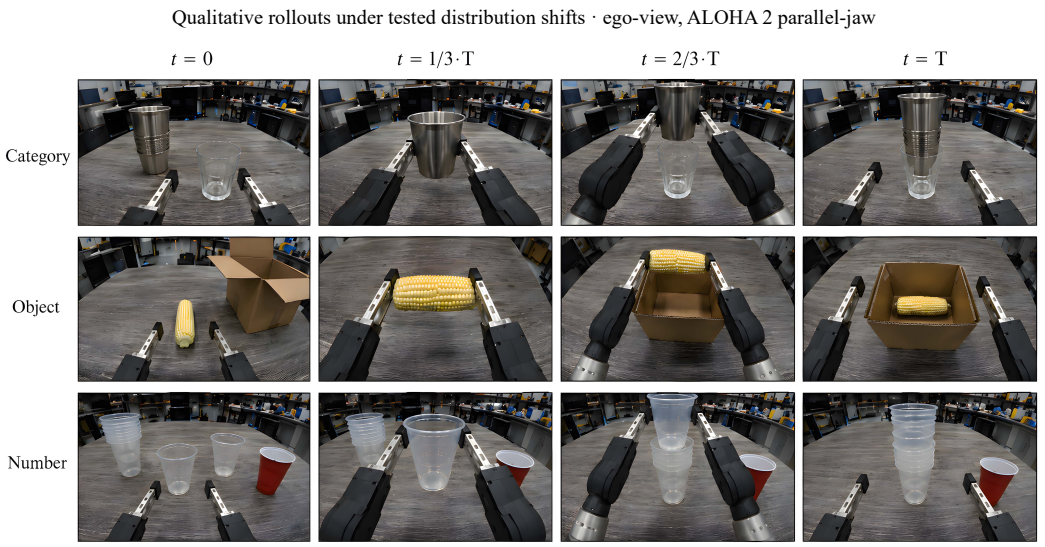


Figure 4: **Successful G1 rollouts under tested distribution shifts.** Rows show cup material, object, and distractor-count changes absent from task-specific G1 fine-tuning demonstrations.

4.3 Performance under tested distribution shifts

We test object substitutions held out from task-specific simulation training and category, object, and distractor shifts held out from G1 fine-tuning. We do not establish absence from earlier pretraining. Figure 4 shows representative rollouts; quantitative results are in Appendices M and L.

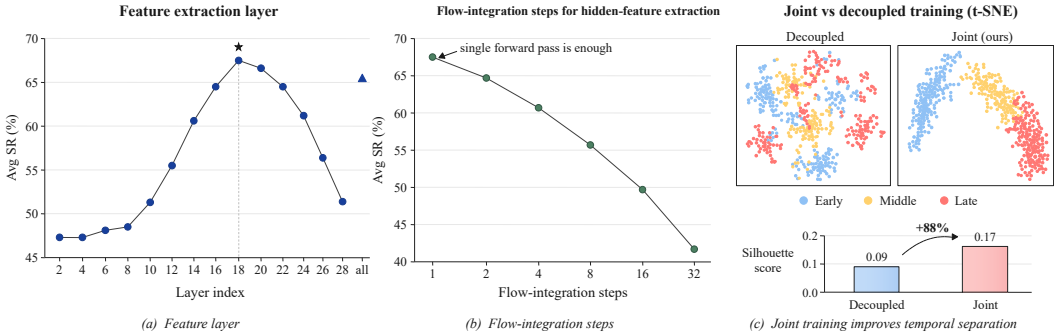
In simulation, CL-WAM reaches 61.0%, 43.4%, and 40.4% on the three tested object substitutions, exceeding DiT4DiT (matched) by 6.5–9.9 points. Across four G1 shifts, each condition has only 20 trials; the observed advantage is one or two additional successes per condition. Per-condition rates and the held-out split are detailed in Appendix M.

4.4 Ablations

Structural ablations and inference controls. Table 3 reports component ablations. Additional capacity and inference controls are discussed in Appendix A. Removing the rollout schedule lowers RoboCasa-GR1 success by 4.10 points, the largest targeted drop. Removing belief correction and memory lowers success by 3.70 and 1.70 points, respectively. Disabling the two pH auxiliary

Table 3: **RoboCasa-GR1 component ablations (avg SR %).** Changes are relative to full CL-WAM. Standard errors are across three training seeds.

Variant	Avg SR	Change (pt)
Full CL-WAM	$67.50_{\pm 1.4}$	—
w/o closed-loop rollout	$63.4_{\pm 1.3}$	-4.10
w/o belief update	$63.8_{\pm 1.5}$	-3.70
w/o pH structural loss	$65.2_{\pm 1.6}$	-2.30
w/o writable memory	$65.8_{\pm 1.2}$	-1.70



(a) Feature layer *(b) Flow-integration steps* *(c) Joint training improves temporal separation*

Figure 5: **Architectural ablations.** (a) Feature extraction layer: success rate peaks at layer 18 of the video DiT. (b) Flow-integration steps for hidden-feature extraction: a single extraction step is optimal. (c) Joint vs decoupled training: t-SNE of hidden features, silhouette score nearly doubles under joint training.

regularizers, while retaining pH flow matching and the structured field, lowers it by 2.30 points. Reducing eight planning candidates to one gives 66.50% versus 67.50%. The planning-pathway-off intervention also changes scoring, plan-token conditioning, and action-flow steps; its -2.92-point difference cannot be attributed to candidate search alone. Appendix A details the interventions.

Architectural ablations. Figure 5 evaluates three backbone design choices. Success rate peaks at video-DiT layer 18; a single integration step performs best for hidden-feature extraction; and joint training increases the temporal-feature silhouette score from 0.09 to 0.17. We use these empirically selected settings in the final model.

4.5 Efficiency

CL-WAM has 527 M trainable parameters (+58 M). Slow/fast updates take 115/32 ms versus 111/30 ms for DiT4DiT (matched); the slow plan refreshes every 24 executed actions. Appendix F gives the latency breakdown.

5 Conclusion

CL-WAM re-frames a WAM as a closed-loop controlled system through four tightly-coupled primitives—persistent belief, writable memory, port-Hamiltonian latent flow, and closed-loop rollout training—under a single dual flow-matching objective. Under the official-aligned evaluation protocol, CL-WAM reaches **99.2**% average success on LIBERO and **67.50**% on RoboCasa-GR1, exceeding Run 1 of the officially released DiT4DiT checkpoint by 10.8 points on the latter. The three-seed matched comparison gives a 12.44-point difference. On the Unitree G1 real-world suite, CL-WAM records 109/140 successful trials (77.9%), compared with 101/140 (72.1%) for DiT4DiT (matched) on the seven tested Unitree G1 tasks.

AI USE STATEMENT

We used generative AI tools to assist with language polishing, improving clarity and concision, shortening the manuscript to meet the page limit, literature discovery, and screening research ideas for potential overlap with prior work. The identified literature and potential overlaps were subsequently checked by the authors through manual review of the original sources. All AI-assisted edits, suggestions, citations, and technical statements were reviewed and verified by the authors before inclusion in the manuscript. Generative AI tools were not treated as authoritative sources, and the authors take full responsibility for the accuracy, originality, experimental results, claims, and final content of the paper.

ETHICS STATEMENT

We do not identify any special ethical concerns arising from this work. The study focuses on robot manipulation in simulation and controlled laboratory environments and does not involve experiments on human subjects or the collection of sensitive personal data. As with general-purpose robot manipulation systems, deployment outside controlled environments may introduce physical safety risks; such deployment should therefore use appropriate safeguards, including collision avoidance, emergency-stop mechanisms, and human supervision. All experiments and reported results were conducted and documented in accordance with standard research integrity practices.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the model architecture, training procedure, inference algorithm, hyperparameters, benchmark protocols, evaluation settings, and ablation configurations in the main paper and appendix. The supplementary material will include an anonymous downloadable source-code repository containing the implementation, configuration files, and scripts required to reproduce the reported experiments, tables, and figures. All datasets and benchmark environments used in this work are publicly available or described with sufficient detail to identify the corresponding public resources.

REFERENCES

Shuai Bai et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

Johan Bjorck, Valts Blukis, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Xiaowei Jiang, Kaushil Kundalia, Jan Kautz, Zhiqi Li, Kevin Lin, Zongyu Lin, Loic Magne, Yunze Man, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Jing Wang, Qi Wang, Shihao Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Seonghyeon Ye, Zhiding Yu, Yizhou Zhao, Zhe Zhang, Ruijie Zheng, and Yuke Zhu. GR00T N1.5: An improved open foundation model for generalist humanoid robots. NVIDIA Research, 2025. URL https://research.nvidia.com/labs/gear/gr00t-n1_5/.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. URL https://arxiv.org/abs/2410.24164.

Ronghan Chen, Yandan Yang, Zuojin Tang, Dongjie Huo, Tong Lin, Haoning Wu, Haoyun Liu, Yuzhi Chen, Lulu Zheng, Botai Yuan, Tianlun Li, Mingxin Wang, Dekang Qi, Bin Hu, Wei Mei, Yuze Xuan, Haolong Yang, Yanqing Zhu, Mu Xu, Zhiheng Ma, and Xinyuan Chang. Abot-m0.5: Unified mobility-and-manipulation world action model. *arXiv preprint arXiv:2607.00678*, 2026. URL https://arxiv.org/abs/2607.00678.

Egor Cherepanov, Nikita Kachaev, Alexey K. Kovalev, and Aleksandr I. Panov. Memory, benchmark & robots: A benchmark for solving complex tasks with reinforcement learning. *arXiv preprint arXiv:2502.10550*, 2025a. URL https://arxiv.org/abs/2502.10550.

Egor Cherepanov, Alexey K. Kovalev, and Aleksandr I. Panov. ELMUR: External layer memory with update/rewrite for long-horizon rl problems. *arXiv preprint arXiv:2510.07151*, 2025b. URL https://arxiv.org/abs/2510.07151.

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

Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian neural networks. In *ICLR Workshop on Integration of Deep Neural Models and Differential Equations*, 2020.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025. URL https://arxiv.org/abs/2510.13626.

Rafal Goebel, Ricardo G. Sanfelice, and Andrew R. Teel. *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. Princeton University Press, 2012.

Sam Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-World: A controllable generative world model for robot manipulation. *arXiv preprint arXiv:2510.10125*, 2025. URL https://arxiv.org/abs/2510.10125.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Yuhang Huang, Jiazhao Zhang, Shilong Zou, Xinwang Liu, Ruizhen Hu, and Kai Xu. LaDi-WM: A latent diffusion-based world model for predictive manipulation. *arXiv preprint arXiv:2505.11528*, 2025. URL https://arxiv.org/abs/2505.11528.

Rudolf E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960. doi: 10.1115/1.3662552.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. URL https://arxiv.org/abs/2406.09246.

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

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

Wei Li, Renshan Zhang, Rui Shao, Jie He, and Liqiang Nie. CogVLA: Cognition-aligned vision-language-action model via instruction-driven routing & sparsification. *arXiv preprint arXiv:2508.21046*, 2025. URL https://arxiv.org/abs/2508.21046.

Wei Li, Peijin Jia, Yuan Ma, Xuefeng Jiang, Titong Jiang, Sheng Sun, Yujian Li, Xin Wen, Han Hong, Zhikang Liu, Bailin Li, and Kun Zhan. Fomovla: Bridging visual foresight and motion guidance for vision-language-action models, 2026. URL https://arxiv.org/abs/2607.14739.

Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Jingbin Cai, Si Liu, Jianlan Luo, Liliang Chen, Shuicheng Yan, Maoqing Yao, and Guanghui Ren. Genie envisioner: A unified world foundation platform for robotic manipulation. *arXiv preprint arXiv:2508.05635*, 2025. URL https://arxiv.org/abs/2508.05635.

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

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

Xiaokang Liu, Zechen Bai, Hai Ci, Kevin Yuchen Ma, and Mike Zheng Shou. World-VLA-Loop: Closed-loop learning of video world model and VLA policy. *arXiv preprint arXiv:2602.06508*, 2026a. URL https://arxiv.org/abs/2602.06508.

Yicheng Liu, Zibin Dong, Baijun Ye, Tianyuan Yuan, Tao Jiang, Anqi Yang, Shicheng Cao, Haonan Liu, Yue Sun, Zihan Guo, Xiao Liu, Dong Ke, Changxun Pan, Chenru Wu, Tailai Cheng, Xiaoshu Ren, Xinlei Zhang, Jianning Cui, Zijie Zhao, Haoyu Zhang, Kaiming Xu, Haodong Yang, Bowen Zhang, Jiahui Niu, Shaoting Zhu, Shiduo Zhang, and Hang Zhao. G0.5: One autoregressive stream for robot reasoning and action, 2026b. URL https://arxiv.org/abs/2608.11739.

Edith Llontop, Yan Chang, and Yuchen Deng. Building generalist humanoid capabilities with NVIDIA isaac GR00T N1.6 using a sim-to-real workflow. NVIDIA Technical Blog, jan 2026.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Hao Luo, Wanpeng Zhang, Yicheng Feng, Sipeng Zheng, Haiweng Xu, Chaoyi Xu, Ziheng Xi, Yuhui Fu, and Zongqing Lu. Being-H0.7: A latent world-action model from egocentric videos. *arXiv preprint arXiv:2605.00078*, 2026. URL https://arxiv.org/abs/2605.00078.

Teli Ma, Jia Zheng, Zifan Wang, Chunli Jiang, Andy Cui, Junwei Liang, and Shuo Yang. DiT4DiT: Jointly modeling video dynamics and actions for generalizable robot control. *arXiv preprint arXiv:2603.10448*, 2026. URL https://arxiv.org/abs/2603.10448.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems (RSS)*, 2024. URL https://www.roboticsproceedings.org/rss20/p050.pdf.

NVIDIA. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025. URL https://arxiv.org/abs/2501.03575.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025. URL https://arxiv.org/abs/2504.16054.

Syama Sundar Rangapuram, Matthias W. Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. Deep state space models for time series forecasting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Artificial Intelligence and Statistics (AISTATS)*, 2011.

Sandeep Routray, Hengkai Pan, Unnat Jain, Shikhar Bahl, and Deepak Pathak. ViPRA: Video prediction for robot actions. *arXiv preprint arXiv:2511.07732*, 2025. URL https://arxiv.org/abs/2511.07732.

Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. MemoryVLA: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. *arXiv preprint arXiv:2508.19236*, 2025. URL https://arxiv.org/abs/2508.19236.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Arjan van der Schaft and Dimitri Jeltsema. Port-Hamiltonian systems theory: An introductory overview. *Foundations and Trends in Systems and Control*, 1(2–3):173–378, 2014. doi: 10.1561/2600000002.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi Fan, and Joel Jang. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026. URL https://arxiv.org/abs/2602.15922.

Jiahan Zhang, Muqing Jiang, Nanru Dai, Taiming Lu, Arda Uzunoglu, Shunchi Zhang, Yana Wei, Jiahao Wang, Vishal M. Patel, Paul Pu Liang, Daniel Khashabi, Cheng Peng, Rama Chellappa, Tianmin Shu, Alan Yuille, Yilun Du, and Jieneng Chen. World-in-world: World models in a closed-loop world. *arXiv preprint arXiv:2510.18135*, 2025a. URL https://arxiv.org/abs/2510.18135.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, Fan Lu, He Wang, Zhizheng Zhang, Li Yi, Wenjun Zeng, and Xin Jin. DreamVLA: A vision-language-action model dreamed with comprehensive world knowledge. *arXiv preprint arXiv:2507.04447*, 2025b. URL https://arxiv.org/abs/2507.04447.

Yuyang Zhang, Wenyao Zhang, Zekun Qi, He Zhang, Haitao Lin, Jingbo Zhang, Yao Mu, Xiaokang Yang, Wenjun Zeng, and Xin Jin. Imagewam: Do world action models really need video generation, or just image editing?, 2026. URL https://arxiv.org/abs/2606.19531.

Haoyu Zhen, Zixian Gao, Qiao Sun, Yilin Zhao, Yuncong Yang, Yilun Du, Pengsheng Guo, Tsun-Hsuan Wang, Yi-Ling Qiao, and Chuang Gan. Action images: End-to-end policy learning via multiview video generation. *arXiv preprint arXiv:2604.06168*, 2026. URL https://arxiv.org/abs/2604.06168v2.

A Discussion, Limitations, and Broader Impact

What the ablations measure. Table 3 reports targeted interventions on the belief update, writable memory, rollout schedule, and pH pathway. Setting `use_ph_structure=false` disables the two 0.01-weighted regularizers in $\mathcal{L}_{\mathrm{pH}}$: the positive-autonomous-power penalty and the v_{corr} magnitude penalty. It retains the pH flow-matching MSE and the structured J_θ, R_θ field; success decreases by 2.30 pt. The power penalty vanishes in exact arithmetic under this parameterization, so this ablation does not isolate a benefit of enforcing skew symmetry or positive semidefiniteness. Replacing the complete structured pH flow with a capacity-matched unconstrained MLP is a separate intervention (-2.20 pt). Removing the belief update, memory, and closed-loop rollout schedule lowers success by 3.70, 1.70, and 4.10 pt, respectively.

Inference controls. We next compare inference-time controls using the same trained checkpoints. Reducing the slow-loop planner from $N{=}8$ candidates to $N{=}1$ yields 66.50% success (-1.00 pt), providing the more focused candidate-count comparison. A broader "planning pathway off (10-step FM)" intervention disables multi-candidate generation and candidate scoring, removes plan-token conditioning, and changes fast action-flow integration from 4 to 10 steps. It reaches 64.58% (-2.92 pt) while retaining the belief update, writable memory, and port-Hamiltonian dynamics pathway. The -2.92-point difference is the combined result of these inference changes and does not estimate the independent contribution of candidate search. The single-rate variant additionally removes the slow/fast two-rate execution schedule. Finally, the capacity-matched DiT4DiT $+58$M control reaches 55.08%, compared with 55.06% for DiT4DiT (matched). Granting DiT4DiT (matched) the same $N{=}8$ slow-loop MPC budget raises it to 60.6%, still 6.90 points below full CL-WAM.

Two empirical findings. First, the applied innovation correction $\|K_\theta r_t\|_F$ is more than four times larger on contact steps than on free-flight steps in held-out RoboCasa-GR1 trajectories (App. H). This statistic combines the learned gate and innovation residual; it does not measure the gate alone. Second, memory specialisation emerges without per-slot supervision: 9 of 64 slots achieve probe AUC above 0.9 on cup-grasped, 7 on microwave-open, and 13 on sub-goal-completed.

Limitations. The standard four-suite LIBERO benchmark is close to saturation: CL-WAM improves the matched three-seed aggregate average from 98.7% to 99.2%. We therefore supplement this saturated benchmark with a controlled evaluation on the ten LIBERO-Long tasks, where CL-WAM reaches 98.0% versus 96.8% for the matched DiT4DiT baseline ($+1.2$ pt). Nevertheless, both settings operate in a high-success regime, so LIBERO alone provides limited resolution for distinguishing strong policies; we therefore use RoboCasa-GR1 as the primary quantitative comparison and treat the G1 results as complementary small-sample hardware evidence. One additional success changes a task-level rate by 5 percentage points; the observed seven-task difference of 77.9% versus 72.1% corresponds to eight trials out of 140. Each real-world OOD condition likewise has only 20 trials, so these hardware results describe the tested conditions rather than a stable effect across untested shifts.

The Unitree G1 study covers seven tasks with 200 demonstrations and 20 rollouts each—sufficient as a case study but smaller than industrial-scale humanoid evaluation. The port-Hamiltonian latent flow is a structural inductive bias rather than a validated physical model. Its energy identity applies to a fixed-context continuous forward pH integration; control, contact, the correction head, changing observations, and Euler discretization prevent us from asserting a full-model passivity or rollout-stability guarantee. Our memory bank is fixed-size at $N{=}64$ slots; we did not study tasks long enough for the slots to be exhausted. The port-Hamiltonian decomposition assumes a smooth ambient state space; rigid-body contacts induce non-smooth jumps that we currently absorb into v_{contact} rather than model with hybrid dynamics (Goebel et al., 2012). The RoboCasa slow-loop score combines a demonstration-progress surrogate, mean predicted contact probability, and an action-square penalty; a fully model-predictive variant with tree search would likely improve long-horizon planning at the price of latency. We evaluated on tabletop manipulation; locomotion and bimanual whole-body tasks are out of scope.

Broader impact. Generalizable manipulation policies may eventually be deployed in homes, hospitals, and warehouses. In the evaluated setup, CL-WAM updates its belief at each policy call and refreshes its slow plan every 24 executed actions. These mechanisms expose intermediate states that can be inspected, but this work does not evaluate safety monitoring or provide a certification guarantee. Deployment in human environments requires additional evaluation of failure modes, contact safety, and recovery.

B Detailed Control-Theoretic Readings of the Four Pillars

Baselines. Unsuffixed DiT4DiT denotes the official benchmark-specific released checkpoint. DiT4DiT (matched) is our protocol-matched three-seed retraining, $+58$M is its capacity control, and DiT4DiT (paper) is the original RoboCasa report. These are distinct measurements. The G1 distribution shifts were absent from task-specific G1 fine-tuning demonstrations; absence from simulation pretraining is unverified. Further model and data details are in Appendix E.

We expand the one-paragraph summary in §3.3 into four pillar-by-pillar discussions.

Belief: learned innovation correction. Classical Kalman filtering motivates innovation-weighted updates: under linear-Gaussian assumptions, its gain is determined by predictive and observation uncertainty (Kalman, 1960). Our low-rank gate K_θ is instead learned through $\mathcal{L}_{\text{belief}}$, which supervises the corrected belief against an offline smoother target with access to K future observations. This objective constrains the resulting belief, not the gain matrix uniquely or a calibrated posterior covariance. On held-out RoboCasa-GR1 trajectories, the measured correction $\|K_\theta r_t\|_F$ averages 0.183 in contact and 0.041 in free flight (App. H); the difference may arise from the gate, the residual, or both.

Memory: information that the rollout would have to re-derive. Slot memory stores facts that are stable over many steps (which object is grasped; which sub-goal is complete) but expensive to re-derive from raw frames. Without it, every rollout step must integrate the same long-horizon evidence from the visible context. This is why memory matters most under *distribution shift*: when the model is unsure about a frame, the cumulative evidence stored in M_t provides a low-variance estimate of which task-state we are in. The InfoNCE retrieval loss is the explicit information-theoretic statement of this property: it lower-bounds the mutual information between $\bar{m}_t$ and the future event embedding (van den Oord et al., 2018).

Port-Hamiltonian: conditional energy balance. Proposition 1 gives an exact energy accounting identity for an increasing pH integration coordinate with fixed conditioning. The implemented diagonal floor yields $R_\theta(c_t) \succeq 10^{-4} I$, so the isolated autonomous $(J_\theta - R_\theta)\nabla H_\theta$ term dissipates the learned latent energy. For the complete field, however, the control port, contact term, and correction head contribute the additional power terms in Eq. (6). In particular, the correction head has no hard norm bound. The identity therefore supplies a structural interpretation of the latent dynamics, not an exponential envelope for a 24-step robot rollout or a certified physical-energy bound.

Closed-loop training: reducing train–deployment covariate shift. Our rollout-mixing schedule is motivated by the central observation behind DAgger (Ross et al., 2011): a policy should be exposed during training to states resembling those induced by its own predictions at deployment. In CL-WAM, the retention probability $\rho_{u,k} = \rho_0 \gamma(u)^{k-1}$ progressively exposes the dynamics and belief update to self-generated visual latents as training progresses and rollout depth increases. This reduces the train–deployment distribution mismatch. Because our method uses offline generative rollouts rather than DAgger's online dataset aggregation and no-regret update procedure, we do not claim that the original DAgger regret guarantee applies directly to our setting.

C Proof and Scope of Proposition 1

Implemented pH parameterization. Here $J_\theta = A_\theta - A_\theta^\top$ is skew-symmetric. Let c_t denote the conditioning representation supplied to the pH heads. The implemented dissipation matrix is

$$\begin{aligned} R_\theta(c_t) &= L_\theta D_\theta(c_t) L_\theta^\top + d_\theta(c_t) I, \\ D_\theta(c_t) &= \operatorname{diag}\big(\operatorname{softplus}(g_\theta(c_t))\big), \\ d_\theta(c_t) &= \operatorname{softplus}(r_\theta(c_t)) + 10^{-4}. \end{aligned} \tag{15}$$

The low-rank factor has width 32, but the positive diagonal term makes the full R_θ positive definite. The separate 10^{-3} diagonal floor in the convexity parameterization of H_θ is not part of R_θ.

The token and candidate-scoring paths call the same pH field and share the context-gated additive control term

$$v_{\text{ctrl},\theta}(a, c) = W_G[\text{MLP}(a) \odot \text{sigmoid}(W_g \ \text{ContextMLP}(c))] \,. \tag{16}$$

This output is added directly to the structured field; no independent $G_\theta(x)$ multiplication follows it. During candidate scoring, c contains the current simulated belief and is recomputed at each planning step. Neither path projects the belief before pH integration. The token path uses zero action in training and the last actually executed action at deployment (zero after reset); demonstration actions enter the separately computed pH training loss. The scorer instead supplies each candidate's hypothetical action at each simulated step. The contact set $\mathcal{C}_t = \{(\hat{n}_i, \hat{\lambda}_i)\}_{i=1}^{N_c}$ contains predicted contact quantities. The reported LIBERO and RoboCasa runs use the weak prior in Eq. (27); no force

or wrench target supervises this branch. The correction head output is scaled by 0.1 and receives the soft penalty $0.01\ \mathrm{mean}(v_{\mathrm{corr}}^2)$; neither operation imposes a hard norm bound on v_{corr}. Skew symmetry and positive dissipation follow from the parameterization, not from an auxiliary loss.

Proof. Fix c and a during one continuous forward pH integration. Let $p_\sigma = \nabla_x H_\theta(x_\sigma^{\mathrm{pH}}; c)$. The chain rule and Eq. (5) give

$$\begin{aligned}\frac{d}{d\sigma} H_\theta(x_\sigma^{\mathrm{pH}}; c) &= p_\sigma^\top v_\theta(x_\sigma^{\mathrm{pH}}; c, a) \\ &= p_\sigma^\top J_\theta p_\sigma - p_\sigma^\top R_\theta(c) p_\sigma + p_\sigma^\top v_{\mathrm{ctrl},\theta}(a, c) + p_\sigma^\top \big(v_{\mathrm{contact}} + v_{\mathrm{corr}}\big) \\ &= -p_\sigma^\top R_\theta(c) p_\sigma + p_\sigma^\top v_{\mathrm{ctrl},\theta}(a, c) + p_\sigma^\top \big(v_{\mathrm{contact}} + v_{\mathrm{corr}}\big), \end{aligned} \tag{17}$$

because $p_\sigma^\top J_\theta p_\sigma = 0$ for $J_\theta^\top = -J_\theta$. This proves Eq. (6). Moreover, Eq. (15) implies

$$p_\sigma^\top R_\theta(c_t) p_\sigma = \Big\| D_\theta(c_t)^{1/2} L_\theta^\top p_\sigma \Big\|_2^2 + d_\theta(c_t) \|p_\sigma\|_2^2 \geq 10^{-4} \|p_\sigma\|_2^2.$$

For the isolated autonomous structural term, omit $v_{\mathrm{ctrl},\theta}$, v_{contact}, and v_{corr}; the preceding bound then gives the stated dissipation inequality. □

Scope. The positive diagonal floor in Eq. (15) means that $R_\theta \equiv 0$ is not a member of the implemented parameterization; a pure-Hamiltonian flow is therefore not an exact special case of the frozen model. Equation (6) assumes fixed conditioning and concerns the continuous forward pH integration coordinate. If $c = c_\sigma$ varies differentiably and H_θ depends on it, the chain rule instead gives

$$\frac{d}{d\sigma} H_\theta(x_\sigma; c_\sigma) = \nabla_x H_\theta^\top \dot{x}_\sigma + (\partial_c H_\theta)^\top \dot{c}_\sigma.$$

Candidate scoring recomputes its context between discrete Euler updates, so Eq. (6) is not an energy guarantee for the complete candidate rollout. The control, contact, and correction terms have no assumed joint hard bound, so the identity does not imply an exponential bound for the complete vector field. Nor does continuous-time dissipation by itself imply monotonicity of the deployed Euler updates. The pH coordinate is distinct from the video/action flow-matching noise times and from physical robot time.

D Two-Rate Inference Algorithm and Training Step

D.1 Expanded closed-loop rollout schedule and loss paths

Teacher-forced dynamics are exposed only to ground-truth observations, whereas deployment requires the model to continue from its own predictions. We therefore mix real and self-generated visual latents during multi-step rollout training.

Crucially, the schedule operates on two distinct axes: the optimizer update index u controls the decay factor γ, while the future rollout index k controls how strongly self-generated inputs are introduced within a rollout. Let U denote the total number of training updates and

$$p(u) = \mathrm{clip}\Big(\frac{u}{U}, 0, 1\Big), \qquad \gamma(u) = 0.60 + \frac{0.99 - 0.60}{2}\big[1 + \cos\big(\pi p(u)\big)\big]. \tag{18}$$

For future step $k \in \{1, \ldots, K_{\mathrm{roll}}\}$, with $K_{\mathrm{roll}} = 24$, the probability of retaining the real visual latent is

$$\rho_{u,k} = \rho_0\, \gamma(u)^{k-1}, \qquad b_{u,k} \sim \mathrm{Bernoulli}(\rho_{u,k}), \qquad \rho_0 = 1. \tag{19}$$

The mixed latent consumed at future step $t + k$ is

$$\tilde{z}_{t+k} = b_{u,k}\, z_{t+k}^{\mathrm{real}} + (1 - b_{u,k})\, \hat{z}_{t+k}, \tag{20}$$

Here z_{t+k}^{real} is the temporally aligned, spatially pooled feature of the frozen visual VAE. The prediction $\hat{z}_{t+k} = \mathrm{VisPred}(\hat{s}_{t+k|t+k-1}^{\mathrm{mix}})$ is formed before observation correction. The same Bernoulli draw mixes the corresponding visual features, object slots, and centres, while proprioceptive and semantic features remain real. The resulting mixed observation corrects the belief and updates memory.

The exponent is $k-1$ because the first future observation corresponds to rollout index zero in the implementation; hence $\rho_{u,1}=1$ for every training update.

Early in training, $\gamma(u)$ is close to 0.99, so real observations remain frequent even at long rollout depths. As training progresses, $\gamma(u)$ approaches 0.60, causing later future steps to rely increasingly on self-generated inputs while the first future step remains real. The belief update and memory write operate on the resulting mixed representation, exposing both modules to the partially self-generated inputs encountered during rollout.

Two losses supervise separate rollout paths. On the scheduled-mixing path, the current corrected belief and memory read produce a pre-correction prior and visual prediction,

$$\hat{s}^{\mathrm{mix}}_{t+k|t+k-1}=\mathrm{BeliefPred}\big(s^{\mathrm{mix}}_{t+k-1},a_{t+k-1},\bar{m}^{\mathrm{mix}}_{t+k-1},p^{\mathrm{contact}}_{t+k-1}\big)\,, \tag{21}$$

$$\hat{z}^{\mathrm{mix}}_{t+k|t+k-1}=\mathrm{VisPred}\Big(\hat{s}^{\mathrm{mix}}_{t+k|t+k-1}\Big)\,. \tag{22}$$

The Bernoulli-mixed visual features, slots, and centres are then combined with real proprioceptive and semantic features to correct the belief and update memory. A second path starts from the same initial belief and memory but recursively uses generated observations without observation correction. Writing z^{pool}_{t+k} for the temporally aligned, spatially pooled frozen-VAE target, the implemented losses are

$$\mathcal{L}_{\mathrm{CL}}=\frac{1}{K_{\mathrm{valid}}}\sum_{k=1}^{K_{\mathrm{valid}}}\mathrm{MSE}\Big(\hat{z}^{\mathrm{mix}}_{t+k|t+k-1},\mathrm{sg}\Big[z^{\mathrm{pool}}_{t+k}\Big]\Big)\,, \tag{23}$$

$$\mathcal{L}_{\mathrm{roll}}=\frac{1}{K_{\mathrm{valid}}}\sum_{k=1}^{K_{\mathrm{valid}}}\mathrm{MSE}\Big(\hat{z}^{\mathrm{free}}_{t+k|t+k-1},\mathrm{sg}\Big[z^{\mathrm{pool}}_{t+k}\Big]\Big)+0.1\,\mathcal{L}_{\mathrm{progress}}. \tag{24}$$

Here $K_{\mathrm{valid}}=24$ for the formal 25-observation training segment. $\mathrm{BeliefPred}$ denotes the belief-prior update represented by f_θ in Eq. (1), with memory-read and contact conditioning; $\mathrm{VisPred}$ is the separate implemented observation model. $\mathcal{L}_{\mathrm{progress}}$ is the mean smooth-L_1 task-cost loss against the demonstration remaining-progress target decreasing from 1 to 0.

D.2 Implemented auxiliary losses

The formal training segment contains $K_{\mathrm{valid}}=24$ observation transitions. Each MSE uses an elementwise mean reduction over its input tensors; losses accumulated per transition are then averaged over K_{valid}. Let x_k be the corrected belief, r_k its memory read, $c_k=[x_k;r_k]$ the pH context, and a_k the demonstration action supplied at that transition. Independently sample $\sigma_k\sim\mathcal{U}[0,1]$ and $\xi_k\sim\mathcal{N}(0,I)$, and define $x_{k,\sigma}=(1-\sigma_k)x_k+\sigma_k\xi_k$ and $g_k=\nabla_x H_\theta(x_{k,\sigma};c_k)$. The implemented pH loss is

$$\begin{aligned}\ell_{\mathrm{pH},k}=\;&\mathrm{MSE}(v_\theta(x_{k,\sigma};c_k,a_k),\,\xi_k-x_k)\\&+0.01\,\mathrm{mean}\Big[\mathrm{ReLU}\big(g_k^\top(J_{\theta,k}-R_{\theta,k})g_k\big)^2\Big]+0.01\,\mathrm{mean}\big[v^2_{\mathrm{corr},k}\big]\,,\end{aligned} \tag{25}$$

$$\mathcal{L}_{\mathrm{pH}}=\frac{1}{K_{\mathrm{valid}}}\sum_{k=1}^{K_{\mathrm{valid}}}\ell_{\mathrm{pH},k}. \tag{26}$$

The first term is flow matching between the corrected belief and independently sampled Gaussian noise, not supervised prediction of the next measured state. Because c_k contains x_k and ξ_k is sampled independently of a_k, the target ξ_k-x_k alone does not identify the change in future belief caused by substituting another action. Candidate propagation is therefore used for internal scoring. The positive autonomous-power term is computed in the implementation but vanishes in exact arithmetic under the parameterized skew symmetry of J_θ and positive semidefiniteness of R_θ.

In the reported LIBERO and RoboCasa configurations, the contact loss uses a weak probability prior:

$$\mathcal{L}_{\mathrm{contact}}=\frac{0.01}{BK_{\mathrm{valid}}N_c}\sum_{b=1}^{B}\sum_{k=1}^{K_{\mathrm{valid}}}\sum_{j=1}^{N_c}p^{\mathrm{contact}}_{b,k,j},\qquad N_c=8, \tag{27}$$

where B is the batch size. No force or wrench labels supervise this loss in these experiments. The implementation also contains a force-supervised branch, but that branch is inactive in the reported configurations.

For each transition with a real observation, let $\hat{C}_k, \hat{S}_k, \hat{e}_k, \hat{z}_k$ be the slot centres, slot features, observation embedding, and pooled visual latent predicted from the pre-correction belief prior. Let $C_k, S_k, e_k, z_k^{\text{pool}}$ be targets encoded from the real observation; the SlotAttention targets C_k, S_k are stop-gradient pseudo-targets. Write $D(C)_{ij} = \|C_i - C_j\|_2$. Then

$$\mathcal{L}_{\text{geo}} = \frac{1}{K_{\text{valid}}} \sum_{k=1}^{K_{\text{valid}}} \Big[\text{MSE}(\hat{C}_k, \text{sg}[C_k]) + 0.1\,\text{MSE}\Big(\text{sort}\, D(\hat{C}_k), \text{sg}[\text{sort}\, D(C_k)]\Big) + 0.1\,\text{Chamfer}_{\text{bidir}}(\hat{S}_k, \text{sg}[S_k])\Big], \tag{28}$$

$$\mathcal{L}_{\text{obs}} = \frac{1}{K_{\text{valid}}} \sum_{k=1}^{K_{\text{valid}}} \Big[\text{MSE}(\hat{e}_k, \text{sg}[e_k]) + \text{MSE}(\hat{z}_k, \text{sg}\Big[z_k^{\text{pool}}\Big])\Big]. \tag{29}$$

The bidirectional Chamfer distance averages the two directional nearest-neighbour squared-Euclidean distances. The geometric targets come from the visual encoder, not annotated object positions. Here z_k^{pool} is the temporally aligned, spatially pooled Cosmos VAE latent, not an RGB image or an unpooled latent map. On the scheduled-mixing path, $\hat{z}_k = \hat{z}^{\text{mix}}_{t+k|t+k-1}$; hence the visual component of $\mathcal{L}_{\text{obs}}$ equals $\mathcal{L}_{\text{CL}}$. Their effective coefficient in Eq. (12) is therefore $\lambda_6 + \lambda_8 = 0.6$, while the observation-embedding MSE has coefficient $\lambda_8 = 0.1$.

D.3 Expanded RoboCasa-GR1 candidate-scoring implementation

Both paths start directly from the corrected belief mean and call the same pH field. The conditioning-token path uses the preceding executed action and four Euler substeps before token projection; candidate scoring instead copies that belief for each candidate and uses its hypothetical actions, with one Euler substep per simulated planning step. We describe the frozen RoboCasa-GR1 deployment configuration. At each slow refresh, the action head uses 32 flow-integration steps to sample $N = 8$ candidate action chunks $\{A_j^{(n)}\}_{n=1}^{N}$. For scoring, each chunk is repeated or truncated to $H = 24$ actions, $\bar{a}_{1:24}^{(n)} = \text{RepeatOrTruncate}_{24}(A_j^{(n)})$. Starting from the corrected belief $b_0^{(n)} = s_t$, the scorer reads the current memory and advances a simulated belief once per planning step:

$$\begin{aligned} r_k^{(n)} &= \text{Read}(M_t, b_{k-1}^{(n)}), \\ q_k^{(n)} &= \nabla_b H_\theta(b_{k-1}^{(n)}), \\ b_k^{(n)} &= b_{k-1}^{(n)} + v_\theta\Big(b_{k-1}^{(n)}, [b_{k-1}^{(n)}; r_k^{(n)}], \bar{a}_k^{(n)}\Big), \qquad k = 1, \ldots, 24. \end{aligned} \tag{30}$$

Here $q_k^{(n)}$ is the energy gradient at the *pre-update* simulated state; the third argument of the pH operator is the candidate action at that step. Each update in Eq. (30) uses one pH Euler substep, independent of the action head's 32 flow-sampling steps.

The deployed scalar score is

$$\begin{aligned} J_n = &\frac{1}{24} \sum_{k=1}^{24} \text{softplus}\, f_{\text{task}}\Big([b_k^{(n)}; r_k^{(n)}]\Big) \\ &+ \frac{0.2}{24} \sum_{k=1}^{24} \left[\text{ReLU}\Big((q_k^{(n)})^\top (J_{\theta,k}^{(n)} - R_{\theta,k}^{(n)}) q_k^{(n)}\Big) + \frac{1}{8} \sum_{m=1}^{8} p_{k,m}^{\text{contact},(n)} \right] \\ &+ \frac{0.01}{24D} \sum_{k=1}^{24} \Big\| \bar{a}_k^{(n)} \Big\|_2^2, \end{aligned} \tag{31}$$

where D is the action dimension and $J_{\theta,k}^{(n)}, R_{\theta,k}^{(n)}$ are evaluated at the pre-update scoring state. Thus the task and auxiliary terms are each averaged over 24 steps, while the action term is the mean

Algorithm 1 One CL-WAM training rollout with scheduled mixing and free recursion.

Require: Observations $\{o_{t+k}\}_{k=0}^{24}$, demonstration actions $\{a_{t+k}\}_{k=0}^{23}$, optimizer update u, total updates U, initial corrected belief s_t and memory M_t.

Set $K_{\text{valid}} \leftarrow 24$, $\rho_0 \leftarrow 1$, $\gamma_{\max} \leftarrow 0.99$, $\gamma_{\min} \leftarrow 0.60$.
$p \leftarrow \text{clip}(u/U, 0, 1)$; $\gamma \leftarrow \gamma_{\min} + \frac{\gamma_{\max}-\gamma_{\min}}{2}[1 + \cos(\pi p)]$.
$(s^{\text{mix}}, M^{\text{mix}}) \leftarrow (s_t, M_t)$; $(s^{\text{free}}, M^{\text{free}}) \leftarrow (s_t, M_t)$.
$\mathcal{E}_{\text{CL}}, \mathcal{E}_{\text{free}}, \mathcal{E}_{\text{progress}} \leftarrow 0$.
Starting from the corrected belief mean s_t, integrate the shared pH field for four Euler substeps with zero action; apply PHToken_θ to the integrated state for action-head conditioning.
for $k = 1, \ldots, K_{\text{valid}}$ **do**
  Encode o_{t+k} and extract its temporally aligned, spatially pooled frozen-VAE target z_{t+k}^{pool}.
  $\bar{m}_{t+k-1}^{\text{mix}} \leftarrow \text{Read}(M^{\text{mix}}, s^{\text{mix}})$; obtain $p_{t+k-1}^{\text{contact}}$ from the contact head.
  $\hat{s}_{t+k|t+k-1}^{\text{mix}} \leftarrow \text{BeliefPred}(s^{\text{mix}}, a_{t+k-1}, \bar{m}_{t+k-1}^{\text{mix}}, p_{t+k-1}^{\text{contact}})$.
  $\hat{z}_{t+k|t+k-1}^{\text{mix}} \leftarrow \text{VisPred}(\hat{s}_{t+k|t+k-1}^{\text{mix}})$.
  $\mathcal{E}_{\text{CL}} \leftarrow \mathcal{E}_{\text{CL}} + \text{MSE}(\hat{z}_{t+k|t+k-1}^{\text{mix}}, \text{sg}\left[z_{t+k}^{\text{pool}}\right])$. // before observation correction
  $\rho_{u,k} \leftarrow \rho_0 \gamma^{k-1}$; sample $b_{u,k} \sim \text{Bernoulli}(\rho_{u,k})$.
  Form $\tilde{e}_{t+k}$ by mixing real and predicted visual features, object slots, and centres using $b_{u,k}$; retain real proprioceptive and semantic features.
  $s^{\text{mix}} \leftarrow \hat{s}_{t+k|t+k-1}^{\text{mix}} + K_\theta(\hat{s}_{t+k|t+k-1}^{\text{mix}}, M^{\text{mix}})[\tilde{e}_{t+k} - h_\theta(\hat{s}_{t+k|t+k-1}^{\text{mix}})]$.
  $M^{\text{mix}} \leftarrow \text{Mem}_\theta(M^{\text{mix}}, \tilde{e}_{t+k}, s^{\text{mix}})$.
  Advance the independent free branch from $(s^{\text{free}}, M^{\text{free}})$ using generated observations without observation correction; obtain $\hat{z}_{t+k|t+k-1}^{\text{free}}$.
  $\mathcal{E}_{\text{free}} \leftarrow \mathcal{E}_{\text{free}} + \text{MSE}(\hat{z}_{t+k|t+k-1}^{\text{free}}, \text{sg}\left[z_{t+k}^{\text{pool}}\right])$.
  Add the task-cost head's smooth-L_1 error against the demonstration remaining-progress target to $\mathcal{E}_{\text{progress}}$.
  From the pre-correction belief prior and stopped real-observation targets, accumulate the geometry and observation losses in Appendix D.2.
  From the corrected belief, its memory-read context, and the demonstration action for this transition, independently sample the pH interpolation coordinate and accumulate $\ell_{\text{pH},k}$ defined in Appendix D.2; also accumulate the weak contact prior, belief loss, and memory loss.
end for
$\mathcal{L}_{\text{CL}} \leftarrow \mathcal{E}_{\text{CL}}/K_{\text{valid}}$; $\mathcal{L}_{\text{progress}} \leftarrow \mathcal{E}_{\text{progress}}/K_{\text{valid}}$.
$\mathcal{L}_{\text{roll}} \leftarrow \mathcal{E}_{\text{free}}/K_{\text{valid}} + 0.1\,\mathcal{L}_{\text{progress}}$.
Form the total objective $\mathcal{L}$ according to Eq. (12); return $\nabla_\theta \mathcal{L}$.

squared action over steps and dimensions. The task-cost head f_{task} is supervised by a demonstration remaining-progress target that decreases from 1 to 0; it does not predict a candidate's realized success probability or return.

The implementation computes the rectified autonomous-power term in Eq. (13). Because J_θ is skew-symmetric and $R_\theta \succeq 0$, however, $q^\top(J_\theta - R_\theta)q = -q^\top R_\theta q \leq 0$, so its ReLU is zero in exact arithmetic. We therefore do not interpret that term as an effective physical-violation detector. The other auxiliary component is the mean *predicted contact probability*, not a measured contact violation.

E Evaluation Protocol and Baseline Provenance

Benchmarks. LIBERO (Liu et al., 2023) tests a Franka Emika Panda across four suites (Spatial, Object, Goal, Long), with 10 tasks per suite. RoboCasa-GR1 (Nasiriany et al., 2024; Bjorck et al., 2025) is a 24-task tabletop suite on the Fourier GR1 humanoid (29-DoF, ego camera, 1,000 demonstrations per task, 720-step horizon). For the controlled RoboCasa-GR1 comparison, CL-WAM, DiT4DiT (matched), and DiT4DiT +58M are each trained with three independent random seeds and evaluated with 50 rollouts per task and seed, yielding $3 \times 24 \times 50 = 3{,}600$ evaluation episodes per method. The official DiT4DiT release is represented by its published Run 1 of a single

Algorithm 2 One RoboCasa-GR1 deployment policy call: observation correction before optional replanning.

Require: Trained model θ; current observation o_t; previous belief s_{t-1}, memory M_{t-1}, and last executed action a_{t-1} of the preceding chunk ($a_{t-1} = 0$ immediately after reset); retained plan token η_{j-1}; policy-call index j, starting at 0.

Set $P \leftarrow 2$, $L \leftarrow 12$, $H \leftarrow 24$, and $N \leftarrow 8$.
$e_t \leftarrow E(o_t)$.
$\hat{s}_{t|t-1} \leftarrow s_{t-1} + \Delta t\, f_\theta(s_{t-1}, a_{t-1}, M_{t-1})$.
$s_t \leftarrow \hat{s}_{t|t-1} + K_\theta(\hat{s}_{t|t-1}, M_{t-1})\big(e_t - h_\theta(\hat{s}_{t|t-1})\big)$.
$M_t \leftarrow \mathrm{Mem}_\theta(M_{t-1}, e_t, s_t)$.
Set $x_0^{\mathrm{pH}} \leftarrow s_t$; read the current memory context, integrate the shared pH field for four Euler substeps using a_{t-1}, and set the conditioning token to $\mathrm{PHToken}_\theta(x_1^{\mathrm{pH}})$.
if $j = 0$ **or** $j \bmod P = 0$ **then**
    Sample N candidate action chunks $\{A_j^{(n)}\}_{n=1}^N$ with 32 action-head flow-integration steps.
    for $n = 1, \ldots, N$ **do**
        $\bar{a}_{1:24}^{(n)} \leftarrow \mathrm{RepeatOrTruncate}_{24}(A_j^{(n)})$.
        Set $b_0^{(n)} \leftarrow s_t$; run the 24 memory reads and one-substep pH belief updates in Eq. (30).
        Compute J_n using Eq. (13).
    end for
    $n_j^\star \leftarrow \arg\min_{1 \le n \le N} J_n$.
    $\eta_j \leftarrow \mathrm{sigmoid}(\gamma_\theta)\ \mathrm{Linear}_\theta(\bar{a}_{1:12}^{(n_j^\star)})$.
else
    $\eta_j \leftarrow \eta_{j-1}$.
end if
Draw fresh action noise $z_j^{\mathrm{fast}} \sim \mathcal{N}(0, I)$, independently of the selected candidate.
Sample executable chunk $a_{t:t+L-1}$ from z_j^{fast} with the four-step action head, conditioned on the current belief, memory, pH token, and plan token η_j.
Return $(a_{t:t+L-1}, s_t, M_t, \eta_j)$.

released checkpoint, with 50 rollouts per task and 1,200 episodes across the same 24 tasks. The real-world suite uses a Unitree G1 humanoid with 16-DoF dual arms and ALOHA 2 grippers on seven household tasks (Pick & Place, Stack Cup, Arrange Flower, Insert Plate, Box Packing, Move Spoon, Drawer Interaction); we collect 200 teleoperated demonstrations per task and evaluate over 20 rollouts each. Figure 3 visualises the real-world suite; every frame is the policy's only visual input, captured by the robot's Intel RealSense D435i head camera.

Baselines. We compare against the GR00T series (Bjorck et al., 2025; Llontop et al., 2026) and a parameter-matched Qwen3-DiT (Bai et al., 2025) baseline. Unless explicitly qualified, *DiT4DiT* denotes the official benchmark-specific released checkpoint: its published Run 1 is the headline RoboCasa-GR1 reference. *DiT4DiT (matched)* denotes our local three-seed retraining under the same data, optimization, and evaluation protocol as CL-WAM; it supports controlled comparisons. *DiT4DiT* +58*M* is a locally trained capacity control that adds approximately the same number of trainable parameters as CL-WAM without its proposed closed-loop components. *DiT4DiT (paper)* denotes the 50.8% RoboCasa-GR1 result in the original paper and is included for historical comparison. The official RoboCasa release, the original paper result, and the matched retraining are distinct measurements and are identified separately in every result table.

For LIBERO-Long, we additionally include LaDi-WM and MemoryVLA. For the real-world G1 experiments, CL-WAM is pre-trained on a subset of the simulated GR1 dataset (241,450 episodes, 15% of the scale used by GR00T-N1.5) and then fine-tuned on the task-specific G1 demonstrations. The distribution-shift conditions below are held out from those G1 fine-tuning demonstrations; this held-out split does not establish absence from the earlier simulated pretraining corpus. CL-WAM adds 58 M trainable parameters over DiT4DiT's 469 M trainable parameters and is trained for 200 k steps on 32×A100 GPUs.

Main-table reporting conventions. For Table 1, best and second-best values among the listed methods are shaded dark and light, respectively, with ties sharing shading. The three independently trained seeds for CL-WAM and DiT4DiT (matched) each have 50 rollouts per task, or 500 per suite and seed. Their reported uncertainty is the standard error across seeds; for Avg, the four suite rates are first averaged within each seed, and the mean and standard error are then computed over those three seed averages. All other LIBERO rows are published results from their cited sources and may use different evaluation budgets. For Table 2, official DiT4DiT Run 1 is one checkpoint with 50 rollouts per RoboCasa task ($1{,}200$ trials); the locally controlled RoboCasa rows use three independently trained seeds ($3{,}600$ trials each). The Unitree G1 local rows use 20 binary trials per task, yielding $101/140$, $103/140$, and $109/140$ successes for DiT4DiT (matched), DiT4DiT $+58$M, and CL-WAM, respectively. Dashes indicate unavailable results; shading is applied to reported values, with ties shared.

Figure conventions and architecture setting. The seven G1 tasks range from precision placement (Stack Cup, Insert Plate, Arrange Flower) to multi-stage execution (Box Packing, Drawer Interaction). Figure 3 shows first-person ego-camera frames; ALOHA 2 parallel-jaw grippers are the only end-effectors. Figure 5 shows that success peaks at video-DiT layer 18, that one integration step gives the best hidden-feature extraction result, and that joint training increases the temporal-feature silhouette score from 0.09 to 0.17. These measured settings are used in the final model.

F Hyperparameters and Implementation Details

Parameter budget per pillar. Excluding the frozen video backbone, the trainable portion of CL-WAM contains 527M parameters, compared with 469M for DiT4DiT under the same accounting convention. The breakdown of the 58 M added over DiT4DiT is: belief encoder f_θ + Kalman gate K_θ + observation model h_θ (21 M, 36%); memory bank + read/write heads (14 M, 24%); port-Hamiltonian heads (J_θ, R_θ, H_θ, and the context-gated control map; 19 M, 33%); contact & correction heads (4 M, 7%). The shared trainable DiT4DiT stack contains 469 M parameters. For the RoboCasa capacity control, DiT4DiT +58M adds a 2048-dimensional unstructured conditioner comprising input and output projections, 11 residual blocks with LayerNorm and SiLU, and a bottleneck sized to match the WAM parameter budget. It maps mean-pooled frozen-backbone features to one gated token appended to the action DiT conditioning sequence during both training and inference. This module contains $58{,}045{,}256$ trainable parameters, compared with $58{,}044{,}413$ in the WAM module, a difference of 843 parameters (0.0015%).

Training breakdown. Total $\sim$48 hours on $32\times$A100-80G is split as: data loading + augmentation 11%; forward pass 42%; backward pass 39%; optimiser step + sync 8%. Belief and memory updates add 4.1% over DiT4DiT's forward pass; the port-Hamiltonian losses add 2.7%.

Inference breakdown (single A100, batch 1). Slow planning pathway. One slow-loop planning update takes 115 ms in CL-WAM versus 111 ms in DiT4DiT, comprising world-model rollout (87 ms), candidate scoring (18 ms), memory read (4 ms), and belief refresh (6 ms). The slow plan is refreshed on the first policy call and every $P{=}2$ policy calls thereafter on RoboCasa, where each policy call executes an action chunk of length $L{=}12$. On LIBERO, the plan is refreshed on the first call and every $P{=}3$ calls thereafter, with $L{=}8$ executed actions per call. Thus, both benchmark configurations refresh the slow plan every $P \times L = 24$ executed action steps, matching the planning horizon $H{=}24$.

Fast action pathway. One fast-path update takes 32 ms in CL-WAM versus 30 ms in DiT4DiT, comprising observation encoding (4 ms), belief prediction (3 ms), innovation correction (5 ms), memory read/write (4 ms), and 4-step flow integration for action generation (16 ms).

G Design Choices and Sensitivity Analysis

We document the key design choices that emerged from sensitivity sweeps, together with the cost of getting each one wrong.

Table 4: **CL-WAM hyperparameters.** Where DiT4DiT shares a value, we leave it identical (marked †) so that all gains are attributable to the four pillars rather than to lower-level tuning.

Module	Symbol	Value
Video DiT layers / hidden†	L_v/d_v	$24/1024$
Action DiT layers / hidden†	L_a/d_a	$12/768$
Trainable params (excl. frozen video backbone)	—	469 M / 527 M
Belief state dim	d_s	1024
Process-noise covariance rank	r_Q	16
Kalman gain rank	r_K	16
Smoother future window	K	8
Memory slots	N	64
Slot dim	d_m	256
Write gate bias init	b_β	2
Memory λ_{sparse} / λ_{stable}	—	$0.05/0.01$
PH state dim	d_x	1024
J_θ rank	r_J	32
R_θ low-rank factor width	r_R	32
R_θ diagonal floor	ϵ_R	10^{-4}
H_θ convexity diagonal floor	ϵ_H	10^{-3}
Control projection bottleneck	r_G	16
pH token Euler substeps	—	4
pH scorer substeps / plan step	—	1
pH positive-power / correction penalties	—	$0.01/0.01$
Bernoulli rollout depth max	K_{roll}	24
Initial retention factor	ρ_0	1.0
Training-time decay factor	$\gamma(u)$	cosine $0.99 \to 0.60$
Slow-loop denoising steps	—	32
Fast-loop denoising steps	—	4
Slow-loop horizon H / candidates N	—	$24/8$
Executed action chunk L	—	RoboCasa: 12; LIBERO: 8
Slow-plan refresh period P	—	RoboCasa: 2 calls; LIBERO: 3 calls
Total objective weights	$\lambda_{1\ldots8}$	$0.5, 0.2, 0.3, 0.5, 0.5, 0.5, 0.2, 0.1$
Optimiser†	AdamW	$\beta = (0.9, 0.95)$, wd 0.01
LR† / warm-up†	—	1×10^{-4}, 2 000 steps
Total steps† / batch size†	—	$200\,000/128$
Hardware†	—	32× A100-80G

Memory size N. We swept $N \in \{16, 32, 64, 128\}$ on RoboCasa-GR1; SR was 63.7, 66.8, **67.5**, and 67.7, respectively. Reducing the memory from 64 to 16 slots costs 3.8 points, whereas increasing it from 64 to 128 changes performance by only $+0.2$ points. We therefore use N=64 as a compact operating point.

Kalman gain rank r. We swept $r \in \{4, 8, 16, 32, \text{dense}\}$; SR was 64.5, 66.2, **67.5**, 67.4, 67.7. The low-rank gain at r=16 matches a dense gain (≤ 0.2 pt difference) at 20× fewer parameters, supporting our claim that the gain's effective rank is small.

Smoother window K. We swept $K \in \{2, 4, 8, 16, 24\}$ for $\mathcal{L}_{\text{belief}}$; SR was 62.7, 64.2, 67.5, 67.6, and 66.9, respectively. Performance improves strongly up to K=8, while larger windows provide no meaningful additional gain (67.6% at K=16 and 66.9% at K=24). Therefore, we use K=8 throughout.

Training-time decay-factor schedule $\gamma(u)$. The final model uses the rollout-depth retention probability $\rho_{u,k} = \rho_0\gamma(u)^{k-1}$ with $\rho_0 = 1$, where $\gamma(u)$ is cosine-annealed from 0.99 to 0.60 over optimizer updates. We compare this schedule against a constant-retention control with $\rho_{u,k} \equiv 0.9$. The annealed schedule improves RoboCasa-GR1 success from 66.8% to 67.5% (+0.7 pt).

Loss weight robustness. We perturbed each of $(\lambda_1, \ldots, \lambda_8)$ by $\pm 50\%$ and measured the resulting RoboCasa-GR1 SR. The largest observed drop was 1.2 points when under-weighting $\mathcal{L}_{\text{CL}}$ to 0.25, while the smallest change was 0.1 points when increasing $\mathcal{L}_{\text{geo}}$ to 0.3. These results indicate that the final performance is not highly sensitive to moderate perturbations of any single loss weight.

Slow-loop horizon H. We tested $H \in \{8, 16, 24, 36, 48\}$; SR was 65.2, 66.3, $\mathbf{67.5}$, 67.2, and 66.8, respectively. Performance improves as the planning horizon increases to $H{=}24$, after which longer horizons provide no further benefit and slightly reduce success. We therefore use $H{=}24$ in all main experiments.

H Per-Pillar Microscope: Where Does Each Pillar Help?

Belief: where the innovation correction is applied. We compute the per-step Frobenius norm of $K_\theta r_t$, where $r_t = e_t - h_\theta(\hat{s}_{t|t-1})$, on $1\,000$ held-out RoboCasa-GR1 trajectories and bucket steps into (a) free flight (no contact), (b) approach (object within 2 cm of gripper), and (c) contact. The mean correction magnitudes are 0.041, 0.106, and 0.183, respectively—more than four times larger in contact than in free flight. Because the measured correction depends on both K_θ and r_t, it does not isolate a change in the gate itself.

Memory: which slots specialise. We freeze a trained CL-WAM and probe the 64 memory slots on 24 randomly-chosen tasks. Using a logistic probe trained to predict "cup grasped", "microwave open", "object substituted", and "sub-goal completed", we find 9, 7, 4, and 13 slots respectively that achieve probe AUC > 0.9. Most slots (25 of 64) remain task-generic; they store low-level scene features. Memory specialisation emerges without any per-slot supervision.

Closed-loop rollout: error accumulation with rollout depth. We measure latent prediction MSE as a function of rollout depth k on the held-out RoboCasa milk subset. DiT4DiT (matched) accumulates error more rapidly as the rollout becomes longer, whereas CL-WAM maintains a consistently lower prediction error across the evaluated horizon. At the deployment horizon $k{=}24$, the latent prediction MSE is 0.0029 for DiT4DiT (matched) and 0.0017 for CL-WAM, corresponding to a 41% relative reduction. This result is consistent with the intended role of closed-loop rollout training in reducing compounding prediction error under self-generated inputs.

I Same-State Action Interventions

Protocol. We restore the same held-out RoboCasa-GR1 simulator snapshot separately for each of the eight action candidates produced by the deployed proposal code. For this diagnostic, each candidate is executed directly to observe its hypothetical consequence. Deployment instead converts the selected candidate into a plan token and samples executable actions from fresh noise in the fast head. The candidate-scoring path uses one pH Euler update per proposed action at each of 24 planning positions. We align planning position k with executed environment action k for evaluation; this convention does not calibrate pH integration time to physical time. All variants use the same initial snapshot, candidate set, and evaluation belief representation.

Prediction and action-effect measures. Let $\hat{b}_{gnk}$ and b^{obs}_{gnk} denote the predicted and observation-updated beliefs for initial-state group g, candidate n, and position k. With belief dimension d_b, the pairwise action-effect error and its zero-effect reference are

$$E_{\text{effect}}(k) = \frac{1}{|G|} \sum_{g \in G} \frac{1}{28 d_b} \sum_{n<m} \left\| (\hat{b}_{gnk} - \hat{b}_{gmk}) - (b^{\text{obs}}_{gnk} - b^{\text{obs}}_{gmk}) \right\|_2^2, \tag{32}$$

$$E_0(k) = \frac{1}{|G|} \sum_{g \in G} \frac{1}{28 d_b} \sum_{n<m} \left\| b^{\text{obs}}_{gnk} - b^{\text{obs}}_{gmk} \right\|_2^2. \tag{33}$$

A truly candidate-independent prediction has $E_{\text{effect}}(k) = E_0(k)$; its effect cosine is undefined because its predicted action-effect vector is zero. We report state and effect errors at $k \in \{1, 4, 8, 12, 24\}$.

J EXTENDED LIBERO AND ROBOCASA-GR1 TABLES

Additional four-suite interpretation. Table 1 reports the standard four-suite LIBERO benchmark under the official-aligned evaluation protocol. Among the additional published methods in the table, G0.5 reports the strongest four-suite average at 98.9%; the official DiT4DiT LIBERO checkpoint reports 98.6%. These published rows are external benchmark references, whereas the DiT4DiT (matched) is the controlled three-seed comparison. CL-WAM and DiT4DiT (matched) use identical downstream training and evaluation protocols. Across three independent training seeds, the four-suite averages are $99.15 \pm 0.14\%$ for CL-WAM and $98.65 \pm 0.20\%$ for DiT4DiT (matched), displayed to one decimal place as 99.2% and 98.7% in the table. The largest suite-level difference is on LIBERO-Long: $98.0 \pm 0.23\%$ versus $96.8 \pm 0.46\%$. Because LIBERO is close to saturation, the aggregate difference is small. Table 5 also reports the Long success counts alongside released comparison models.

Additional RoboCasa and G1 comparisons. Table 2 summarizes the 24-task RoboCasa-GR1 and seven-task real-world G1 results. On RoboCasa-GR1, CL-WAM succeeds on $2430/3600$ evaluation episodes ($\mathbf{67.50}\%$). Run 1 of the officially released DiT4DiT checkpoint succeeds on $680/1200$ episodes (56.7%) over the same 24 tasks, giving a descriptive benchmark-level difference of 10.8 percentage points. For a controlled comparison, DiT4DiT (matched) succeeds on $1982/3600$ episodes (55.06%), 12.44 points below CL-WAM, while DiT4DiT $+58$M obtains $1983/3600$ (55.08%). CL-WAM exceeds the official release on 18 of the 24 tasks and exceeds DiT4DiT (matched) on all 24; the complete task-level results appear in Appendix J.

On the seven-task Unitree G1 suite, CL-WAM succeeds in $109/140$ trials (77.9%), compared with $101/140$ (72.1%) for DiT4DiT (matched) and $103/140$ (73.6%) for the capacity-matched DiT4DiT $+58$M control. These are observed differences over 20 binary trials per task: CL-WAM records the same number of successes on Pick & Place, two more on Box Packing and Stack Cup, and one more on each of the other four tasks. Per-task results appear in Table 9 of Appendix M.

LIBERO-Long-10 comparison. Table 5 gives success counts for the matched Long comparison and includes released models as external references. CL-WAM succeeds on $1470/1500$ episodes ($\mathbf{98.00}\%$), compared with $1452/1500$ (96.80%) for DiT4DiT (matched), our primary protocol-matched comparator, corresponding to a $+1.20$ point absolute improvement. For external reference, the official LIBERO DiT4DiT checkpoint obtains $483/500$ (96.60%) on the same ten-task suite, closely matching the performance of our matched rerun. We additionally evaluate LaDi-WM and MemoryVLA, which obtain 63.00% and 91.60%, respectively. Despite operating in a high-success regime, CL-WAM reduces the number of failed episodes from 48 to 30 relative to DiT4DiT (matched). The aligned seed-level improvements are $+0.8$, $+1.2$, and $+1.6$ percentage points (mean $+1.20$; seed-level standard error 0.23 points).

Table 5: **Controlled evaluation on the ten LIBERO-Long tasks.** CL-WAM and DiT4DiT (matched) aggregate three independently trained seeds, with 500 evaluation episodes per seed (1,500 total). Released comparison models are evaluated over 500 episodes under the same ten-task suite.

Method	Success / Trials	Success Rate (%) ↑
LaDi-WM	$315/500$	63.00
MemoryVLA	$458/500$	91.60
DiT4DiT (Ma et al., 2026)	$483/500$	96.60
DiT4DiT (matched) (Ma et al., 2026)	$1452/1500$	96.80
CL-WAM (ours)	$\mathbf{1470/1500}$	**98.00**

We report the official 24 RoboCasa-GR1 tasks and all 40 LIBERO tasks in Tables 6 and 7. On RoboCasa-GR1, CL-WAM exceeds the official DiT4DiT release on 18 of 24 tasks and DiT4DiT (matched) on all 24 tasks. The corresponding aggregate rates are 67.50%, 56.7%, and 55.06%, respectively. The standard LIBERO task-level table compares CL-WAM with DiT4DiT (matched); the additional controlled LIBERO-Long-10 comparison appears in Table 5.

Table 6: **Full RoboCasa-GR1 results on the official 24-task tabletop suite (success rate %).** DiT4DiT (paper) is the original reported result; DiT4DiT is Run 1 of the official released checkpoint. These two columns each have 50 trials per task. DiT4DiT (matched) and CL-WAM each pool three independently trained seeds with 50 trials per task and seed (150 per task). Per-task values are rounded to one decimal place; averages are computed from raw success counts before rounding. The aggregate totals are $609/1200$, $680/1200$, $1982/3600$, and $2430/3600$ in column order. Best and second-best values in each row are shaded dark and light, respectively.

Task	DiT4DiT (paper)	DiT4DiT	DiT4DiT (matched)	**CL-WAM**
BottleToCabinetClose	48.0	50.0	40.0	54.0
CanToDrawerClose	74.0	80.0	78.0	89.3
CupToDrawerClose	52.0	50.0	52.7	66.0
MilkToMicrowaveClose	50.0	58.0	25.3	40.0
PotatoToMicrowaveClose	36.0	40.0	23.3	38.0
WineToCabinetClose	42.0	60.0	50.7	64.0
FromCuttingboardToBasket	52.0	54.0	68.0	78.7
FromCuttingboardToCardboardbox	48.0	50.0	76.0	87.3
FromCuttingboardToPan	76.0	80.0	80.7	91.3
FromCuttingboardToPot	62.0	52.0	84.7	95.3
FromCuttingboardToTieredbasket	50.0	44.0	59.3	72.0
FromPlacematToBasket	50.0	58.0	52.7	66.0
FromPlacematToBowl	56.0	64.0	74.0	85.3
FromPlacematToPlate	32.0	66.0	29.3	44.0
FromPlacematToTieredshelf	18.0	44.0	19.3	34.0
FromPlateToBowl	56.0	64.0	63.3	75.3
FromPlateToCardboardbox	58.0	50.0	61.3	74.0
FromPlateToPan	68.0	58.0	65.3	77.3
FromPlateToPlate	58.0	62.0	61.3	74.0
FromTrayToCardboardbox	38.0	52.0	34.0	48.0
FromTrayToPlate	56.0	64.0	65.3	77.3
FromTrayToPot	54.0	68.0	46.7	60.0
FromTrayToTieredbasket	46.0	50.0	38.0	52.0
FromTrayToTieredshelf	38.0	42.0	72.0	76.7
Average (24 tasks)	50.8	56.7	55.06	**67.50**

Table 7: **Full LIBERO (40 tasks across 4 suites, success rate %).** Across the 40 tasks, CL-WAM outperforms DiT4DiT (matched) on 25, matches it on 12, and is lower on 3 saturated Object-suite tasks.

	LIBERO-Spatial / Object		LIBERO-Goal / Long	
Task index	DiT4DiT (matched)	**CL-WAM**	DiT4DiT (matched)	**CL-WAM**
#1	99.3 / 100.0	100.0 / 100.0	100.0 / 98.0	100.0 / 99.3
#2	100.0 / 100.0	100.0 / 100.0	100.0 / 97.3	100.0 / 98.7
#3	99.3 / 100.0	100.0 / 100.0	99.3 / 97.3	100.0 / 98.7
#4	98.7 / 100.0	99.3 / 99.3	99.3 / 97.3	100.0 / 98.0
#5	98.7 / 100.0	99.3 / 100.0	99.3 / 96.7	100.0 / 98.0
#6	98.0 / 100.0	98.7 / 100.0	99.3 / 96.7	99.3 / 98.0
#7	98.0 / 100.0	98.7 / 99.3	99.3 / 96.7	99.3 / 97.3
#8	98.0 / 100.0	98.7 / 100.0	98.7 / 96.0	99.3 / 97.3
#9	98.0 / 100.0	98.7 / 100.0	98.7 / 96.0	99.3 / 97.3
#10	98.0 / 100.0	98.7 / 99.3	98.0 / 96.0	98.7 / 97.3
Avg	98.6 / **100.0**	**99.2** / 99.8	99.2 / 96.8	**99.6 / 98.0**

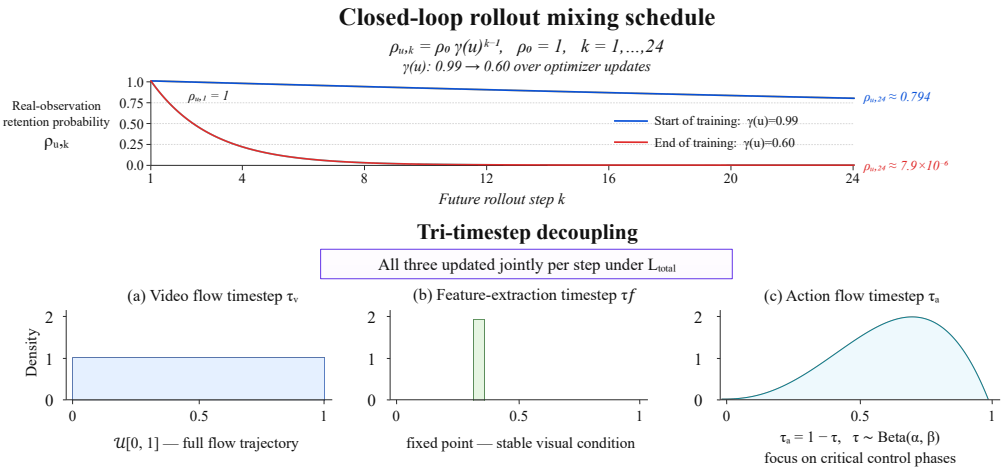


Figure 6: **Closed-loop rollout mixing schedule and tri-timestep decoupling.** Top: the closed-loop mixing schedule depends on both optimizer update u and future rollout depth k. At update u, the probability of retaining the real visual latent at future step k is $\rho_{u,k} = \rho_0\gamma(u)^{k-1}$, with $\rho_0 = 1$ and $k = 1, \ldots, 24$. The decay factor $\gamma(u)$ is cosine-annealed from 0.99 to 0.60 over optimizer updates; the two curves show the start and end of training. Thus $\rho_{u,1} = 1$ throughout training, whereas later rollout steps become increasingly self-generated as training progresses. Bottom: the three flow timesteps are decoupled. The video-flow timestep τ_v is sampled uniformly over $[0, 1]$; the feature-extraction timestep τ_f is fixed to provide a stable visual condition; and the action-flow timestep is defined as $\tau_a = 1 - \tau$ with $\tau \sim \mathrm{Beta}(\alpha, \beta)$. All three branches are optimized jointly under the total objective.

K Training Schedule Visualisation

Figure 6 visualises the two distinct axes of the closed-loop rollout mixing schedule. The optimizer-update index u controls the training-time decay factor $\gamma(u)$, which is cosine-annealed from 0.99 to 0.60 over training, whereas the future rollout index $k \in \{1, \ldots, K_{\mathrm{roll}}\}$ controls the real-observation retention probability $\rho_{u,k} = \rho_0\gamma(u)^{k-1}$ with $\rho_0 = 1$ and $K_{\mathrm{roll}} = 24$. Thus, annealing occurs along the training axis u, while the exponential decay occurs along the rollout-depth axis k. The same figure also visualises the tri-timestep decoupling used for video flow, hidden-feature extraction, and action flow. The complete CL-WAM architecture is shown in Figure 2 of the main text.

L LIBERO-Plus Perturbation Robustness

LIBERO-Plus (Fei et al., 2025) evaluates changes in camera viewpoint, robot initial state, language, illumination, background, sensor noise, and object layout. Table 8 places our results alongside independently reported world-action-model results. Published baseline values are taken from Being-H0.7 (Luo et al., 2026), Cosmos-Policy (Kim et al., 2026), ImageWAM (Zhang et al., 2026), and ABot-M0.5 (Chen et al., 2026); they are not paired reruns under our evaluation stack.

M Real-World and Task-Level Distribution-Shift Results

Held-out split and observed rates. We evaluate transfer to conditions held out from task-specific training. In simulation, task-specific training uses only the three BottleToDrawerClose, BottleToCabinetClose, and BottleToMicrowaveClose tasks; Can, Cup, Milk, and Wine are introduced only in this evaluation. On the G1, the tested category, object, and distractor-count changes are absent from the corresponding real-world fine-tuning demonstrations. Here "held out" refers to these task-specific training stages; we do not infer that the tested objects were absent from all earlier pretraining data.

Table 8: **LIBERO-Plus success rates (%, higher is better).** For CL-WAM, Overall pools all seven perturbation categories: $8{,}484/10{,}030 = 84.6\%$; the category rates are computed from their respective success and trial counts. Published baseline values are transcribed from the cited sources rather than rerun under our evaluation stack. The Being-H0.7 source reports its zero-shot overall result without a corresponding category breakdown. The two ImageWAM backbone sizes are listed separately.

Method	Camera	Robot	Lang.	Light	BG	Noise	Layout	Overall
Being-H0.7	–	–	–	–	–	–	–	82.1
Cosmos-Policy	75.8	63.3	81.7	96.5	88.9	92.7	82.2	82.2
ImageWAM (FLUX.2 4B)	80.8	50.3	91.4	98.1	85.5	93.8	80.5	83.1
ImageWAM (FLUX.2 9B)	79.8	58.7	95.2	96.1	91.2	93.3	83.1	85.2
ABot-M0.5	70.5	87.4	88.6	94.0	89.7	75.5	85.2	83.4
CL-WAM (ours)	**75.5**	**70.5**	**90.8**	**92.0**	**88.2**	**92.7**	**85.5**	84.6

Qualitative rollouts appear in Figure 4; quantitative object-shift and LIBERO-Plus results appear in Figure 8 and Appendix L, respectively.

In simulation, CL-WAM reaches 61.0%, 43.4%, and 40.4% on ToDrawerClose, ToCabinetClose, and ToMicrowaveClose, respectively, outperforming DiT4DiT (matched) by 6.5–9.9 points under object substitutions held out from the three task-specific Bottle training tasks.

Each real-world OOD condition comprises 20 binary trials. Across the four tested conditions, CL-WAM records 45% on category-shifted Stack Cup, 60% under quantity variation, 75% on category-shifted Arrange Flower, and 40% on object-shifted Box Packing. These observed rates exceed the DiT4DiT (matched) rates by one or two successful trials per condition (5–10 percentage points). The sample sizes support a descriptive comparison in these tested settings.

Figure 4 shows representative successful rollouts: metallic and clear-glass cups replace translucent plastic cups in the corresponding G1 fine-tuning demonstrations; a corn cob replaces the demonstration eggplant; and an extra red distractor appears in the four-cup scene. These are illustrations of the tested conditions, not evidence that the objects were absent from earlier simulation pretraining.

Table 9: **Real-world Unitree G1 full results (success rate %).** Each task is evaluated over 20 binary rollouts. The 7-task average is the arithmetic mean of the seven task-level success rates. Best value per row is shaded dark; second best is shaded light. Ties receive the same shading.

Task	GR00T-N1.5	Qwen3-DiT	DiT4DiT (matched)	DiT4DiT +58M	**CL-WAM**
Pick & Place	85	10	**100**	**100**	**100**
Box Packing	0	0	50	55	**60**
Stack Cup	25	5	60	65	**70**
Drawer Interaction	20	0	90	90	**95**
Arrange Flower	25	0	75	75	**80**
Move Spoon	15	5	45	45	**50**
Insert Plate	–	–	85	85	**90**
7-task avg	–	–	72.1	73.6	**77.9**

Figure 7 reports per-task real-world success rates on the seven-task Unitree G1 suite. Figure 8 reports observed success rates under simulated object substitution and the tested real-world distribution shifts. The corresponding ego-view photos and illustrative rollouts appear in Figures 3 and 4.

N Controlled Validation: Closed-Loop Primitives as the Structural Deficit

To test the hypothesis that the open-loop WAM plateau is structural rather than parametric, we run a controlled three-way comparison on RoboCasa-GR1. Three variants share the same Cosmos-Predict2.5-2B (NVIDIA, 2025) video backbone, the same action DiT, the same training data subset, the same number of optimisation steps, and the same hyperparameters.The first is a capacity-matched

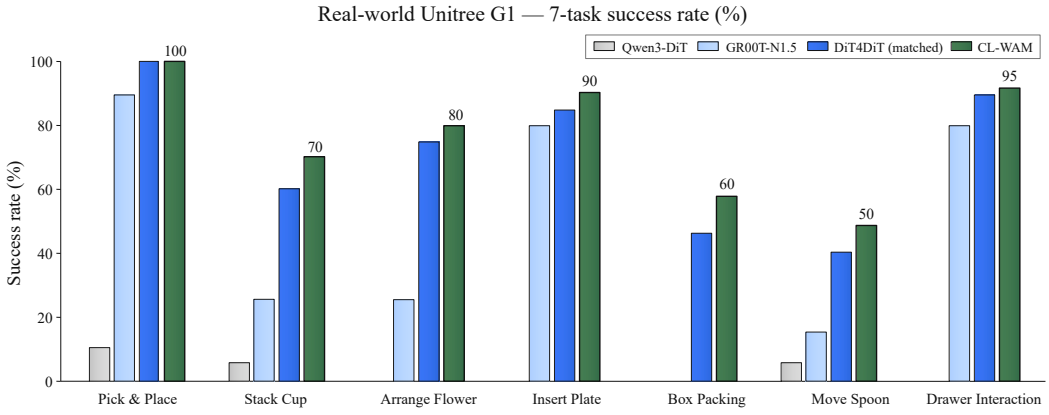


Figure 7: **Observed real-world success rates on Unitree G1.** Each task has 20 binary trials. A dash denotes an unreported method–task result (GR00T-N1.5 and Qwen3-DiT on Insert Plate); an absent bar at other positions denotes 0% success. Relative to DiT4DiT (matched), CL-WAM records the same number of successes on Pick & Place, two more on Box Packing and Stack Cup, and one more on each of the other four tasks.

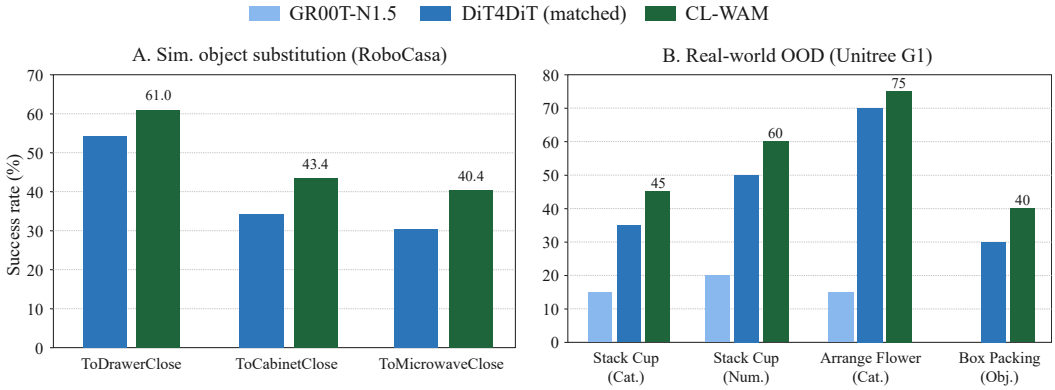


Figure 8: **Observed success under task-level distribution shifts.** Left: simulated substitutions held out from task-specific Bottle-task training, across three close-the-receptacle tasks. Right: four tested Unitree G1 distribution-shift conditions. Each real-world condition has 20 binary trials; its observed 5–10-point difference from DiT4DiT (matched) corresponds to one or two additional successes.

DiT4DiT baseline with an additional 58M trainable parameters but without the proposed closed-loop primitives. The second adds only the closed-loop rollout schedule of Section 3.3. The third is the full CL-WAM, which further adds belief, memory, and the port-Hamiltonian flow. The closed-loop schedule alone raises the 24-task average from 55.08% to 62.33% (+7.25 pt); adding belief, memory, and the port-Hamiltonian flow raises it to 67.50% (a further +5.17 pt). The latent-prediction-error curve in Figure 9 shows substantially slower error accumulation once closed-loop training is introduced, with the full CL-WAM producing the lowest error at long rollout depths. This empirical trend is consistent with the motivation from DAgger-style covariate-shift analyses, although our nonlinear generative setting does not directly instantiate the original DAgger assumptions.

O Additional Future-Frame Rollouts

To complement Figure 2, Figure 10 shows four selected future-frame sequences generated from one ego-view observation and a language goal. Across the displayed frames, the scenes retain

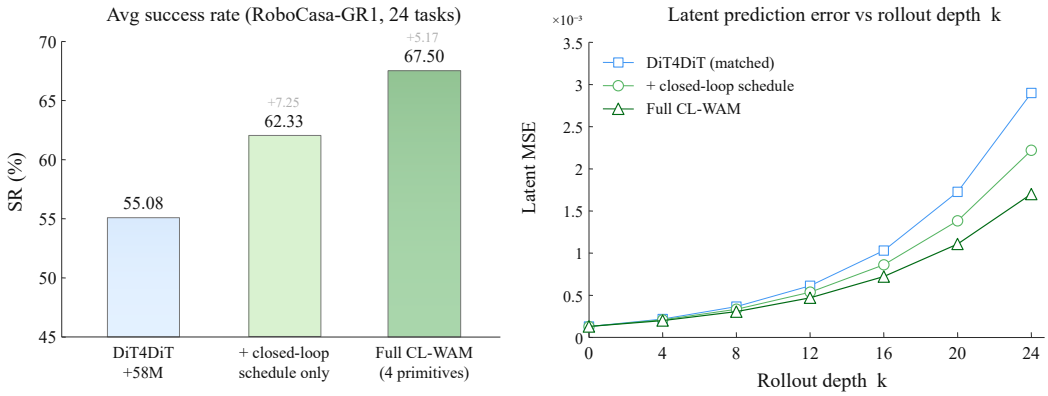


Figure 9: **Controlled validation of the closed-loop design.** Left: RoboCasa-GR1 average success rate under the DiT4DiT (+58M), the closed-loop schedule alone, and full CL-WAM. Right: latent prediction MSE as a function of rollout depth k. Prediction error accumulates more rapidly for DiT4DiT (matched), while the closed-loop schedule reduces error growth and full CL-WAM maintains the lowest error over long rollouts.

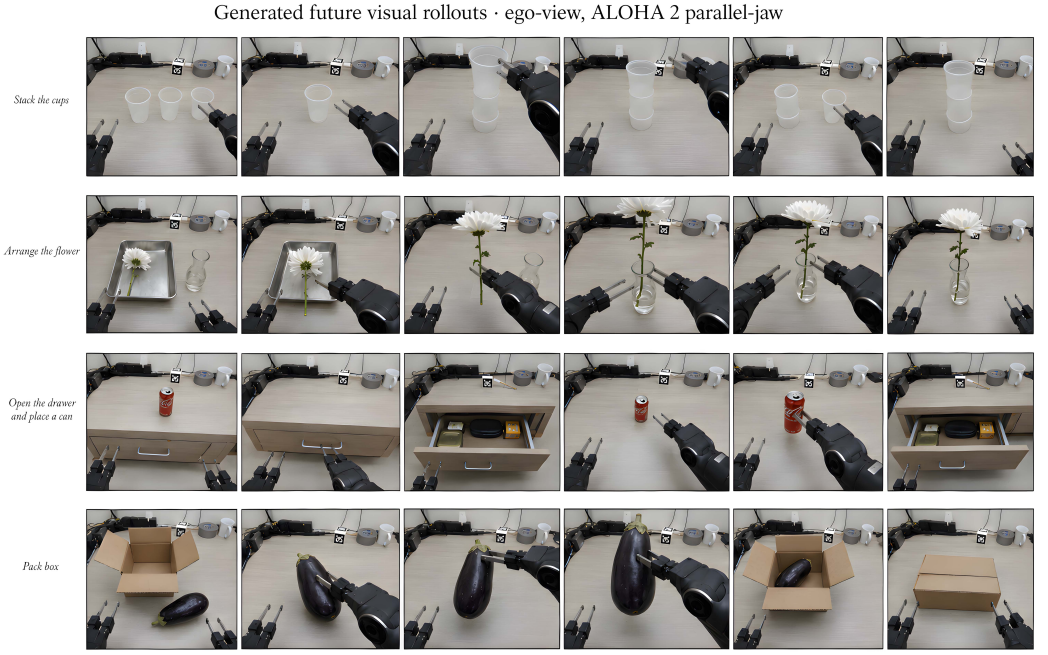


Figure 10: **Generated future visual rollouts.** Four representative real-world tasks (rows) sampled at six time steps each ($t = 0, 5, 10, 15, 20, 25$). All frames are first-person ego-view, matching the policy's only sensory input; the ALOHA 2 parallel-jaw gripper enters the bottom of every frame. Top to bottom: *Stack the cups* (multi-stage stacking), *Arrange the flower* (precision insertion), *Open the drawer and place a can* (extended-horizon manipulation with sub-goals), and *Pack box* (deformable-object handling). Across all four sequences, the video module produces visually coherent, task-relevant interactions and visually plausible grasp geometries.

recognizable object and gripper geometry and depict task-relevant gripper–object interactions. These examples provide a qualitative view of the video module's predictions over the sampled rollout positions.