Coordinating Humanoid Loco-Manipulation through Action-Conditioned Latent Foresight

Ju Dong[1,†], Xinyu Xiao[4], Yiming Zhou[4,†], Tianshuo Yang[2], Xuerong Liu[2], Lei Zhang[1], Jian Chen[5], Haocheng Zhao[3], Yu Fu[1], Yujie Wei[4], Diwen Zheng[3], Alois Christian Knoll[3], Angela P. Schoellig[3], Ping Luo[2,*], Jianwei Zhang[1]

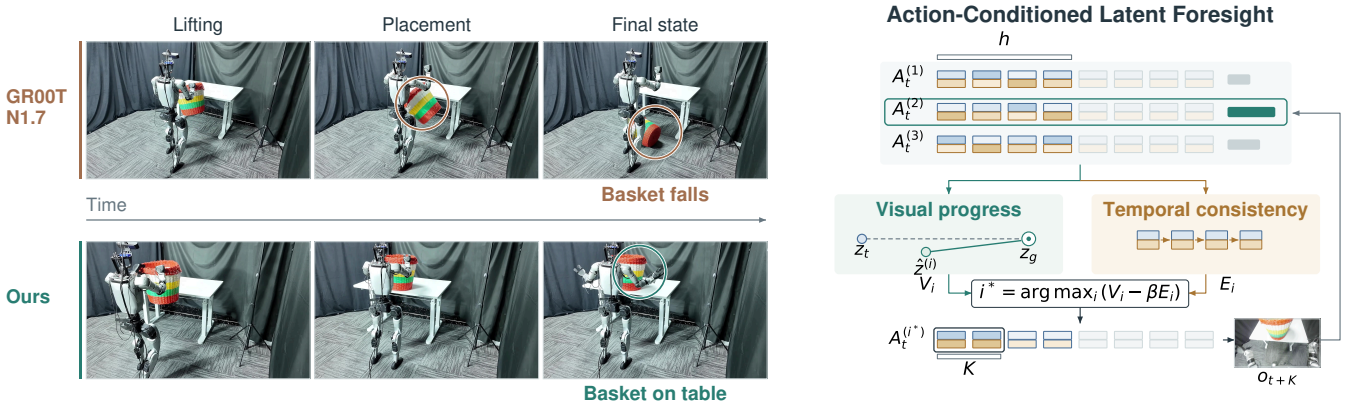


Fig. 1: Left: two independent Place Basket on Table recordings with GR00T N1.7 and our method. Right: each sampled sequence $A_t^{(i)}$ is scored on its first h commands. Visual progress V_i is the predicted reduction in distance to the subgoal z_g from current features z_t to predicted features $\hat{z}^{(i)}$; temporal-consistency energy E_i measures next-command prediction errors. The robot executes the first K commands of the best-scoring sequence i^* and replans from o_{t+K}.

***Abstract*—Humanoid loco-manipulation requires coordinated body motion and hand interaction over time, and a plausible-looking command sequence can still drop a held object or leave an interaction incomplete. We present action-conditioned latent foresight, a latent world–action model (WAM) that evaluates proposed command sequences before execution. An action generator proposes joint motion–hand command sequences from vision, language, and the robot state. An action-conditioned endpoint predictor estimates the visual features each sequence prefix reaches and measures progress toward a demonstration-derived subgoal, while a joint temporal prior scores temporal consistency given the robot state and preceding commands. The robot executes the first commands of the best-scoring sequence and replans from the next observation. During training, action consistency regularization reconstructs the first demonstrated command from observed and predicted feature changes, encouraging the predictor to retain action-relevant information. On ten tasks in the humanoid loco-manipulation simulation benchmark SIMPLE, our method achieves 89.7% average success, compared with 73.7% for Ψ_0 and 82.0% for DiT4DiT. On seven real-robot tasks, it achieves 85.7% under standard conditions and 65.7% under unseen object configurations, while GR00T N1.7 achieves 70.0% and 48.6%, respectively. Project page: https://ju6276.github.io/Action-Grounded-Foresight/.**

[†]Project leads. [*]Corresponding author.
[1]TAMS (Technical Aspects of Multimodal Systems), Department of Informatics, University of Hamburg, Hamburg, Germany.
[2]The University of Hong Kong, Hong Kong, China.
[3]Technical University of Munich, Germany.
[4]Kinetix AI, China.
[5]University of Science and Technology of China, Hefei, China.

I. Introduction

Everyday humanoid tasks require coordinated actions over time. To place a basket on a table, the robot must raise and position the basket while maintaining its grasp until the table supports it. To load a washer, it must open the door, carry a T-shirt to the drum without dropping it, and release it only once it is inside. These tasks connect whole-body motion with hand interaction: an execution decision concerns both the scene a command sequence will reach and the relationships among commands along the way.

Vision-language-action (VLA) policies generate commands from images and language instructions [1–3], while humanoid models incorporate human motion and predictive representations [4–7]. When an action generator proposes several sequences, the robot must decide which one to execute. Latent world models provide predictions for goal-directed planning [8, 9]. Most world–action models use future prediction to improve command generation [10–12] rather than to compare sampled sequences before execution. Existing verifiers use VLM judgments [13, 14], while DreamTrajectory combines predicted trajectory agreement with action smoothness [15]. Two signals are relevant for evaluating proposed humanoid actions: predicted visual progress evaluates the endpoint

relative to a task goal, while temporal consistency evaluates relationships among commands within the sequence.

We introduce *action-conditioned latent foresight*, a latent world–action model (WAM) that uses both signals for humanoid loco-manipulation (Fig. 1). An action generator proposes joint motion–hand command sequences; an action-conditioned endpoint predictor scores their visual progress toward a demonstration-derived subgoal, and a joint temporal prior scores their temporal consistency. The robot executes the first commands of the best-scoring sequence and replans from the next observation.

Our contributions are threefold:

- We propose action-conditioned latent foresight, which scores generated motion–hand sequences before execution using predicted visual progress and a learned joint temporal prior.
- We develop action consistency regularization for latent prediction. A training-only inverse action decoder reconstructs the first demonstrated command from both observed and predicted feature changes, encouraging the predicted features to retain information about the executed action.
- We evaluate ten SIMPLE [16] simulation tasks and seven real-robot tasks, including unseen object configurations. Our method exceeds the evaluated baselines in average success; closed-loop ablations show gains from action consistency regularization and joint temporal scoring.

II. Related Work

A. Humanoid VLAs and World–Action Models

VLA policies transfer vision–language representations to command generation [1, 2]. Heterogeneous co-training improves open-world generalization [3], and diffusion action heads generate continuous action chunks [17]. For humanoids, EgoVLA uses retargeted wrist and hand motion [18], Ψ_0 exploits egocentric pretraining [4], and WholeBodyVLA learns manipulation and locomotion latent actions from video before decoding upper-body and locomotion commands [5].

World–action models couple prediction with command generation. DreamZero jointly models video and actions [10]; DiT4DiT and MotionWAM condition action or motion diffusion on intermediate video-model features [6, 11]. MotionWAM combines SONIC motion codes with hand-control channels [6], while ω-0 generates controller-compatible actions from future embeddings [7]. WholeBodyWAM pretrains a motion expert to learn whole-body motion priors for robot action generation [19]. Fast-WAM and InternW0-Δ use future supervision during training without generating video at test time [12, 20]. In our method, the action generator supplies command sequences, and each sequence prefix is scored before execution using action-conditioned prediction and temporal consistency.

B. Action Supervision for Latent Prediction

Latent world models predict action-conditioned feature transitions for goal-directed planning [8, 9]. V-JEPA 2.1 learns dense video features [21], and a study of JEPA world models examines how representation, predictor, and planner choices affect physical planning [22]. VLA-JEPA learns VLM latent tokens through future-state prediction and uses them to condition flow-matching action generation [23]. Related models combine feature prediction with action modeling [24], or predict future latent states alongside actions [25]. Our predictor takes the proposed command prefix as input and estimates its endpoint features, allowing command sequences sampled from the same observation to receive different visual scores.

Action supervision can also arise from discrete or constrained continuous latents [26, 27], inverse dynamics that turn generated video into executable actions [28], or visual correspondences [29]. Our action consistency regularization uses an inverse action decoder to reconstruct the first demonstrated command from observed and predicted feature changes, providing a training objective for the endpoint predictor rather than a test-time action decoder.

C. Temporal Action Priors and Sequence Scoring

Action-sequence models [30, 31] and cross-embodiment action priors [32] learn regularities that guide generation. PRISM combines state-conditioned priors with planning distributions [33], and DynaGuide uses learned dynamics to steer diffusion [34]. Action verification evaluates proposed actions before execution: RoboMonkey learns action preferences with a VLM verifier [13], while FOREWARN aligns predicted latent rollouts with behavior descriptions for VLM-based action choice [14].

DreamTrajectory combines agreement with intended end-effector trajectories and fixed smoothness penalties [15]; LeWAM's DemoDPO uses demonstration preferences for offline policy refinement [25]. Our temporal prior learns to predict the next complete command from the initial robot state and preceding motion–hand commands. Its prediction error is combined with visual progress when scoring proposed actions.

III. Method

A. Problem Formulation and Receding-Horizon Scoring

Given an egocentric image o_t, instruction ℓ, and proprioceptive state s_t, our action generator samples N normalized command chunks $A_t^{(i)} \in \mathbb{R}^{H \times d_a}$:

$$A_t^{(i)} \sim \pi_\omega(\cdot \mid x_t), \quad A_t^{(i)} = (a_{t,0}^{(i)}, \ldots, a_{t,H-1}^{(i)}), \qquad (1)$$

where $x_t = (o_t, \ell, s_t)$. Each command contains body-motion components m_k and hand components u_k defined by the controller interface, with $d_a = d_m + d_u$. On the real robot, m_k is a SONIC motion latent [35] and u_k contains hand joint targets.

Let H be the proposal length, h the evaluation horizon, and K the executed prefix length, with $1 \le K \le h \le H$ and $h \ge 2$. The endpoint predictor and the temporal prior both score the prefix $B_t^{(i)} = A_t^{(i)}[0:h]$. Progress is measured toward temporally ordered visual subgoals $G = (g_1, \ldots, g_J)$ extracted from successful demonstrations, with active subgoal g_t.

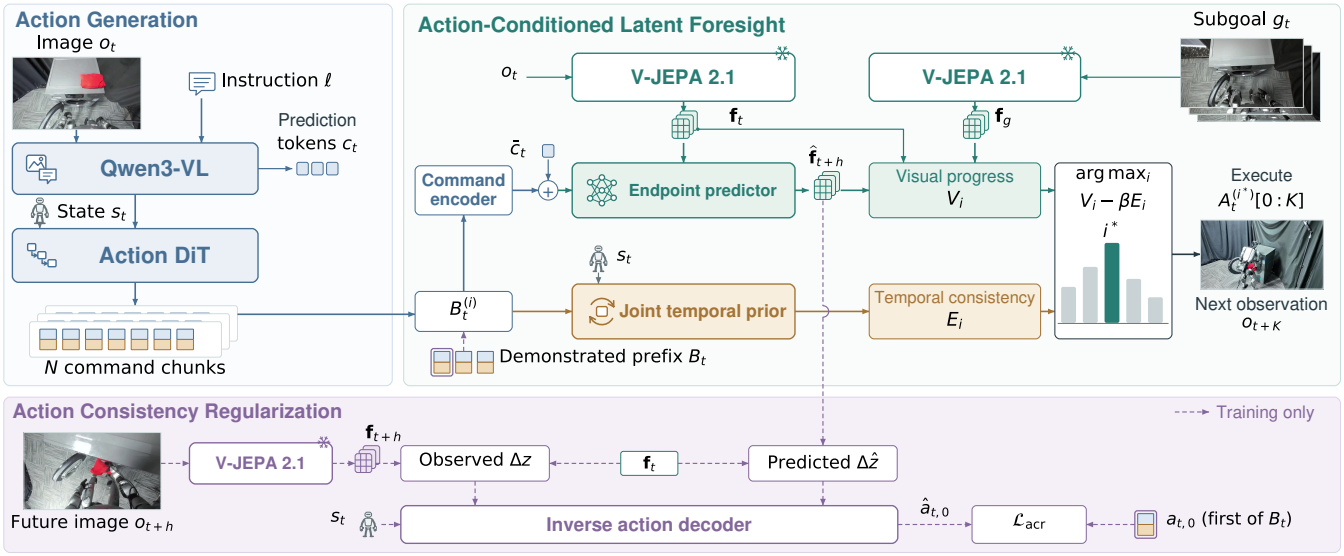


Fig. 2: Architecture. $\mathbf{f}$ denotes V-JEPA features. Each prefix $B_t^{(i)}$ receives visual progress V_i toward the active subgoal g_t (front of the stack) and temporal-consistency energy E_i. In training, the demonstrated prefix B_t replaces sampled prefixes, and its first command $a_{t,0}$ is reconstructed as $\hat{a}_{t,0}$ from Δz and from $\Delta\hat{z}$. Score bars are schematic; snowflakes: frozen weights.

Let z_t and z_g be normalized current and goal features, $\widehat{\mathbf{f}}_{t+h}^{(i)}$ the predicted endpoint features, and pool spatial mean pooling followed by normalization, so that $\hat{z}^{(i)} = \text{pool}(\widehat{\mathbf{f}}_{t+h}^{(i)})$. With cosine distance dist, each sequence receives a score that combines predicted visual progress with temporal-consistency energy:

$$V_i = \text{dist}(z_t, z_g) - \text{dist}(\hat{z}^{(i)}, z_g),$$
$$i^* = \arg\max_i \{V_i - \beta E_i\}, \quad E_i = E_\phi(B_t^{(i)}; s_t). \tag{2}$$

In a receding-horizon loop, the robot executes the first K commands of $A_t^{(i^*)}$ and repeats the decision from the next observation; $h > K$ provides lookahead beyond the next decision. A subgoal tracker advances g_t to the next subgoal in G when $\text{dist}(z_t, z_g) < \epsilon_g$.

B. Action Generation

Qwen3-VL-2B [36] encodes the image and instruction into separate policy tokens c_t^{act} and prediction tokens c_t (Fig. 2). Action DiT [37] alternates cross-attention to c_t^{act} with self-attention over state tokens, learned tokens, and noisy action tokens. Its flow-matching velocity head [38] generates command sequences through numerical integration from initial noise ξ.

C. Action-Conditioned Endpoint Dynamics

We fine-tune the V-JEPA 2.1 ViT-L/16 encoder (384 px) [21] with its official self-supervised training recipe on HIW-500 [39], over 500 hours of whole-body humanoid demonstrations collected through Unitree G1 teleoperation in real homes. We then freeze the resulting encoder for task training and inference; it also extracts the demonstration subgoals. Let Φ denote this encoder applied to a single image, repeated to the encoder clip length, with its last temporal feature block retained. Current, goal, and future images are encoded separately:

$$\mathbf{f}_t = \Phi(o_t), \quad \mathbf{f}_g = \Phi(g_t), \quad \mathbf{f}_{t+h} = \Phi(o_{t+h}). \tag{3}$$

Thus $z_t = \text{pool}(\mathbf{f}_t)$ and $z_g = \text{pool}(\mathbf{f}_g)$. During training, the demonstrated prefix $B_t = (a_{t,0}, \ldots, a_{t,h-1})$ starting at time t is paired with the observation o_{t+h}.

A command encoder C_η produces one token per command with positional encoding. We add the mean of the prediction tokens, $\bar{c}_t$, to each command token, giving $q_t(B) = C_\eta(B) + \bar{c}_t$. The endpoint predictor is

$$\widehat{\mathbf{f}}_{t+h} = \mathcal{P}_\theta(\mathbf{f}_t, q_t(B_t)). \tag{4}$$

D. Joint Temporal Prior

A shared GRU predicts the next complete motion–hand command from the initial robot state and preceding commands:

$$r_k = \text{GRU}_\phi(r_{k-1}, e_\phi([s_t; m_{k-1}; u_{k-1}])),$$
$$[\tilde{m}_k; \tilde{u}_k] = \text{MLP}_\phi(r_k), \qquad k = 1, \ldots, h-1. \tag{5}$$

The hidden state starts at $r_0 = 0$ for each sequence and s_t is repeated at every step, so both motion and hand predictions depend on the preceding commands of both groups. The energy combines mean squared prediction errors for motion and hand commands:

$$E_m = \frac{\sum_{k=1}^{h-1} \|m_k - \tilde{m}_k\|_2^2}{(h-1)d_m},$$
$$E_u = \frac{\sum_{k=1}^{h-1} \|u_k - \tilde{u}_k\|_2^2}{(h-1)d_u},$$
$$E_\phi(B; s_t) = w_m E_m + w_u E_u, \quad w_m + w_u = 1. \tag{6}$$

We use $w_m = d_m/d_a$ and $w_u = d_u/d_a$ so each normalized coordinate contributes equally.

E. Action Consistency Regularization

We train the endpoint predictor with endpoint feature matching and action consistency regularization. Dense visual supervision and pooled alignment are

$$\begin{aligned}\mathcal{L}_{\mathrm{wm}} &= \frac{\left\|\widehat{\mathbf{f}}_{t+h} - \mathrm{sg}(\mathbf{f}_{t+h})\right\|_1}{|\mathbf{f}_{t+h}|},\\ \Delta z &= \mathrm{pool}(\mathbf{f}_{t+h}) - \mathrm{pool}(\mathbf{f}_t),\\ \Delta\hat{z} &= \mathrm{pool}(\widehat{\mathbf{f}}_{t+h}) - \mathrm{sg}(\mathrm{pool}(\mathbf{f}_t)),\\ \mathcal{L}_{\Delta} &= \mathrm{MSE}(\Delta\hat{z}, \mathrm{sg}(\Delta z)).\end{aligned} \tag{7}$$

Here sg stops gradients. The shared frozen current feature cancels in $\mathcal{L}_{\Delta}$, yielding pooled endpoint alignment.

Action consistency regularization uses an inverse action decoder D_ψ to reconstruct the first demonstrated command from observed and predicted feature changes:

$$\begin{aligned}\mathcal{L}_{\mathrm{acr}} = \ &\mathrm{MSE}(D_\psi(\mathrm{sg}(\Delta z), s_t), a_{t,0})\\ &+ \mathrm{MSE}(D_\psi(\Delta\hat{z}, s_t), a_{t,0}).\end{aligned} \tag{8}$$

The observed branch trains the decoder on demonstrated transitions. The predicted branch passes command-reconstruction gradients to the predictor and its conditioning modules, encouraging predictions to retain action-relevant information.

The prior learns from demonstrations $\mathcal{D}$ through $\mathcal{L}_{\mathrm{prior}} = \mathbb{E}_{\mathcal{D}} E_\phi(B; s_t)$. Together with the action generator's flow-matching loss, the total objective is

$$\begin{aligned}\mathcal{L} = \mathcal{L}_{\mathrm{act}} &+ \lambda_{\mathrm{wm}}\mathcal{L}_{\mathrm{wm}} + \lambda_{\Delta}\mathcal{L}_{\Delta}\\ &+ \lambda_{\mathrm{acr}}\mathcal{L}_{\mathrm{acr}} + \lambda_{\mathrm{prior}}\mathcal{L}_{\mathrm{prior}}.\end{aligned} \tag{9}$$

All trainable components are optimized jointly with this objective.

IV. Experiments

A. Experimental Setup

TABLE I: Platforms and implementation settings.

Setting	Value
Real robot	Unitree G1
Egocentric camera	ZED2
Training hardware	8× NVIDIA A100
Evaluation (sim./real)	1× NVIDIA RTX 4090D (48 GB)
Onboard deployment test	Jetson Orin NX (16 GB)
Proposal length	Sim.: $H = 30$; real: $H = 40$
Evaluation horizon	$h = 20$
Executed prefix length	$K = 10$; commands at 30 Hz
Command chunks	$N = 8$
Joint temporal prior	GRU, 2 layers, hidden width 512
Loss weights	$\lambda_{\mathrm{wm}} = 0.1$, $\lambda_{\Delta} = 0.05$, $\lambda_{\mathrm{acr}} = 0.02$, $\lambda_{\mathrm{prior}} = 0.01$
Score weight	$\beta = 0.1$
Subgoals per task	$J = 5$
Subgoal tracker	$\epsilon_g = 0.15$

Tasks and demonstrations: We evaluate ten tasks from SIMPLE [16], a simulation benchmark for humanoid loco-manipulation (Fig. 3), using its official demonstrations collected through motion planning and VR teleoperation. Its standard 36-dimensional interface contains 14 hand joint targets and 22 channels for arm, torso, and locomotion control. We run ten trials per task at each of three randomization levels: L0 varies distractors and backgrounds; L1 adds materials and lighting; L2 adds initial object and robot poses. Real-robot evaluation covers seven G1 tasks (Figs. 1 and 3) under standard conditions and unseen object configurations, with ten trials per task and condition. The real-robot interface combines a 64-dimensional SONIC motion latent with 14 hand joint targets. Demonstrations are collected using SONIC: 400 per task, except for 200 in Place Basket on Table. Unseen configurations change object identity, placement, orientation, or appearance to states absent from the corresponding task demonstrations, while preserving the task objective.

Visual subgoals: For offline subgoal extraction, we mean-pool all spatiotemporal tokens from each repeated-image clip using the frozen visual encoder, accumulate feature changes between consecutive frames, and sample J subgoal images at evenly spaced values of this cumulative change. The images remain in temporal order, and standard and unseen-configuration trials use the same sequence for each task; the baselines do not use subgoals.

Implementation: Table I summarizes the platforms and implementation settings. Each method trained in our experiments uses a separate policy per task. Simulation policies use 40,000 training steps and a per-GPU batch size of 32; real-robot policies use 90,000 steps and a per-GPU batch size of 4.

Baselines: For the simulation comparison, we train and evaluate DiT4DiT [11], Fast-WAM [12], and VLA-JEPA [23]; task-success results for the other baselines are taken from SIMPLE [16]. For the real-robot comparison, we train and evaluate GR00T N1.7 [17], Ψ_0 [4], and DiT4DiT.

B. Task Performance

Tables II–IV report closed-loop task success.

Simulation: Our method achieves 89.7% average success across ten SIMPLE tasks, compared with 82.0% for DiT4DiT, 73.7% for Ψ_0, and 49.0% for VLA-JEPA. Success averages 95.6% on the six main tasks and 80.8% on the four additional tasks. OpenFaucet is the most difficult task, with 17/30 successes.

Real robot: Our method achieves 85.7% success under standard conditions and 65.7% under unseen object configurations, compared with 70.0% and 48.6% for GR00T N1.7, the strongest baseline. Under standard conditions, our method succeeds in all ten trials on four tasks. The gain of 15.7 percentage points comes mainly from Load Washer (10/10 versus 2/10). GR00T N1.7 failures on this task include dropping the T-shirt during transport, releasing it before it is inside the drum, and not opening the washer door. The first two resemble the Place Basket on Table failure in Fig. 1, where the basket falls during placement: the grasp is not maintained until the object reaches

its target. Dispose of Trash is the only task where our method has a lower success rate than GR00T N1.7; its failures include the partial lid opening in Fig. 6. Under unseen configurations, baseline success ranges from 0/10 to 6/10 per task. Our method has the highest or tied-highest success on all seven tasks and reaches at least 7/10 on four of them.

C. Qualitative Results and Generalization

In Water Plant and Load Washer (Fig. 4), the robot uses its right hand in standard trials and its left hand in the unseen object configurations with a reversed watering-can spout or a T-shirt placed on its left, completing both tasks. The left-hand T-shirt grasp does not appear in the demonstrations.

D. Data Efficiency

We compare training with 25, 50, and 100 demonstrations per task on five SIMPLE tasks: OpenOven, PickAndPlace-AndHugContainer, OpenFaucet, OpenTrashCan, and Mobile P&P.

At each of the three demonstration budgets, our method has higher average success than Ψ_0 on this five-task subset (Fig. 5). With 100 demonstrations per task, it reaches 84.0%, compared with 56.7% for Ψ_0, a gain of 27.3 percentage points.

E. Failure Analysis

OpenFaucet accounts for 13 of 31 simulation failures (41.9%). The robot repeatedly moves its hand near the faucet

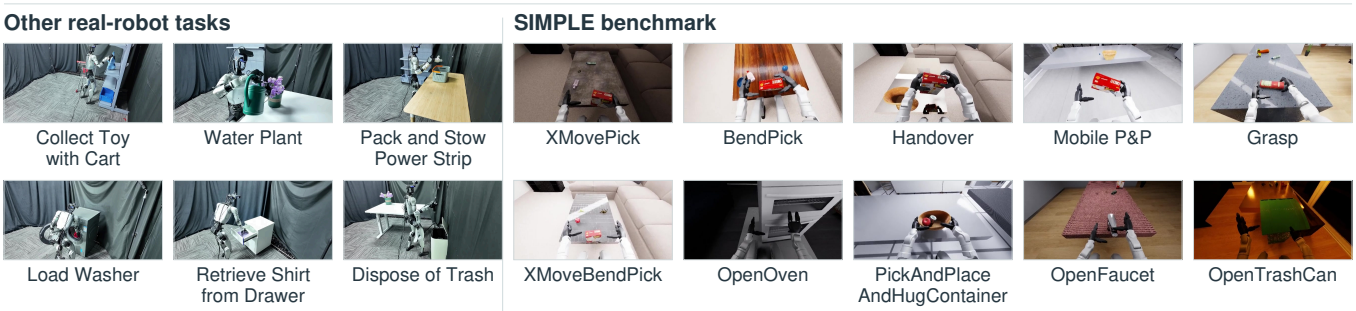


Fig. 3: Representative task scenes: the other six real-robot tasks and the ten SIMPLE simulation tasks. Place Basket on Table is shown in Fig. 1.

TABLE II: Six main SIMPLE tasks: L0/L1/L2 successes per ten trials. DiT4DiT, Fast-WAM, VLA-JEPA, and Ours are evaluated by us; other rows are from SIMPLE [16]. Task names follow SIMPLE's results table: Mobile P&P is LocomotionPickBetweenTables, and BendPick and Grasp (TabletopGrasp) use motion-planning demonstrations. Bold/underline mark best/second-best.

	Main SIMPLE Tasks (L0/L1/L2)						
Method	XMovePick	BendPick	Handover	Mobile P&P	Grasp	XMoveBendPick	**Avg. (%)**
Ψ_0 [4]	**10**|**10**|6	**10**|**10**|**10**	7|7|**10**	7|5|6	**10**|**10**|8	**10**|9|9	85.6
GR00T N1.6 [17]	**10**|**10**|7	7|7|6	1|3|3	0|0|0	9|9|7	4|4|1	48.9
$\pi_{0.5}$ [3]	7|5|1	**10**|**10**|8	5|4|5	3|3|3	**10**|**10**|8	0|0|0	51.1
InternVLA-M1 [40]	0|0|0	5|5|0	0|0|0	0|0|0	0|0|0	3|5|7	13.9
H-RDT [41]	0|0|2	0|0|1	0|1|0	0|0|0	0|0|0	0|0|0	2.2
DreamZero [10]	**10**|**10**|**10**	9|9|8	7|8|9	5|3|3	9|**10**|7	0|0|1	65.6
EgoVLA [18]	0|1|2	7|5|8	0|4|3	0|0|0	**10**|**10**|7	3|5|4	38.3
DP [30]	3|3|2	**10**|8|6	3|2|4	4|0|0	8|9|8	0|0|0	38.9
ACT [31]	**10**|**10**|5	**10**|9|9	7|7|**10**	5|5|5	**10**|**10**|8	9|**10**|**10**	82.8
DiT4DiT [11]	**10**|9|9	9|9|**10**	**9**|9|**10**	7|7|7	**10**|9|7	9|9|9	87.8
Fast-WAM [12]	**10**|**10**|6	0|3|1	6|5|7	0|0|0	**10**|7|7	0|0|0	40.0
VLA-JEPA [23]	0|0|0	1|0|0	7|8|9	7|7|**9**	9|8|5	0|3|4	42.8
Ours	**10**|**10**|7	**10**|9|**10**	8|**10**|9	**10**|**10**|**9**	**10**|**10**|**10**	**10**|**10**|**10**	**95.6**

TABLE III: Four additional SIMPLE tasks. Results for Ψ_0 are from SIMPLE [16]; we evaluate the other methods. Reporting follows Table II. Overall is the average success rate across all ten tasks.

	Additional SIMPLE Tasks (L0/L1/L2)					
Method	OpenOven	PickAndPlace AndHugContainer	OpenFaucet	OpenTrashCan	**Avg. (%)** (4 tasks)	**Overall (%)** (10 tasks)
Ψ_0 [4]	7|5|4	7|6|3	3|3|4	7|**9**|**9**	55.8	73.7
DiT4DiT [11]	**10**|**10**|9	7|8|**8**	5|6|**7**	6|7|5	73.3	82.0
Fast-WAM [12]	1|1|0	0|0|0	2|1|0	2|2|1	8.3	27.3
VLA-JEPA [23]	8|9|7	8|8|7	**6**|2|4	4|3|4	58.3	49.0
Ours	9|**10**|**10**	**9**|**9**|**8**	5|**7**|5	**8**|8|**9**	**80.8**	**89.7**

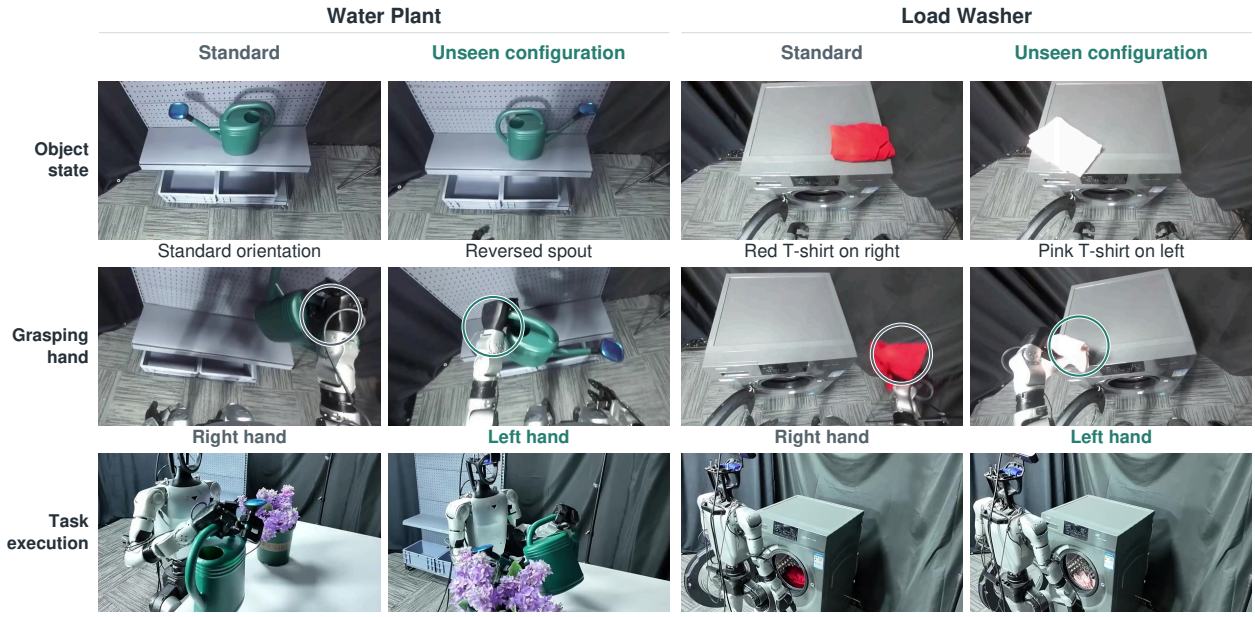


Fig. 4: Grasping behavior of our method under standard and unseen object configurations. Rings mark the grasping hand.

TABLE IV: Real-robot success under standard and unseen object configurations. Cells show successes out of ten trials; Avg. is the mean success rate across tasks (%). Bold/underline mark best/second-best.

Method	Dispose of Trash	Collect Toy with Cart	Water Plant	Pack and Stow Power Strip	Load Washer	Retrieve Shirt from Drawer	Place Basket on Table	**Avg.**
(a) Standard conditions								
GR00T N1.7	**9/10**	6/10	8/10	**8/10**	2/10	8/10	8/10	70.0
Ψ_0	2/10	6/10	6/10	4/10	2/10	2/10	1/10	32.9
DiT4DiT	4/10	**8/10**	8/10	6/10	8/10	4/10	6/10	62.9
Ours	6/10	6/10	**10/10**	**8/10**	**10/10**	**10/10**	**10/10**	**85.7**
(b) Unseen object configurations								
GR00T N1.7	**4/10**	**4/10**	5/10	5/10	6/10	4/10	6/10	48.6
Ψ_0	2/10	3/10	0/10	3/10	3/10	2/10	2/10	21.4
DiT4DiT	1/10	**4/10**	5/10	5/10	4/10	3/10	5/10	38.6
Ours	**4/10**	**4/10**	**8/10**	**6/10**	**7/10**	**8/10**	**9/10**	**65.7**

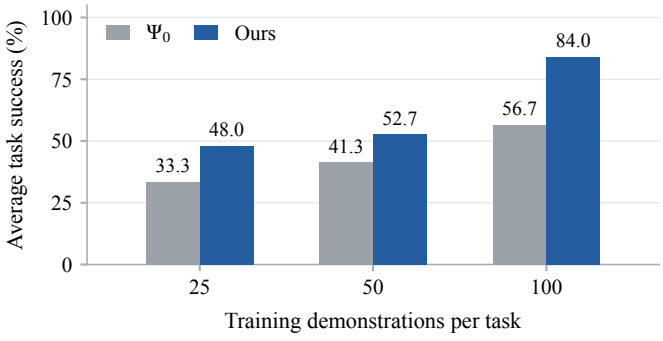


Fig. 5: Average success on five SIMPLE tasks versus the number of training demonstrations per task.

handle but fails to open it (Fig. 6(a)). In Dispose of Trash, pressing the pedal opens the lid only partly; the robot keeps holding the object and does not release it (Fig. 6(b)). In Collect Toy with Cart, the robot lifts the soft toy but cannot hold it securely, and the toy falls before reaching the cart (Fig. 6(c)).

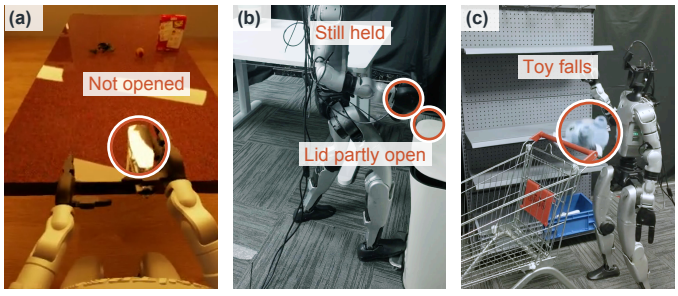


Fig. 6: Failure cases of our method: (a) OpenFaucet (simulation), (b) Dispose of Trash, and (c) Collect Toy with Cart.

F. Ablation Studies

We evaluate three design choices through closed-loop task success: prediction formulation, action consistency regularization, and temporal scoring.

1) Prediction Formulation: The feature-shift variant jointly encodes a short clip and predicts each visual feature block

from the preceding ones. Command tokens are resampled to one group per visual block, added position-wise to the VLM prediction tokens, and interleaved with the blocks in a block-causal predictor; at inference, the current image is repeated to form the clip, and the predicted blocks are pooled for scoring. The endpoint model instead encodes current and future observations separately and predicts a single endpoint from the first h commands. With joint encoding, feature-shift inputs can carry information from later frames during training that the repeated-image clip lacks at inference; separate encoding avoids this mismatch. The endpoint model reaches 89.7% versus 84.3% on the ten SIMPLE tasks (Table V), improving both task sets.

TABLE V: Prediction formulation: closed-loop success on the SIMPLE benchmark (%).

Configuration	Main (6 tasks)	Additional (4 tasks)	Overall (10 tasks)
Feature-shift variant	91.7	73.3	84.3
Endpoint (Ours)	**95.6**	**80.8**	**89.7**

Component ablations: Both ablations use the four additional SIMPLE tasks in Table III, on which the full model has lower average success than on the six main tasks: ten complete episodes per task at each level (L0/L1/L2), totaling 120 per configuration. Within each ablation, configurations share demonstration data, training budgets, test initial conditions, and subgoals.

2) Action Consistency Regularization: We train two models with action consistency regularization enabled or disabled, keeping the other objectives unchanged. Both use visual-only scoring at inference (Table VI).

Action consistency regularization increases success from 66/120 episodes (55.0%) to 79/120 (65.8%), a gain of 10.8 percentage points. Success is higher on all four tasks.

TABLE VI: Action consistency regularization (ACR) with visual-only scoring. Counts show successes out of ten trials at L0/L1/L2; Avg. is the success rate over all 120 episodes.

Configuration	Open Oven	PickAnd PlaceAnd Hug Container	Open Faucet	Open TrashCan	Avg. (%)
w/o ACR	7|7|6	7|6|6	3|4|3	6|6|5	55.0
w/ ACR	8|8|8	8|7|7	4|5|4	7|6|7	65.8

3) Temporal Prior: We use the trained action generator, endpoint predictor, and joint prior from the full model, and compare alternative temporal scores at every decision (Table VII). The first- and second-order penalties average $\|a_k - a_{k-1}\|_2^2/d_a$ and $\|a_k - 2a_{k-1} + a_{k-2}\|_2^2/d_a$, respectively, over valid steps of the normalized command prefix. The factorized prior predicts each group from its own history and the current state; the joint prior uses the histories of both groups.

The factorized prior is trained with the other components frozen, using the joint prior's group weights and a parameter count within 1%. All temporal scores use a fixed $\beta = 0.1$; visual-only scoring sets $\beta = 0$.

The joint prior achieves the highest closed-loop success among the tested scoring methods: 97/120 episodes (80.8%). This is 15.0 percentage points above visual-only scoring and 9.2 above the factorized prior. Scoring with first- and second-order smoothness penalties achieves 67.5% and 70.0% success, respectively, 1.7–4.2 percentage points above visual-only scoring, compared with 5.8 for the factorized prior. The joint prior also exceeds the factorized prior on each of the four tasks, consistent with scoring dependencies between motion and hand commands.

TABLE VII: Temporal-scoring ablation. Counts follow Table VI; the joint-prior row uses the same four-task results as Table III.

Configuration	Open Oven	PickAnd PlaceAnd Hug Container	Open Faucet	Open TrashCan	Avg. (%)
Visual only	8|8|8	8|7|7	4|5|4	7|6|7	65.8
1st-order smoothness	8|8|8	8|8|7	4|5|4	7|7|7	67.5
2nd-order smoothness	8|9|8	8|7|8	4|6|5	7|7|7	70.0
Factorized prior	8|9|9	8|8|8	4|6|4	8|7|7	71.7
Joint prior (Ours)	9|10|10	9|9|8	5|7|5	8|8|9	80.8

G. Inference Cost and Deployment

Our method uses less than half the GPU memory of DiT4DiT and Fast-WAM and runs faster (Table VIII); GR00T N1.6 remains faster at similar memory use. Our task-success evaluations use the RTX 4090D listed in Table I. In an additional onboard deployment test, our method runs on the robot's Jetson Orin NX (16 GB), averaging 12.84 GB of system memory and an inference frequency of 0.73 Hz across seven tasks.

TABLE VIII: Inference cost on SIMPLE. Hz denotes command chunks per second; timing for Ours includes sequence generation, visual prediction, and prior scoring.

Method	GPU memory (GB) ↓	Trainable params. (B)	Inference (Hz) ↑
GR00T N1.6 [17]	6.8	1.62	15.46
DiT4DiT [11]	21	2.35	3.8
Fast-WAM [12]	28	6.7	5.2
Ours	7.2	3.0	9.1

V. Conclusion and Future Work

Action-conditioned latent foresight scores generated motion–hand sequences by predicted visual progress and temporal consistency before execution. It achieves 89.7% success in simulation and 85.7% and 65.7% on the real robot under standard and unseen object configurations. In closed-loop ablations, action consistency regularization adds 10.8 percentage points and the joint prior exceeds the factorized prior by 9.2 percentage points.

The failure cases include incomplete opening of articulated mechanisms and loss of grasp on a soft object. The visual score also relies on task-specific demonstration subgoal images. In the onboard deployment test, inference runs at 0.73 Hz. Future work will add contact and object-state supervision to latent prediction and extend temporal scoring across consecutive command chunks.

REFERENCES

[1] M. J. Kim, K. Pertsch, S. Karamcheti *et al.*, "OpenVLA: An Open-Source Vision-Language-Action Model," in *Conference on Robot Learning*, 2024.

[2] K. Black, N. Brown, D. Driess *et al.*, "π_0: A Vision-Language-Action Flow Model for General Robot Control," in *Robotics: Science and Systems*, 2025.

[3] Physical Intelligence, K. Black, N. Brown *et al.*, "$\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization," *arXiv preprint arXiv:2504.16054*, 2025.

[4] S. Wei, H. Jing, B. Li *et al.*, "Ψ_0: An Open Foundation Model Towards Universal Humanoid Loco-Manipulation," *arXiv preprint arXiv:2603.12263*, 2026.

[5] H. Jiang, J. Chen, Q. Bu *et al.*, "WholeBodyVLA: Towards Unified Latent VLA for Whole-Body Loco-Manipulation Control," *arXiv preprint arXiv:2512.11047*, 2025.

[6] J. Zheng, T. Ma, Y. Fan, Z. Wang, S. Yang, and J. Liang, "MotionWAM: Towards Foundation World Action Models for Real-Time Humanoid Loco-Manipulation," *arXiv preprint arXiv:2606.09215*, 2026.

[7] Z. Li, Z. Zhang, Y. Wei *et al.*, "ω-0: A Latent Predictive World Action Model for Concurrent Humanoid Loco-Manipulation," *arXiv preprint arXiv:2608.06375*, 2026.

[8] G. Zhou, H. Pan, Y. LeCun, and L. Pinto, "DINO-WM: World Models on Pre-trained Visual Features enable Zero-shot Planning," in *International Conference on Machine Learning*, 2025.

[9] M. Assran, A. Bardes, D. Fan *et al.*, "V-JEPA 2: Self-Supervised Video Models Enable Understanding, Prediction and Planning," *arXiv preprint arXiv:2506.09985*, 2025.

[10] S. Ye, Y. Ge, K. Zheng *et al.*, "World Action Models are Zero-shot Policies," *arXiv preprint arXiv:2602.15922*, 2026.

[11] T. Ma, J. Zheng, Z. Wang *et al.*, "DiT4DiT: Jointly Modeling Video Dynamics and Actions for Generalizable Robot Control," *arXiv preprint arXiv:2603.10448*, 2026.

[12] T. Yuan, Z. Dong, Y. Liu, and H. Zhao, "Fast-WAM: Do World Action Models Need Test-time Future Imagination?" *arXiv preprint arXiv:2603.16666*, 2026.

[13] J. Kwok, C. Agia, R. Sinha *et al.*, "RoboMonkey: Scaling Test-Time Sampling and Verification for Vision-Language-Action Models," *arXiv preprint arXiv:2506.17811*, 2025.

[14] Y. Wu, R. Tian, G. Swamy, and A. Bajcsy, "From Foresight to Forethought: VLM-In-the-Loop Policy Steering via Latent Alignment," in *Robotics: Science and Systems*, 2025.

[15] Z. Yang, W. Zhang, X. Chen *et al.*, "DreamTrajectory: Trajectory-Guided Action Generation with World Model Alignment for Mobile Manipulation," *arXiv preprint arXiv:2608.01381*, 2026.

[16] S. Wei, Z. Ni, J. Liu *et al.*, "SIMPLE: Simulation-Based Policy Learning and Evaluation for Humanoid Loco-manipulation," *arXiv preprint arXiv:2606.08278v1*, 2026.

[17] NVIDIA, J. Bjorck, F. Castañeda *et al.*, "GR00T N1: An Open Foundation Model for Generalist Humanoid Robots," *arXiv preprint arXiv:2503.14734*, 2025.

[18] R. Yang, Q. Yu, Y. Wu *et al.*, "EgoVLA: Learning Vision-Language-Action Models from Egocentric Human Videos," *arXiv preprint arXiv:2507.12440*, 2025.

[19] B. Zhang, Q. Zhang, S. Bai *et al.*, "WholeBodyWAM: Learning Whole-Body World Action Models with Scalable Motion Priors," *arXiv preprint arXiv:2609.18197*, 2026.

[20] X. Miao, Z. Li, B. Fang *et al.*, "InternW0-Δ: A World Action Model Bridging Predictive Dynamics and Actions with 20K+ Hours of Open Data," *arXiv preprint arXiv:2609.31394*, 2026.

[21] L. Mur-Labadia, M. Muckley, A. Bar *et al.*, "V-JEPA 2.1: Unlocking Dense Features in Video Self-Supervised Learning," *arXiv preprint arXiv:2603.14482*, 2026.

[22] B. Terver, T.-Y. Yang, J. Ponce, A. Bardes, and Y. LeCun, "What Drives Success in Physical Planning with Joint-Embedding Predictive World Models?" *Transactions on Machine Learning Research*, 2026.

[23] J. Sun, W. Zhang, Z. Qi *et al.*, "VLA-JEPA: Enhancing Vision-Language-Action Model with Latent World Model," *arXiv preprint arXiv:2602.10098*, 2026.

[24] Y. Lin, J. He, S. Bao *et al.*, "JEPA-WAM: Learning Vision-Language-Action Policies with Joint-Embedding World Modeling," *arXiv preprint arXiv:2608.09381*, 2026.

[25] X. Fang, B. Duan, H. Wu, J. Wang, and G.-J. Qi, "LeWAM: Latent evolving World Action Model," *arXiv preprint arXiv:2609.27455*, 2026.

[26] S. Ye, J. Jang, B. Jeon *et al.*, "Latent Action Pretraining from Videos," in *International Conference on Learning Representations*, 2025.

[27] Q. Garrido, T. Nagarajan, B. Terver, N. Ballas, Y. LeCun, and M. Rabbat, "Learning Latent Action World Models In The Wild," *arXiv preprint arXiv:2601.05230*, 2026.

[28] Y. Du, M. Yang, B. Dai *et al.*, "Learning Universal Policies via Text-Guided Video Generation," in *Advances in Neural Information Processing Systems*, 2023.

[29] P.-C. Ko, J. Mao, Y. Du, S.-H. Sun, and J. B. Tenenbaum, "Learning to Act from Actionless Videos through Dense Correspondences," in *International Conference on Learning Representations*, 2024.

[30] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. C. Burchfiel, and S. Song, "Diffusion Policy: Visuomotor Policy Learning via Action Diffusion," in *Robotics: Science and Systems*, 2023.

[31] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware," in *Robotics: Science and Systems*, 2023.

[32] D. Jing, T. Zhang, J. Liu *et al.*, "Learning Action Priors for Cross-embodiment Robot Manipulation," *arXiv preprint arXiv:2606.26095*, 2026.

[33] Y. Wang, J. Xia, B. Sun *et al.*, "PRISM: PRior-guided Imagination Sampling in world Models," *arXiv preprint arXiv:2606.07974*, 2026.

[34] M. Du and S. Song, "DynaGuide: Steering Diffusion Polices with Active Dynamic Guidance," in *Advances in Neural Information Processing Systems*, 2025.

[35] Z. Luo, Y. Yuan, T. Wang *et al.*, "SONIC: Supersizing Motion Tracking for Natural Humanoid Whole-Body Control," *Science Robotics*, vol. 11, no. 117, p. eaed4592, 2026.

[36] S. Bai, Y. Cai, R. Chen *et al.*, "Qwen3-VL Technical Report," *arXiv preprint arXiv:2511.21631*, 2025.

[37] W. Peebles and S. Xie, "Scalable Diffusion Models with Transformers," in *IEEE/CVF International Conference on Computer Vision*, 2023.

[38] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow Matching for Generative Modeling," in *International Conference on Learning Representations*, 2023.

[39] BitRobot, Unitree, and Hugging Face, "HIW-500: Humanoids In-the-Wild Dataset for Robot Learning," https://bitrobot-foundation.github.io/humanoids-in-the-wild-500-hours/, 2026.

[40] X. Chen, Y. Chen, Y. Fu *et al.*, "InternVLA-M1: A Spatially Guided Vision-Language-Action Framework for Generalist Robot Policy," *arXiv preprint arXiv:2510.13778*, 2025.

[41] H. Bi, L. Wu, T. Lin *et al.*, "H-RDT: Human Manipulation Enhanced Bimanual Robotic Manipulation," *arXiv preprint arXiv:2507.23523*, 2025.

APPENDIX A
UNSEEN OBJECT CONFIGURATIONS

TABLE S1: Unseen object configurations.

Task	Configurations
Dispose of Trash	Tissue packet; tea bottle.
Collect Toy with Cart	Two arrangements of the blue target toy and two neighboring distractors.
Water Plant	Watering can shifted left; shifted right; spout reversed.
Pack and Stow Power Strip	Power strip rotated; repositioned on the tabletop.
Load Washer	Purple or pink T-shirt, placed on either the right or left side of the washer top.
Retrieve Shirt from Drawer	Target T-shirt repositioned; gray T-shirt nearby; pink T-shirt nearby.
Place Basket on Table	Blue-rimmed basket replacing the red-rimmed basket.

APPENDIX B
VISUAL SUBGOAL EXTRACTION

Dispose of Trash



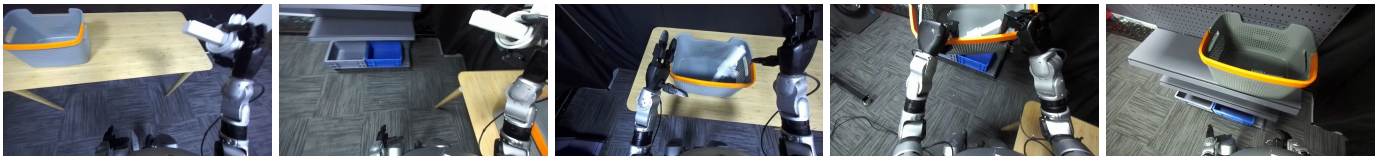
Collect Toy with Cart



Water Plant



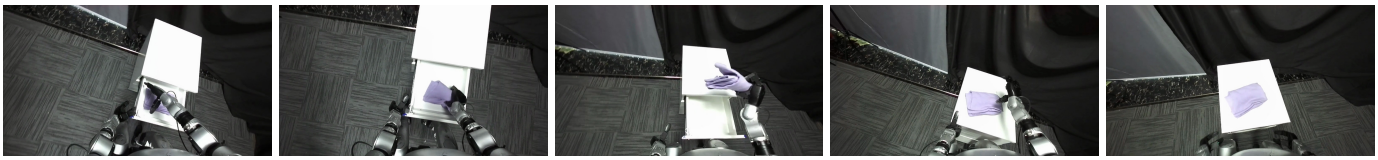
Pack and Stow Power Strip



Load Washer



Retrieve Shirt from Drawer



Place Basket on Table



Fig. S1: Visual subgoals extracted with the HIW-500-fine-tuned V-JEPA 2.1 encoder. Each row shows five temporally ordered subgoals from one task demonstration.