LocoWM: High-Precision Locomotion through World-Model-Guided Residual Adaptation

Anonymous authors
Paper under double-blind review

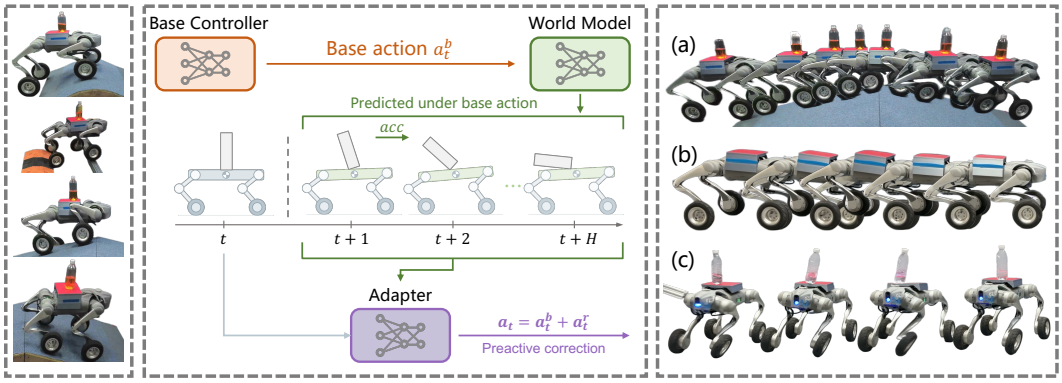


Figure 1: **Left:** Real-world transport of unsecured payloads across varied terrains. **Center:** LocoWM combines the base controller, action-conditioned world model, and residual adapter in a sequential pipeline: the base controller proposes a_t^b, the world model predicts future task substates conditioned on a_t^b, and the residual adapter uses these predictions to generate a preactive correction a_t^r, yielding the executed action $a_t = a_t^b + a_t^r$. **Right:** Real-world demonstrations of (a) terrain leveling, (b) acceleration compensation, and (c) push recovery.

Abstract

High-precision locomotion combines motion-command tracking with precise regulation of task-relevant physical states, enabling robots to interact reliably with their surroundings during motion. Joint end-to-end optimization can leave precision objectives insufficiently optimized, while reactive residual control adjusts actions only after deviations become observable. We present **LocoWM**, a world-model-guided preactive residual adaptation framework for high-precision locomotion. A base policy provides command-following locomotion, while an action-conditioned world model predicts a sequence of future physical states from proprioceptive history and the proposed base action. A residual adapter conditions on this predicted sequence to generate additive action corrections that compensate for anticipated deviations. Two-stage training first learns locomotion and action-conditioned dynamics, then freezes both modules while training the adapter, separating locomotion acquisition from precision adaptation. Experiments spanning terrain leveling, acceleration compensation, and push recovery demonstrate improved control precision and disturbance robustness over end-to-end and reactive residual baselines. Demos are available at: LocoWM.

1 Introduction

Reinforcement learning has enabled agile legged locomotion (Hwangbo et al., 2019; Lee et al., 2020), efficient skill acquisition (Rudin et al., 2022), and traversal of challenging terrain with legs and wheels (Miki et al., 2022; Lee et al., 2024). Many tasks additionally require precise regulation of task-relevant body poses or physical states while following motion commands. For example, humanoids may need to stabilize end effectors while walking (Jang et al., 2026), and wheeled-legged robots may need to coordinate body orientation with acceleration (Kanno et al., 2025). These tasks exemplify the challenge of *high-precision locomotion*: meeting task-specific precision requirements while following motion commands. Joint end-to-end optimization can yield competent locomotion while leaving task-specific precision objectives insufficiently optimized (Huang et al., 2026). To

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.

address this imbalance, precision learning can be separated from locomotion acquisition, allowing task-specific compensation for locomotion-induced disturbances.

Residual control provides a modular way to improve task precision by learning corrections to a base controller (Yu & Rosendo, 2022; Zhao et al., 2025). Existing observation-conditioned residual controllers generate corrections from current and historical measurements (Jang et al., 2026; Huang et al., 2026). Such *reactive* feedback responds to observed deviations without explicitly predicting their subsequent evolution. Forecasting future task states enables *preactive* correction, allowing the controller to compensate for anticipated deviations before they become observable (Liu et al., 2026). Model predictive control (MPC) implements this anticipatory principle by using analytical dynamics to predict future states and optimize actions online (Di Carlo et al., 2018). Its effectiveness depends on model accuracy and computational resources, while unmodeled dynamics can degrade precision and robustness (Cheng et al., 2025; Zhou et al., 2026).

Learning dynamics directly from interaction (Chua et al., 2018; Ha & Schmidhuber, 2018) can supply the forecasts needed for preactive control without relying on an analytical model. Recent world-model methods learn latent transition dynamics and use them to predict future states for control (Lai et al., 2025; Li et al., 2026; Zhang et al., 2026a). This learned dynamics interface exposes action-conditioned future consequences in a form that a downstream controller can consume directly.

We introduce LocoWM, a framework for high-precision locomotion through world-model-guided residual adaptation. The central idea is to forecast the physical quantities defining task precision, termed *task substates*, and use their anticipated evolution to guide action corrections. As illustrated in the center of Figure 1, the base policy proposes an action, and the world model predicts a short future substate sequence from this action and proprioceptive history. The residual adapter combines this forecast with current observations and the base action to generate a correction for anticipated deviations, yielding the executed action $a_t = a_t^b + a_t^r$.

We evaluate LocoWM on a Go2-W wheeled-legged robot through the three tasks illustrated on the right of Figure 1. Terrain leveling tests posture regulation under uneven support, acceleration compensation tests the coordination of body tilt with acceleration and braking, and push recovery tests stability recovery after external disturbances. Our main contributions are:

- We introduce a physically meaningful task-substate interface for LocoWM that unifies the precision objective, world-model prediction target, and residual-control input around the same task-relevant quantities.
- We develop a two-stage preactive residual architecture that first learns locomotion and task-substate dynamics, then freezes both modules while training an adapter for task precision and command following. At inference, the future sequence is predicted in one forward pass and directly guides action corrections without online trajectory optimization.
- Experiments demonstrate higher task success rates than end-to-end policies and reactive residual baselines. Relative to the current-state reconstruction baseline, LocoWM improves success by up to 29.6 percentage points and achieves maximum reductions of 91.8% in roll RMS error and 83.0% in peak vertical acceleration across the evaluated conditions.

2 Related Work

High-Precision Locomotion and Residual Adaptation. High-precision locomotion couples motion-command tracking with the stabilization of a task-relevant body part or payload. SEEC (Jang et al., 2026) and Hold My Beer (Li et al., 2025b) stabilize humanoid end effectors under locomotion-induced disturbances, while SteadyTray (Huang et al., 2026) learns to balance unsecured objects during tray transport. For wheeled-legged robots, base-angle adjustment can compensate for acceleration-induced payload tilt (Kanno et al., 2025). Residual learning provides a modular way to adapt an existing controller to such task-specific objectives (Yu & Rosendo, 2022; Ankile et al., 2026). Existing residual methods can be broadly divided according to the information used to condition the correction. Observation-conditioned residuals derive corrections from current and historical measurements (Jang et al., 2026; Huang et al., 2026); because they react only after a deviation becomes observable, they are inherently reactive. To anticipate the effect of an action before the resulting deviation occurs, model-based residual controllers use analytical dynamics models to forecast future states, often within MPC (Jeon et al., 2026; Cheng et al., 2025; Zhou et al., 2026). Analytical-model look-ahead is brittle

under model mismatch and unmodeled contact or payload dynamics. LocoWM instead uses a learned short-horizon dynamics predictor to condition preactive residual corrections on action-conditioned future task substates in a single forward pass.

World Models for Embodied AI. World models learn dynamics from interaction and support embodied control through several routes. *Imagined rollouts* use predicted trajectories to generate synthetic experience for policy learning (Li et al., 2025a; Levy et al., 2026; Wu et al., 2022). *Online planning* predicts future transitions and optimizes action sequences at decision time (Hansen et al., 2025; Shirwatkar et al., 2025; Lai et al., 2025). *Predictive representation learning* uses future-state modeling objectives to construct features for downstream policies (Zhang et al., 2026b; Danesh et al., 2026; Choi et al., 2026; Yan et al., 2026). LocoWM uses predicted task-relevant future states at inference time to condition a residual controller. Specifically, it forecasts action-conditioned task substates in one forward pass and maps them to an additive action correction without online action-sequence optimization.

3 METHODOLOGY

We present LocoWM, a framework for high-precision locomotion that combines a locomotion policy, an action-conditioned world model, and a residual adapter. The world model maps proprioceptive history and the base policy's proposed action to a short sequence of task substates in one forward pass. The residual adapter uses this sequence to generate additive action corrections without online action optimization. By conditioning on predicted future task substates, LocoWM shifts residual control from *reactive* correction of observed deviations to *preactive* correction of anticipated ones.

3.1 HIGH-PRECISION LOCOMOTION

Problem Formulation. We consider a *high-precision locomotion* task that simultaneously satisfies two objectives: (1) tracking the user command, and (2) precisely regulating a task-relevant joint or physical quantity throughout locomotion. Examples include keeping a mounted platform level (Huang et al., 2026), stabilizing an end effector (Li et al., 2025b; Jang et al., 2026), and limiting the tilt of an unsecured payload (Kanno et al., 2025).

We model the task as a goal-conditioned Markov decision process $(\mathcal{S}, \mathcal{O}, \mathcal{G}, \mathcal{A}, \mathcal{Z}, P, r, \gamma)$. At time t, $s_t \in \mathcal{S}$ denotes the full state, including the robot, terrain, and any mounted payload or platform; command $g_t \in \mathcal{G}$ specifies the desired robot motion; action $a_t \in \mathcal{A}$ specifies leg-joint position targets and wheel-joint velocity targets for the low-level controllers; observation $o_t = [s^{\text{prop}}_{t-L:t},\, g_{t-L:t},\, a_{t-L-1:t-1}] \in \mathcal{O}$ stacks an L-step history of proprioceptive signals, user commands, and previous actions, which constitute the information available to the policy at deployment; transition dynamics $P(s_{t+1} \mid s_t, a_t)$ govern how the full state evolves; reward $r = r^{\text{track}} + r^{\text{prec}}$ combines a command-tracking term and a precision term, and $\gamma \in (0, 1)$ is the discount factor.

We introduce a *task substate* $z_t = h(s_t) \in \mathcal{Z}$, where h projects the full state onto a semantically meaningful quantity that must be kept precise; z_t may be privileged, computable during training but absent from the observation o_t. It plays three roles and this is what unifies LocoWM: (i) it **defines the precision objective** through a reward term $r^{\text{prec}}(z_t, z_t^*)$, where z_t^* is the desired precision target; (ii) it is **what the world model predicts**: LocoWM predicts $\hat{z}_{:+H}$ as the task-relevant consequence of the current locomotion action;[1] and (iii) its prediction $\hat{z}_{:+H}$ **conditions the residual adapter**, enabling preactive correction of predicted deviations.

Task Instantiation. In this paper, we instantiate the formulation on a legged-wheel robot that carries a tray on its back (Huang et al., 2026). The goal is to regulate the back orientation so that the mounted tray counteracts the robot's motion, staying level during steady locomotion and tilting to absorb inertial forces under acceleration, thereby ensuring steady payload transportation (Kanno et al., 2025). The task substate is defined as $z_t = [\boldsymbol{\theta}, \boldsymbol{a}, \boldsymbol{\omega}]$, $\boldsymbol{\theta} = [\theta_p, \theta_r]$ denote the back's pitch and roll angles, $\boldsymbol{a}$ are the linear accelerations, and $\boldsymbol{\omega} = \dot{\boldsymbol{\theta}}$ are the angular velocities. The high-precision objective requires the components of z_t to jointly satisfy

$$\begin{bmatrix}\theta_p \\ \theta_r\end{bmatrix} = \arctan\left(\frac{-1}{\sqrt{a_x^2 + a_y^2 + g^2}}\begin{bmatrix}a_x \\ a_y\end{bmatrix}\right), \quad [\boldsymbol{\omega}, a_z] = \mathbf{0}, \tag{1}$$

[1] : $+H$ denotes $t+1 : t+H$

where g is the gravitational acceleration. Geometrically, this objective aligns the tray normal with the *effective gravity*, which is the resultant of true gravity and the inertial reaction to acceleration. Under this objective, the payload experiences no in-plane force and therefore does not slide or tip over, while $[\boldsymbol{\omega}, a_z] = \mathbf{0}$ further ensures steady transportation by eliminating oscillations about the target orientation and vertical jolts. The detailed task definition and Implementation can be found in Appendix A .

3.2 LocoWM: World-Model Guided Residual Adaptation

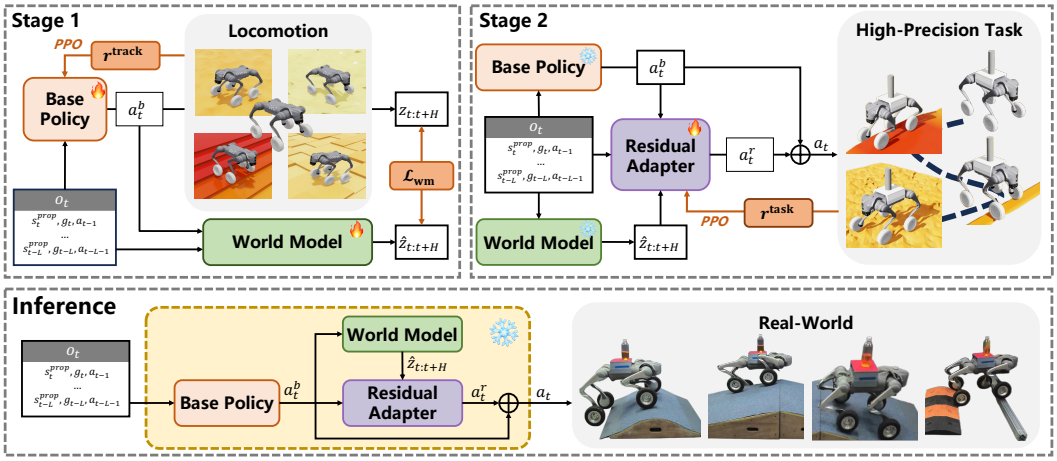


Figure 2: Overview of LocoWM. **Stage 1: Base Policy & World Model.** The base policy learns command-following locomotion, while the world model predicts future task substates from the same trajectories using a separate loss. **Stage 2: Residual Adapter.** The base policy and world model are frozen, and the adapter learns precision corrections. **Inference.** All modules are frozen; the base action, predicted substates, and residual correction are computed sequentially, yielding $a_t = a_t^b + a_t^r$.

LocoWM separates locomotion acquisition from precision adaptation through the two training stages in Figure 2. The base policy supplies the motion prior, the world model predicts its task-relevant consequences, and the residual adapter learns corrections on top of both frozen modules. This factorization is essential for the precision objective: if tracking and precision were optimized jointly from the outset, the dense tracking reward could dominate the sparser precision signal, allowing the robot to keep moving without regulating the tray (Huang et al., 2026). We therefore train the base policy with the tracking objective alone, freeze it together with the world model, and then learn a lightweight adapter that cannot perturb the locomotion prior.

Stage 1: Learning locomotion and task-substate dynamics. The first stage learns the motion prior and the task-substate dynamics from a shared rollout stream. We train the base policy $\pi^b(o_t)$ with PPO (Schulman et al., 2017) using r^{track}, which combines velocity tracking with regularization of motion smoothness, joint and torque limits, and contact behavior. The regularization encourages smooth, low-agitation motion, which defines the task-substate dynamics that the world model must forecast and the residual adapter must later compensate. An erratic base policy would make both prediction and precision control harder. To obtain a reusable prior under the conditions encountered after adaptation, Stage 1 randomizes robot's physical parameters, terrain, and external pushes.

Alongside the base policy, we train an action-conditioned world model on the same rollout data,

$$f_\psi : (o_t, a_t^b) \longmapsto \hat{z}_{:+H}, \qquad a_t^b = \pi^b(o_t), \tag{2}$$

which predicts the next task-substate sequence in a single forward pass. The prediction target $z_{:+H}$ is collected along the base-policy trajectory and is available only during training. The world model minimizes

$$\mathcal{L}_{\text{WM}}(\psi) = \mathbb{E}_{(o_t, a_t^b, z_{:+H}) \sim \pi^b} \left[\left\| f_\psi(o_t, a_t^b) - z_{:+H} \right\|^2 \right]. \tag{3}$$

The base policy and world model share interaction data but have separate optimization objectives. The prediction gradient updates only ψ and never propagates into the base policy.

Three design properties make f_ψ particularly well suited to guide the residual and distinguish LocoWM from prior world-model designs.

- **Causal and on-distribution dynamics**: because f_ψ is trained on the base-policy rollout stream and is queried with the same observation–base-action interface at inference, its training inputs match the interface used at deployment. Its predictions therefore describe how the system will evolve under the base policy's actions, rather than the behavior of an oracle controller or a generic forward model.
- **Observable-to-semantic prediction**: the input (o_t, a_t^b) is purely deployable, while the output is the low-dimensional task substate itself rather than the full state or a latent code. Predicting only the task substate keeps f_ψ physically interpretable: $\hat{z}_{:+H}$ corresponds directly to future tilt and angular velocity, without requiring an analytical dynamics model.
- **Single-shot prediction**: the entire chunk $\hat{z}_{:+H}$ is emitted in one forward pass, so LocoWM can use the forecast to guide the adapter without online rollouts, or action-sequence optimization as in planner-based world models (Hansen et al., 2025; Shirwatkar et al., 2025).

Stage 2: Learning precision through residual adaptation. After Stage 1, we freeze both π^b and f_ψ and train a residual adapter

$$\pi^r : (o_t, a_t^b, \hat{z}_{:+H}) \longmapsto a_t^r, \qquad a_t = a_t^b + a_t^r. \tag{4}$$

The adapter's output layer is zero-initialized so that $a_t^r \approx \mathbf{0}$. At the start of Stage 2, the combined controller therefore has the same mean action as the frozen base policy, preserving the locomotion prior and providing a stable starting point. Corrective behavior emerges gradually through training instead of perturbing a competent controller from the first update.

Only the adapter and its value function are updated in Stage 2. PPO uses the combined reward $r_t^{\text{track}} + r_t^{\text{prec}}$, where r_t^{prec} penalizes deviation of the measured task substate from the high-precision objective. Because the base policy already follows the command g_t, the adapter can focus on regulating the task substate. The tracking term prevents corrective actions that would compromise command following. This stage thus optimizes precision on top of a fixed locomotion prior rather than changing the base-policy parameters. Reward terms, randomization, and curricula are detailed in Appendix B.

Inference: Predicting and correcting within the feedback loop. At each control step, proprioceptive measurements, commands, and previous actions update the observation history o_t. The frozen modules are evaluated in order:

$$a_t^b = \pi^b(o_t), \qquad \hat{z}_{:+H} = f_\psi(o_t, a_t^b), \qquad a_t^r = \pi^r(o_t, a_t^b, \hat{z}_{:+H}). \tag{5}$$

The sum $a_t = a_t^b + a_t^r$ is mapped to motion targets for the low-level controllers. The next observation closes the feedback loop, while no module is updated during deployment. The forecast describes task-substate evolution under the base-policy continuation and is provided to the adapter alongside the observation and the proposed base action. The complete forecast is produced once per step, and the adapter maps it directly to an action correction without iterative model rollouts or online action search.

From reactive to preactive control. Methods that derive corrections from current observations, observation histories, or a reconstructed current state can act only after a deviation appears in the observations and are therefore reactive (Jang et al., 2026; Huang et al., 2026). LocoWM adds the action-conditioned prediction $\hat{z}_{:+H}$, making future task-substate evolution an explicit input to the correction. For example, a predicted tilt at time $t+1$ can inform a residual action at time t, before that deviation becomes observable. This changes the residual controller from correcting an observed error to correcting an anticipated consequence of the proposed locomotion action.

Physical interface and modularity. The three modules are composed in native, physically meaningful spaces rather than through a shared latent representation. At the input, the predicted substate $\hat{z}_{:+H}$ lies in the same physical space as the observation o_t, allowing the adapter to ground the forecast directly against its sensory input. At the output, the correction is applied as an additive residual $a_t = a_t^b + a_t^r$ in the base policy's action space rather than through feature-wise (FiLM-style) modulation. The modules therefore do not require a shared encoder, which decouples their optimization and keeps the base policy, world model, and adapter independently trainable in their respective stages. The same task substate consequently links the precision objective, the prediction target, and the control input, while each module retains a separate training objective.

4 EXPERIMENTS

Our evaluation addresses four questions. **(Q1)** Does LocoWM improve terrain leveling and payload retention across terrains, and how does it affect command tracking (Section 4.2)? **(Q2)** Does LocoWM coordinate platform tilt with acceleration and braking to compensate for inertial forces (Section 4.3)? **(Q3)** How does payload-retention success vary with the direction and magnitude of external pushes (Section 4.4)? **(Q4)** Do these precision-control behaviors transfer zero-shot to the physical robot (Section 4.5)?

4.1 EXPERIMENTAL SETUP

Our main experiments run on a Unitree Go2-W wheeled-legged quadruped carrying a rigid, back-mounted stabilization platform 30×20 cm. [2] At deployment the policy reads proprioception only, with no external motion capture and no platform-mounted force sensor. [3]

Terrains. For the tasks defined in Section 3.1 and Appendix A, we evaluate on four terrain families, each designed to induce a distinct disturbance to the robot's pose.

- **Slopes**: require the robot to maintain a stable body pitch while climbing and descending.
- **Bumps**: repeatedly perturb the body as the robot traverses uneven ground.
- **One-sided bridge**: creates a persistent height difference between the left and right sides, requiring the robot to maintain a level body.
- **Rough terrain**: introduces continuous irregular disturbances.

Baselines. We compare LocoWM against four baselines that progressively add its key components. This design isolates the contribution of each component: *End-to-End* → *React* demonstrates the benefit of residual adaptation; *React* → *Recon* measures the value of privileged state reconstruction; *Recon* → *LocoWM* tests the benefit of preactive control by replacing current-state reconstruction with future task-substate prediction.

- **Base Policy**: The frozen locomotion policy trained solely on the velocity tracking reward.
- **End-to-End**: A monolithic, single-stage policy trained directly on both tracking and precision rewards.
- **React** (Huang et al., 2026): A reactive residual adapter conditioned exclusively on deployable observations o_t.
- **Recon** (Sun et al., 2025): Extends *React* with the reconstructed current task substate $\hat{z}_t$, retaining reactive control without an explicit future-state forecast.

Metrics. We evaluate platform precision, command tracking, and payload retention using four metrics. Formal definitions, units, and aggregation conventions are provided in Appendix B.7.

- **Platform angle error.** Pitch and roll errors measure deviations from the desired orientation. RMS and peak errors characterize sustained deviations and transient excursions, respectively.
- **Vertical acceleration.** Mean and peak acceleration magnitudes characterize ride smoothness and vertical impacts that can disturb the payload.
- **Command tracking error.** The discrepancy between commanded and measured velocity captures locomotion performance and its trade-off with payload stabilization.
- **Success rate.** The fraction of trials retaining the payload throughout the evaluation interval measures transport success.

4.2 TERRAIN LEVELING

We command the robot to gradually accelerate to a target velocity and then traverse each terrain at approximately constant speed. The metrics are reported in Table 1. LocoWM attains the most precise leveling on all terrains. A slope creates a fore–aft height difference, so climbing and descending load the *pitch* axis. LocoWM reduces the pitch RMS to 2.61°, compared with 28.18° for the Base Policy. A one-sided bridge raises the wheels on one side and therefore loads the *roll* axis. In this setting, LocoWM cuts the roll RMS to 0.92°, an order of magnitude below the next-best Recon and over 30× below the Base Policy. Bumps introduce abrupt contact disturbances, for which peak

[2] To examine whether the same design extends beyond this embodiment, we additionally evaluate the framework on bimanual humanoid tray transport; the experimental setup and results are provided in Appendix D.

[3] Additional implementation details are provided in Appendix B .

vertical acceleration measures impact attenuation. LocoWM limits $|a_z|_{\max}$ to 3.97 m/s^2, compared with 10.17 m/s^2 for Recon, indicating improved impact attenuation over current-state reconstruction. These lower errors relative to the reactive React and Recon baselines support the usefulness of future task-substate predictions for preactive residual correction during terrain traversal.

Higher payload-retention success can coexist with larger command-tracking errors, reflecting the trade-off between command following and payload stabilization. LocoWM matches the task-agnostic Base Policy closely and stays the lowest of all residual adapters, even overtaking the Base Policy on the bridge. Together, precise leveling and a smoother ride accompany the highest reported payload retention on every terrain, reaching 98.1% on the bridge and 87.5% on bumps.

Table 1: Terrain leveling across four terrains. Within each terrain, **bold** and underline mark the best and second best.

Simulation Results		Pitch (deg)		Roll (deg)		$\lvert a_z \rvert$ (m/s^2)		e^{track}	Succ (%)
Terrain	**Method**	RMS	peak	RMS	peak	mean	peak		
Slope	Base Policy	28.18	29.98	9.96	21.51	1.58	8.54	**0.28**	45.3
	End-to-End	15.18	32.38	2.99	6.15	0.89	7.19	0.42	62.8
	React	5.39	20.89	2.15	8.16	0.83	7.12	0.43	59.8
	Recon	3.01	11.97	1.25	7.14	0.74	5.94	0.46	61.5
	LocoWM(ours)	**2.61**	**3.90**	**0.31**	**1.23**	**0.41**	**3.10**	0.45	**91.1**
Bump	Base Policy	1.06	14.25	2.26	14.06	3.46	19.61	**0.65**	21.1
	End-to-End	0.99	17.04	1.31	15.41	3.06	19.47	0.65	50.2
	React	0.56	10.52	0.28	10.31	2.41	10.28	1.00	45.2
	Recon	0.42	11.51	0.11	10.32	2.37	10.17	0.93	67.7
	LocoWM(ours)	**0.18**	**3.00**	**0.09**	**2.47**	**1.22**	**3.97**	0.87	**87.5**
Bridge	Base Policy	5.91	16.90	30.81	44.40	1.91	21.27	0.43	9.1
	End-to-End	4.81	25.56	25.66	42.73	1.31	8.66	0.80	2.2
	React	2.35	10.44	21.07	21.31	1.38	15.68	0.68	78.4
	Recon	1.56	9.36	11.24	20.00	1.36	14.38	0.74	95.5
	LocoWM(ours)	**0.91**	**2.52**	**0.92**	**3.54**	**0.27**	**2.45**	**0.33**	**98.1**
Rough	Base Policy	4.47	12.47	7.46	10.47	1.43	8.55	**0.13**	91.1
	End-to-End	3.01	10.19	2.12	14.93	1.01	6.74	0.42	95.4
	React	1.84	9.63	2.24	5.86	1.00	**6.07**	0.33	95.6
	Recon	2.08	9.04	1.34	4.12	0.98	6.37	0.25	96.2
	LocoWM(ours)	**1.73**	**4.35**	**1.19**	**4.07**	**0.93**	6.17	0.20	**97.8**

4.3 Acceleration Compensation

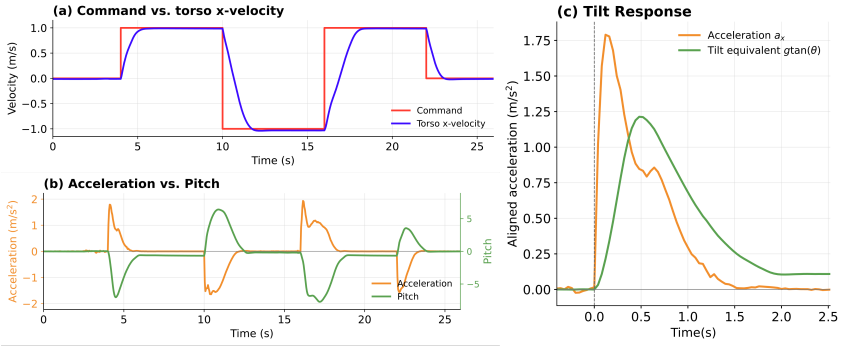


Figure 3: Acceleration compensation on flat ground. **(a)** Torso x-velocity tracking; **(b)** Acceleration–pitch responses; **(c)** Single-event tilt response.

We additionally evaluate the acceleration compensation task, in which the robot repeatedly accelerates and brakes on flat ground. Unlike the terrain leveling task, this maneuver requires the body to actively pitch to counteract the inertial forces generated by acceleration and braking. We command the robot to track a ± 1 m/s step-velocity profile, and as shown in Figure 3(a), the torso velocity follows both the positive and negative velocity commands. Over the resulting repeated acceleration and deceleration events, Figure 3(b) shows that each change in acceleration is accompanied by a corresponding pitch response. The single-event trace in Figure 3(c) further shows that the tilt-equivalent acceleration $g \tan \theta$ follows the torso acceleration a_x with the appropriate sign, indicating that LocoWM banks

into the acceleration. This coordinated response is consistent with LocoWM's preactive control: the residual adapter uses the world model's prediction of upcoming in-plane acceleration (a_x, a_y) to guide pitch correction, producing the compensation condition $g \tan \theta^\star = a$ in Equation (1). The platform therefore remains aligned with the effective gravity direction, allowing a freely placed payload to remain stable without sliding or toppling.

4.4 Push Recovery

In the push recovery task, we drive the robot forward at 0.5 m/s while applying pushes to its torso, varying both push direction and magnitude and recording the payload-retention success rate. As shown in Figure 4(a), the direction-averaged success rate decreases as the push force increases for all methods, whereas LocoWM consistently achieves the highest success rate across the tested force range.Figure 4(b) further reveals the directional dependence of this robustness: all methods tolerate larger lateral pushes than forward or backward pushes, and LocoWM maintains the outermost 50% success-rate envelope in nearly every direction. The corresponding success-rate map for LocoWM in Figure 4(c) exhibits the same anisotropic pattern, with high success rates extending to larger force magnitudes along the lateral directions than along the forward and backward directions. Together, these results show that LocoWM's preactive residual controller provides a broader payload-retention range under the evaluated disturbances. The success-rate maps of the baseline methods are provided in Appendix C.1 .

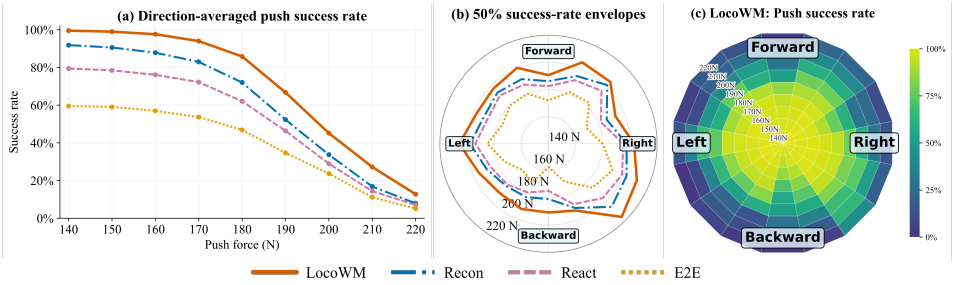
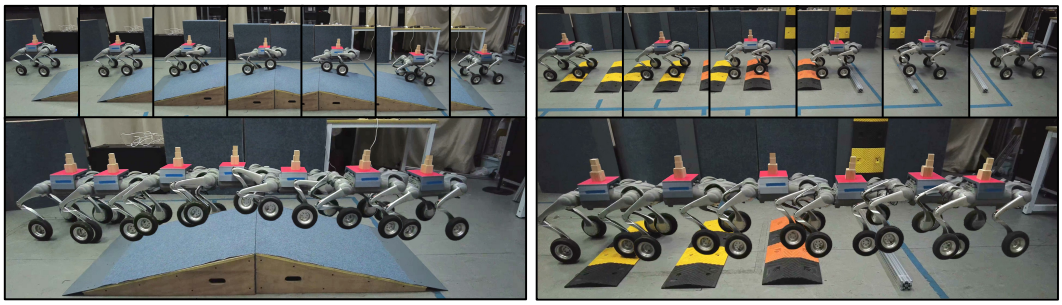


Figure 4: Payload retention during push recovery. **(a)** Direction-averaged success rates versus push magnitude. **(b)** The 50% success-rate envelopes of all methods. **(c)** LocoWM success-rate map: angle denotes push direction, radius denotes force magnitude, and color denotes success rate.

4.5 Real-World Results

We deploy LocoWM zero-shot on the physical robot and demonstrate terrain leveling, acceleration compensation, and push recovery. [4]



(a) Single-Bridge (b) Bump

Figure 5: Real-world terrain leveling on (a) a one-sided bridge and (b) bumps. In each panel, the top row shows successive video frames, while the bottom row combines robot cutouts extracted from individual frames onto a common background to illustrate the motion sequence.

[4] The remaining real-world experimental results are available in Appendix C.2 .

In the terrain leveling task, the robot carries an unsecured payload across a one-sided bridge, slopes, bumps, and wavy terrain. These surfaces combine gradual height changes with abrupt contacts, inducing sustained body inclination and coupled roll–pitch disturbances. As shown in Figure 5, LocoWM reconfigures the body and legs as the support height changes, keeping the back-mounted platform approximately level in both representative sequences while the payload remains upright.

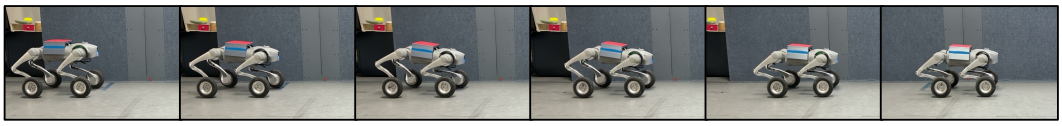
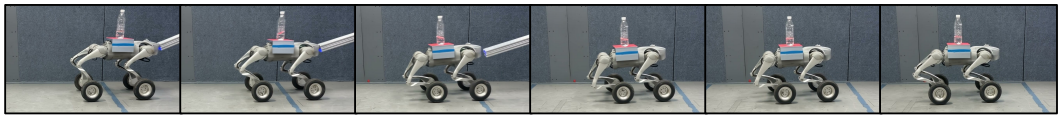
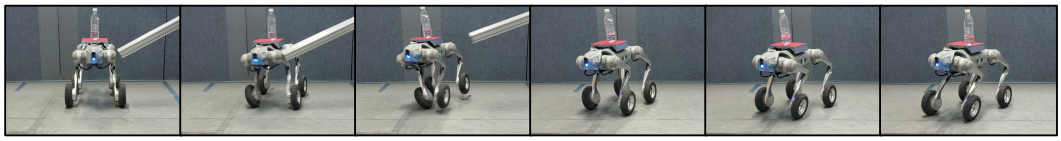


Figure 6: Real-world acceleration compensation on flat ground.

In the acceleration compensation task, the robot accelerates and decelerates on flat ground. The sequence in Figure 6 shows the platform pitching with the longitudinal motion, providing a real-world counterpart to the coordinated tilt response observed in simulation. The body therefore adjusts its orientation as the motion changes rather than leaving the platform fixed during acceleration.



(a) Push backward



(b) Push left

Figure 7: Real-world push recovery with an unsecured bottle. Frames progress from left to right through force application, the transient response, and recovery; the bottle remains on the platform in both sequences.

In the push recovery task, we apply backward and leftward forces to the torso while an unsecured bottle rests on the back-mounted platform. LocoWM changes the body posture during each disturbance and then brings the platform back toward level. Although the bottle tilts transiently, it remains on the platform throughout both recovery sequences. Together, these demonstrations show that the preactive residual controller transfers its precision behaviors to hardware without task-specific retraining.

5 Conclusion

We presented LocoWM, a world-model-guided preactive residual adaptation framework for high-precision locomotion. The framework combines a base locomotion policy, an action-conditioned world model that forecasts future task substates, and a residual adapter that uses these forecasts to refine the base action. Experiments on Go2-W demonstrate higher task success rates than end-to-end and reactive residual baselines. Relative to current-state reconstruction, success improves by up to 29.6 percentage points across the evaluated terrains, alongside lower posture errors and reduced vertical impacts. Acceleration compensation and push recovery further demonstrate coordinated body tilting and stability recovery after external disturbances.

Limitations. The main experiments center on the Go2-W wheeled-legged platform and three predefined task settings. To examine the generality of LocoWM beyond this setting, Appendix D further provides a simulation-based analysis on bimanual Unitree G1 humanoid transport.

Future Work. Future work will further investigate future physical-state prediction as an interface between established motor skills and task-specific precision objectives. This direction will examine how prediction-guided adaptation can extend existing motion skills to broader task settings while preserving command following and precision control. It will also explore richer task-substate representations and longer-horizon prediction for capturing more complex physical consequences.

Ethics Statement

This work studies high-precision robot locomotion and stable payload transport through learned dynamics and residual control. No human participants, animal subjects, or personal data were involved. The physical experiments with the Go2-W platform were conducted in a controlled environment with appropriate safety precautions.

Reproducibility Statement

The LocoWM architecture and two-stage training procedure are described in Section 3.2. Appendix A specifies the task substates and precision objectives. Appendix B details the simulation and control settings, observations, prediction targets, rewards, domain randomization, and curricula. The evaluation tasks, baselines, and metrics are defined in Section 4.1, with additional experimental results provided in Appendix C.

Use of AI

AI was used for writing assistance and language polishing during manuscript preparation. The authors reviewed and verified all AI-assisted content and take full responsibility for the research claims, experimental results, and final content of this paper.

References

Lars Ankile, Zhenyu Jiang, Rocky Duan, Guanya Shi, Pieter Abbeel, and Anusha Nagabandi. Residual Off-Policy RL for Finetuning Behavior Cloning Policies. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2026. URL https://arxiv.org/abs/2509.19301.

Jin Cheng, Dongho Kang, Gabriele Fadini, Guanya Shi, and Stelian Coros. RAMBO: RL-Augmented Model-Based Whole-Body Control for Loco-Manipulation. *IEEE Robotics and Automation Letters*, 10(9):9462–9469, 2025. URL https://doi.org/10.1109/lra.2025.3594984.

Yohan Choi, Min-Jun Kim, Jin-Sung Kim, Yong-Jae Kim, and Youn-Hee Han. DAWN: Noise-Robust Quadruped Parkour via Depth-Denoising World Models. *arXiv preprint arXiv:2609.29092*, 2026. URL https://arxiv.org/abs/2609.29092.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://dl.acm.org/doi/10.5555/3327345.3327385.

Mohamad H. Danesh, Chenhao Li, Amin Abyaneh, Anas Houssaini, Kirsty Ellis, Glen Berseth, Marco Hutter, and Hsiu-Chin Lin. Morphology-conditioned world model for cross-embodiment quadrupedal locomotion. *arXiv preprint arXiv:2604.08780*, 2026. URL https://arxiv.org/abs/2604.08780.

Jared Di Carlo, Patrick M. Wensing, Benjamin Katz, Gerardo Bledt, and Sangbae Kim. Dynamic Locomotion in the MIT Cheetah 3 Through Convex Model-Predictive Control. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–9, 2018. URL https://dspace.mit.edu/handle/1721.1/138000.

David Ha and Jürgen Schmidhuber. Recurrent World Models Facilitate Policy Evolution. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://dl.acm.org/doi/10.5555/3327144.3327171.

Nicklas Hansen, Jyothir S V, Vlad Sobal, Yann LeCun, Xiaolong Wang, and Hao Su. Hierarchical World Models as Visual Whole-Body Humanoid Controllers. In *International Conference on Learning Representations*, 2025. URL https://iclr.cc/virtual/2025/poster/30793.

Anlun Huang, Zhenyu Wu, Soofiyan Atar, Yuheng Zhi, and Michael Yip. SteadyTray: Learning Object Balancing Tasks in Humanoid Tray Transport via Residual Reinforcement Learning. *arXiv preprint arXiv:2603.10306*, 2026. URL https://arxiv.org/abs/2603.10306.

Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26):eaau5872, 2019. URL https://doi.org/10.1126/scirobotics.aau5872.

Jaehwi Jang, Zhuoheng Wang, Ziyi Zhou, Feiyang Wu, and Ye Zhao. SEEC: Stable End-Effector Control with Model-Enhanced Residual Learning for Humanoid Loco-Manipulation. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2026. URL https://arxiv.org/abs/2509.21231.

Se Hwan Jeon, Ho Jae Lee, Seungwoo Hong, and Sangbae Kim. Residual MPC: Blending Reinforcement Learning With GPU-Parallelized Model Predictive Control. *IEEE Transactions on Robotics*, 42:3469–3488, 2026. URL https://doi.org/10.1109/tro.2026.3721669.

Kimikage Kanno, Kenji Hashimoto, and Ikuo Mizuuchi. Relative Tilt Suppression of a Carried Object Using Base Link Angle Adjustment on a Quadruped-Wheeled Robot. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10684–10689, 2025. URL https://doi.org/10.1109/iros60139.2025.11247400.

Hang Lai, Jiahang Cao, Jiafeng Xu, Hongtao Wu, Yunfeng Lin, Tao Kong, Yong Yu, and Weinan Zhang. World Model-Based Perception for Visual Legged Locomotion. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11531–11537, 2025. URL https://doi.org/10.1109/icra55743.2025.11128762.

Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5(47):eabc5986, 2020. URL https://doi.org/10.1126/scirobotics.abc5986.

Joonho Lee, Marko Bjelonic, Alexander Reske, Lorenz Wellhausen, Takahiro Miki, and Marco Hutter. Learning robust autonomous navigation and locomotion for wheeled-legged robots. *Science Robotics*, 9(89):eadi9641, 2024. URL https://doi.org/10.1126/scirobotics.adi9641.

Jacob Levy, Tyler Westenbroek, Kevin Huang, Fernando Palafox, Patrick Yin, Shayegan Omidshafiei, Dong-Ki Kim, Abhishek Gupta, and David Fridovich-Keil. Simulation Distillation: Pretraining World Models in Simulation for Rapid Real-World Adaptation. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, jul 2026. URL https://www.roboticsproceedings.org/rss22/p017.html.

Chenhao Li, Andreas Krause, and Marco Hutter. Robotic World Model: A Neural Network Simulator for Robust Policy Optimization in Robotics. *arXiv preprint arXiv:2501.10100*, 2025a. URL https://arxiv.org/abs/2501.10100.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Luyao Zhang, Mingrui Yu, Zelin Gao, Nan Xue, Boyu Zhou, Xing Zhu, Mingyu Ding, Yujun Shen, and Yinghao Xu. Causal World Modeling for Robot Control. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, jul 2026. URL https://www.roboticsproceedings.org/rss22/p016.html.

Yitang Li, Yuanhang Zhang, Wenli Xiao, Chaoyi Pan, Haoyang Weng, Guanqi He, Tairan He, and Guanya Shi. Hold My Beer: Learning Gentle Humanoid Locomotion and End-Effector Stabilization Control. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 4506–4523. PMLR, 2025b. URL https://proceedings.mlr.press/v305/li25i.html.

Yuxi Liu, Lijun Han, Ziming Wang, Ao Zhang, Cong Yang, and Wei Sui. World-model-augmented visual locomotion for humanoids on foothold-constrained terrain, 2026. URL https://arxiv.org/abs/2609.02542.

Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62):eabk2822, 2022. URL https://doi.org/10.1126/scirobotics.abk2822.

Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 91–100. PMLR, 2022. URL https://proceedings.mlr.press/v164/rudin22a.html.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

Aditya Shirwatkar, Naman Saxena, Kishore Chandra, and Shishir Kolathaya. PIP-Loco: A Proprioceptive Infinite Horizon Planning Framework for Quadrupedal Robot Locomotion. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11198–11204, 2025. URL https://doi.org/10.1109/icra55743.2025.11128382.

Wandong Sun, Long Chen, Yongbo Su, Baoshi Cao, Yang Liu, and Zongwu Xie. Learning Humanoid Locomotion with World Model Reconstruction. *arXiv preprint arXiv:2502.16230*, 2025. URL https://arxiv.org/abs/2502.16230.

Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. DayDreamer: World Models for Physical Robot Learning. In *Proceedings of the 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 2226–2240. PMLR, 2022. URL https://proceedings.mlr.press/v205/wu23c.html.

Yashuai Yan, Tobias Egle, Christian Ott, and Dongheui Lee. Efficiently learning robust torque-based locomotion through reinforcement with model-based supervision. *IEEE Robotics and Automation Letters*, 11(4):4155–4162, 2026. URL https://doi.org/10.1109/lra.2026.3664534.

Chen Yu and Andre Rosendo. Multi-Modal Legged Locomotion Framework with Automated Residual Reinforcement Learning. *IEEE Robotics and Automation Letters*, 7(4):10312–10319, 2022. URL https://doi.org/10.1109/lra.2022.3191071.

Qihang Zhang, Lin Li, Luyao Zhang, Shuai Yang, Yiming Luo, Shuaiting Li, Ruilin Wang, Junke Wang, Jiahao Shao, Gangwei Xu, Jiaming Zhou, Yishu Shen, Yudong Jin, Fangyi Xu, Shuailei Ma, Jiaqi Liao, Guanxing Lu, Zifan Shi, Yongkun Wen, Yujie Zhao, Weixuan Tang, Xinyang Wang, Chaojian Li, Jiapeng Zhu, Ka Leong Cheng, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Native Video-Action Pretraining for Generalizable Robot Control. *arXiv preprint arXiv:2607.08639*, 2026a. URL https://arxiv.org/abs/2607.08639.

Zhikai Zhang, Jun Guo, Chao Chen, Jilong Wang, Chenghuai Lin, Yunrui Lian, Han Xue, Zhenrong Wang, Maoqi Liu, Jiangran Lyu, Huaping Liu, He Wang, and Li Yi. Track Any Motions under Any Disturbances. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2026b. URL https://arxiv.org/abs/2509.13833.

Siheng Zhao, Yanjie Ze, Yue Wang, C. Karen Liu, Pieter Abbeel, Guanya Shi, and Rocky Duan. ResMimic: From General Motion Tracking to Humanoid Whole-body Loco-Manipulation via Residual Learning. *arXiv preprint arXiv:2510.05070*, 2025. URL https://arxiv.org/abs/2510.05070.

Hongyu Zhou, Xiaoyu Zhang, and Vasileios Tzoumas. Adaptive Legged Locomotion via Online Learning for Model Predictive Control. *IEEE Robotics and Automation Letters*, 11(2):1778–1785, 2026. URL https://doi.org/10.1109/lra.2025.3644161.

APPENDICES

A Task Definition and Implementation **14**

A.1 Task Definition. 14

A.2 Task Substate Implementation. 14

A.3 High-Precision Objective. 14

B Implementation, Training, and Evaluation Details **15**

B.1 Observations 15

B.2 World Model Prediction 15

B.3 Rewards 16

B.4 Domain Randomization 16

B.5 Curriculum Learning 16

B.6 Network Architecture and Optimization 17

B.7 Metric Definitions and Aggregation 18

C Additional Experimental Results **19**

C.1 Simulation Results 19

C.2 Real-World Demonstrations 22

D Humanoid Transport **23**

D.1 Experimental Setup 23

D.2 Comparison Results 23

D.3 Visualization 23

A Task Definition and Implementation

A.1 Task Definition.

As described in Section 3.1, we study the stabilization of a back-mounted platform during wheeled-legged locomotion, with the goal of maintaining the stability of an unsecured payload. The platform is rigidly attached to the torso, so its orientation is regulated through coordinated leg and wheel motion. During acceleration and braking on flat ground, the platform must tilt to compensate for inertial loading and reduce the effective force parallel to its surface (Figure 8, left). On uneven terrain, changes in wheel contact heights disturb the torso orientation, requiring compensatory motion to keep the platform level (Figure 8, right). Together, these conditions require the robot to regulate platform orientation according to its motion while tracking locomotion commands.

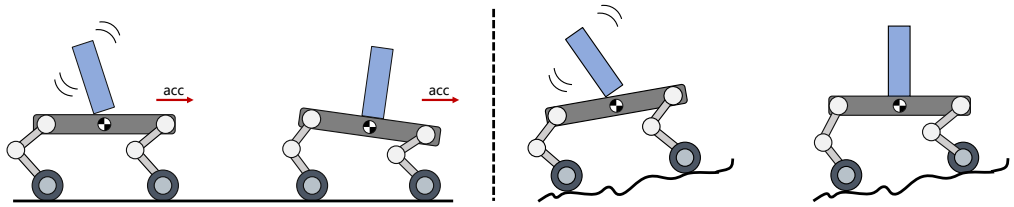


Figure 8: Back-mounted platform stabilization. **Left: Acceleration Compensation.** Acceleration can destabilize a payload on a level platform; coordinated body tilting compensates for inertial loading. **Right: Terrain Leveling.** Uneven support heights disturb the platform orientation; adjustments to the leg configuration counteract torso tilt and maintain a level platform.

A.2 Task Substate Implementation.

The task involves platform pitch and roll, linear acceleration, and angular velocity. The implemented task substate uses the nine-dimensional encoding, concatenating base angular velocity, world-frame linear acceleration, and projected gravity:

$$z_t^{\text{impl}} = \left(\boldsymbol{\omega}_t^{\mathcal{B}}, \boldsymbol{a}_t^{\mathcal{B}}, \boldsymbol{g}_t^{\text{proj},\mathcal{B}}\right) \in \mathbb{R}^9,$$

where $\mathcal{W}$ and $\mathcal{B}$ denote the world and base frames, respectively. The implemented task substate preserves all the platform tilt, linear acceleration, and tilt-rate information in the definition in Section 3.1. Using the ZYX Euler-angle convention, let $R = R_z(\theta_y)R_y(\theta_p)R_x(\theta_r)$ map the base frame to the world frame. With normalized projected gravity

$$\boldsymbol{u} = \frac{\boldsymbol{g}^{\text{proj},\mathcal{B}}}{\|\boldsymbol{g}^{\text{proj},\mathcal{B}}\|} = R^\top (0, 0, -1)^\top,$$

the conversion is,

$$\begin{aligned} \theta_p &= \operatorname{atan2}\left(u_x, \sqrt{u_y^2 + u_z^2}\right), \\ \theta_r &= \operatorname{atan2}(-u_y, -u_z), \\ \begin{bmatrix} \dot{\theta}_p \\ \dot{\theta}_r \end{bmatrix} &= \begin{bmatrix} 0 & \cos\theta_r & -\sin\theta_r \\ 1 & \sin\theta_r \tan\theta_p & \cos\theta_r \tan\theta_p \end{bmatrix} \boldsymbol{\omega}^{\mathcal{B}}, \\ \boldsymbol{a}^{\mathcal{W}} &= R\boldsymbol{a}^{\mathcal{B}}. \end{aligned} \tag{6}$$

For $|\theta_p| < \pi/2$, these relations recover $z_t = [\boldsymbol{\theta}_t, \boldsymbol{a}_t, \dot{\boldsymbol{\theta}}_t]$ in the main formulation, establishing equivalence for the task quantities.

A.3 High-Precision Objective.

We use the task substate from the main formulation, $z_t = [\boldsymbol{\theta}, \boldsymbol{a}, \boldsymbol{\omega}]$, where $\boldsymbol{\theta} = [\theta_p, \theta_r]$, $\boldsymbol{a} = [a_x, a_y, a_z]$, and $\boldsymbol{\omega} = \dot{\boldsymbol{\theta}}$. To construct the normalized high-precision target in Equation (1), consider a translational approximation at the platform reference point, with the world vertical axis pointing

upward. Neglecting local inertial terms induced by platform rotation, gravity and the inertial reaction produce the effective load per unit mass

$$\boldsymbol{f}_{\text{eff}} = -g\boldsymbol{e}_z - \boldsymbol{a} = [-a_x \quad -a_y \quad -(g + a_z)]^\top .$$

For fixed horizontal acceleration, the steady precision target eliminates vertical acceleration and orientation rates, giving $a_z^* = 0$ and $\boldsymbol{\omega}^* = \mathbf{0}$. At this target, the effective-load magnitude is $D = \sqrt{a_x^2 + a_y^2 + g^2}$, and its normalized horizontal components are $[-a_x/D, -a_y/D]^\top$. The componentwise target mapping in the main formulation sets the tangents of pitch and roll to these normalized components:

$$\begin{bmatrix} \tan\theta_p^* \\ \tan\theta_r^* \end{bmatrix} = -\frac{1}{D}\begin{bmatrix} a_x \\ a_y \end{bmatrix}.$$

Taking the componentwise inverse tangent and combining the steady-state conditions yields exactly the high-precision target in Equation (1):

$$\begin{bmatrix} \theta_p^* \\ \theta_r^* \end{bmatrix} = \arctan\left(-\frac{1}{\sqrt{a_x^2 + a_y^2 + g^2}} \begin{bmatrix} a_x \\ a_y \end{bmatrix} \right), \qquad [\boldsymbol{\omega}^*, a_z^*] = \mathbf{0}.$$

When $a_x = a_y = 0$, the target reduces to a level platform; nonzero horizontal acceleration produces a bounded tilt target through the normalized mapping.

B Implementation, Training, and Evaluation Details

All policies are trained in Isaac Lab at a 5 ms physics step (200 Hz) and a control decimation of 4, giving a 50 Hz control loop; each episode lasts 20 s. The robot is a Unitree Go2-W (12 leg joints, 4 wheels) with the stabilization platform rigidly mounted on the base. Leg joints use position control (K_p=50, K_d=1) and wheels velocity control (K_p=0, K_d=0.5); both share a 23.5 N m effort limit, a 30 rad/s velocity limit.

B.1 Observations

We use an asymmetric actor–critic. The policy observes only deployable proprioception with an L-step history (L=6), $s_t^{prop} = [\, v_c, \omega, g, q{-}q_0, \dot{q}]_t$; the critic additionally observes privileged quantities. The world model predicts the task substate, emitting the chunk $\hat{z}_{t+1:t+H}$ in one forward pass. Additive uniform noise is applied to the policy group only.

Table 2: Observation terms. Group: P = policy, C = critic, W = world-model target.

Term	Symbol	Group	Dim	Hist.	Noise	Scale
Velocity command	v_c	P/C	3	6	–	1.0
Base angular velocity	ω	P/C/W	3	6	$\pm$0.2	0.25
Projected gravity	g	P/C/W	3	6	$\pm$0.05	1.0
Joint positions (legs)	$q{-}q_0$	P/C	12	6	$\pm$0.01	1.0
Joint velocities	$\dot{q}$	P/C	16	6	$\pm$1.5	0.05
Previous action	a_{t-1}	P/C	16	1	–	1.0
Base linear velocity	v	C	3	1	–	2.0
Terrain height scan	–	C	$\approx$187	1	–	1.0
Base linear accel. (Stage 2)	a^{base}	C/W	3	6	–	0.25
Ideal projected gravity (Stage 2)	$g^\star$	C	3	6	–	1.0

B.2 World Model Prediction

Each prediction sample consists of the policy observation, the current base-policy action, and up to H consecutive future substates from the same environment. For a sample at rollout index t, the label at prediction offset $k \in \{0, \ldots, H-1\}$ is valid only when $t + k < T$ and $\prod_{j=0}^{k}(1 - d_{t+j}) = 1$, where T is the rollout length and d_t indicates episode termination or truncation. This mask excludes

post-reset states from the future labels of pre-reset samples. Samples near an episode boundary retain their valid prediction prefix, with the remaining offsets excluded from the loss. Future sequences are constructed within each rollout batch without stitching across batches; offsets beyond the batch boundary are zero-filled and masked out. Observation histories persist across rollout batches and are reset at episode boundaries. When the available history is insufficient, missing entries are filled by repeating the first observation of the new episode.

The acceleration signal used to construct the desired orientation is processed by a first-order exponential IIR low-pass filter, $y_t = \alpha x_t + (1-\alpha) y_{t-1}$, with $\alpha = 1 - \exp(-2\pi f_c / f_s)$. The configured cutoff frequency is $f_c = 1$ Hz and the control frequency is $f_s = 50$ Hz. Filtering precedes a componentwise hard dead band: horizontal acceleration components with magnitude below 0.1 m/s^2 are set to zero. The resulting components are multiplied by 1.5 before constructing the desired gravity direction. The filter is initialized from the first raw sample. Reset-marked calls reinitialize the filter only when they bypass its same-step cache; under the current reward-before-reset evaluation order, automatic episode resets retain the cached filter state.

B.3 Rewards

Table 3 lists the reward terms for both stages (σ^2=0.25). Stage 2 drops the keep-level terms (flat orientation, base height) in favor of a projected-gravity tracking reward, so the platform banks into accelerations rather than staying flat. The ideal projected gravity $g^\star$ is constructed from the low-pass-filtered (2 Hz) horizontal base acceleration with a 0.1 m/s^2 dead-band.

Table 3: Reward terms across the two training stages. A single value is shared by both stages.

Component	Expression	Weight (S1 / S2)
Lin. vel. tracking (x)	$\exp(-(v_x^{\mathrm{cmd}} - v_x)^2/\sigma^2)$	0.75 / 1.0
Lin. vel. tracking (y)	$\exp(-(v_y^{\mathrm{cmd}} - v_y)^2/\sigma^2)$	0.75
Ang. vel. tracking (z)	$\exp(-(\omega_z^{\mathrm{cmd}} - \omega_z)^2/\sigma^2)$	0.75
Base angle tracking	$\exp(-\lVert g_{xy} - g^\star_{xy}\rVert^2/\sigma^2)$	– / 0.5
Vertical velocity	v_z^2	-2.0 / -10.0
Roll/pitch rate	$\omega_x^2 + \omega_y^2$	-0.05
Flat orientation	$g_x^2 + g_y^2$	-0.5 / –
Joint torques	$\sum_j \tau_j^2$	-2.0e-4
Leg joint accel.	$\sum_{\mathrm{leg}} \ddot{q}^2$	-2.5e-7
Wheel joint accel.	$\sum_{\mathrm{wheel}} \ddot{q}^2$	-2.5e-9
Base height	$(h^{base} - 0.40)^2$	-10.0
Undesired contacts	$\sum \mathbb{1}(\text{non-foot contacts})$	-1.0
Action rate	$\sum (a_t - a_{t-1})^2$	-0.01
Stand-still (no cmd)	$\mathbb{1}[\lVert v_c\rVert<0.1] \sum_{\mathrm{leg}} \lvert q - q_0\rvert$	-0.25
Hip deviation	$\sum_{\mathrm{hip}} (q - q_0)^2$	-0.3
Joint deviation (legs)	$\sum_{\mathrm{leg}} (q - q_0)^2$	-0.1

B.4 Domain Randomization

Table 4 lists all randomization. Operations are *add* (offset), *scale* (multiplicative), or *abs* (absolute), applied at startup (once per environment), reset (each episode), or on a recurring interval. The delayed-PD model lags the applied command by 0–3 physics steps (0–15 ms).

B.5 Curriculum Learning

Terrain Curriculum follows the standard velocity-based progression: an environment is promoted when it walks past half the terrain length and demoted when it covers less than half its commanded distance.

Command Curriculum is per-environment and per-axis. Each axis $i \in \{x, y, z\}$ holds a level $\ell_i \in [0.1, 1.0]$ that scales the full command range to $[\ell_i v_i^{\min}, \ell_i v_i^{\max}]$. Each environment maintains

Table 4: Domain randomization and perturbations. *log-uniform.

Term	Min	Max	Unit	Op.	When
Base mass	-1.0	2.0	kg	add	startup
Body inertia	0.5	1.5	$\times$	scale	startup
Base CoM offset (x, y, z)	-0.05	0.05	m	add	startup
Foot static friction	0.5	1.0	–	abs	startup
Foot dynamic friction	0.5	0.8	–	abs	startup
Foot restitution	0.0	0.5	–	abs	startup
Actuator stiffness K_p	0.8	1.2	$\times$	scale*	reset
Actuator damping K_d	0.8	1.2	$\times$	scale*	reset
Initial joint positions	0.95	1.05	$\times$	scale	reset
Initial base yaw	$-\pi$	π	rad	abs	reset
Initial base height	0.0	0.2	m	add	reset
Initial base lin. vel. (x)	-0.5	0.5	m/s	abs	reset
Initial base lin. vel. (y)	-0.15	0.15	m/s	abs	reset
Initial base lin. vel. (z)	-0.2	0.2	m/s	abs	reset
Init. base ang. vel. (r/p/y)	-0.35	0.35	rad/s	abs	reset
Reset impulse force / torque	-10	10	N, N m	abs	reset
Push velocity (x)	-0.5	0.5	m/s	add	interval
Push velocity (y)	-0.3	0.3	m/s	add	interval

an exponential moving average (EMA) of its per-episode tracking-reward rate,

$$\bar{r}_i \leftarrow (1-\alpha)\bar{r}_i + \alpha\left(\sum r_i / T_{\text{ep}}\right),$$

where T_{ep} is the episode length and $\alpha = 0.5$. At every reset, the command range ℓ_i is adjusted by $\pm\Delta$ ($\Delta = 0.1$) according to this EMA. The range is expanded if $\bar{r}_i > \tau_{\uparrow} w_i$ and shrunk if $\bar{r}_i < \tau_{\downarrow} w_i$, where w_i is the tracking weight and $(\tau_{\uparrow}, \tau_{\downarrow}) = (0.8, 0.5)$ for the linear axes and $(0.6, 0.3)$ for yaw. As a result, each environment adapts its difficulty independently, and only an environment that consistently tracks commands within its current range is assigned a wider one.

Command Resampling. At each command resampling (6–10 s; the command is held at 0 for the first 50 steps of every episode), environments are divided into three regimes. With probability 10%, the robot receives a **zero command** for standing. With probability 5%, it receives a **bang–bang** step command sampled from the corners of the command box (each axis at min/0/max, excluding the all-zero command) to encourage recovery from saturated transients. The remaining 85% of environments use **bucket sampling**.

When the command range of an axis has just expanded ($\ell_i > \ell_i^{\text{prev}}$), samples are drawn from three bins with probabilities $(0.15, 0.70, 0.15)$: the newly opened negative frontier, the previously mastered core, and the newly opened positive frontier. Otherwise, commands are sampled uniformly from the current range. This strategy allocates 30% of samples to the newly expanded boundaries while keeping 70% within the previously mastered region, allowing the curriculum to expand progressively without causing a sharp drop in tracking performance.

B.6 Network Architecture and Optimization

Network architecture. The base policy and world model are MLPs with hidden widths $(512, 256, 128)$, while the residual adapter uses $(256, 128)$. All three networks use ELU hidden activations and linear output layers. Each training stage uses a separate critic with hidden widths $(512, 256, 128)$ and the privileged observations listed in Table 2. The adapter's output layer is initialized to zero.

Optimization. In Stage 1, PPO trains the base policy and critic, while a separate Adam optimizer trains the world model on the same rollouts using the prediction loss. Prediction gradients do not update the base policy. In Stage 2, the base policy and world model are frozen; PPO updates the residual adapter and a newly initialized critic. Both stages use the PPO settings in Table 5.

Training and deployment. Training uses Isaac Lab with a physics step of 5 ms, four physics steps per policy update, and a maximum episode duration of 20 s. The deployment controller is configured

Table 5: PPO hyperparameters used in both training stages.

Hyperparameter	Value
Optimizer	Adam
Initial learning rate	10^{-3}
Learning-rate schedule	Adaptive, target KL 0.01
Rollout length per environment	24
Epochs per update	5
Discount factor	0.99
GAE parameter	0.95
Clipping parameter	0.2
Entropy coefficient	0.01
Value-loss coefficient	1.0
Clipped value loss	Enabled
Maximum gradient norm	1.0
Initial action standard deviation	1.0

to run at 50 Hz, corresponding to a nominal prediction horizon of 100 ms. All network parameters remain fixed during deployment.

B.7 Metric Definitions and Aggregation

For each trial i, let $\mathcal{T}_i$ denote the valid measurement samples and $T_i = |\mathcal{T}_i|$. For a scalar signal $x_{i,t}$, define its root mean square (RMS), peak magnitude, and mean absolute magnitude as

$$R_i(x) = \sqrt{\frac{1}{T_i}\sum_{t\in\mathcal{T}_i} x_{i,t}^2}, \qquad P_i(x) = \max_{t\in\mathcal{T}_i} |x_{i,t}|, \qquad A_i(x) = \frac{1}{T_i}\sum_{t\in\mathcal{T}_i} |x_{i,t}|. \tag{7}$$

Platform angle error. For pitch or roll, indexed by $q \in \{p, r\}$, the angular error is

$$e_{q,i,t} = \mathrm{wrap}_{[-\pi,\pi)}\left(\theta_{q,i,t} - \theta^\star_{q,i,t}\right), \tag{8}$$

where $\theta^\star_{q,i,t}$ is the desired orientation. The terrain leveling task uses a horizontal target, $\theta^\star_{p,i,t} = \theta^\star_{r,i,t} = 0$. For the acceleration compensation task, the target depends on acceleration, as specified in Equation (1) and Appendix A.

Vertical acceleration. Let $a_{z,i,t}$ denote translational acceleration along the world vertical axis at the evaluated platform or base reference point. Its trial-level mean and peak magnitudes are

$$\bar{a}_{z,i} = A_i(a_z), \qquad a^{\mathrm{peak}}_{z,i} = P_i(a_z), \tag{9}$$

both expressed in $\mathrm{m/s^2}$.

Command tracking error. For forward-speed tracking, the discrepancy between measured and commanded velocities is

$$e^{\mathrm{track}} = R_i\big(v_{i,t} - v^{\mathrm{cmd}}_{i,t}\big), \tag{10}$$

where both velocities are expressed in the same coordinate frame.

Cross-trial aggregation. Let $\mathcal{V} = \{i : T_i > 0\}$ denote trials with valid measurement samples. We compute each continuous-signal metric separately for each trial and report its mean across trials:

$$\bar{m} = \frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}} m_i, \tag{11}$$

where m_i is the corresponding trial-level metric.

In particular, peak metrics are obtained by first taking the maximum magnitude within each trial and then averaging these trial-level peaks:

$$\overline{P}(x) = \frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}} \max_{t\in\mathcal{T}_i} |x_{i,t}|. \tag{12}$$

Columns labeled "max" in Table 1 report this mean of per-trial peak magnitudes. The same aggregation applies to the pitch and roll peak errors and the vertical-acceleration peaks.

C Additional Experimental Results

C.1 Simulation Results

C.1.1 Terrain Leveling

Figure 9 shows velocity tracking and orientation responses during terrain leveling with LocoWM. The forward velocity command increases from 0 to 1 m/s at approximately 2 s and returns to 0 at approximately 19 s. The measured velocity fluctuates around the target during traversal and returns close to zero after the stop command. The corresponding pitch and roll traces show orientation variations within a few degrees, with larger excursions in pitch than in roll.

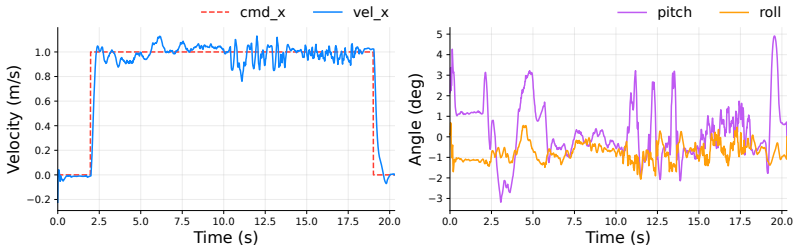
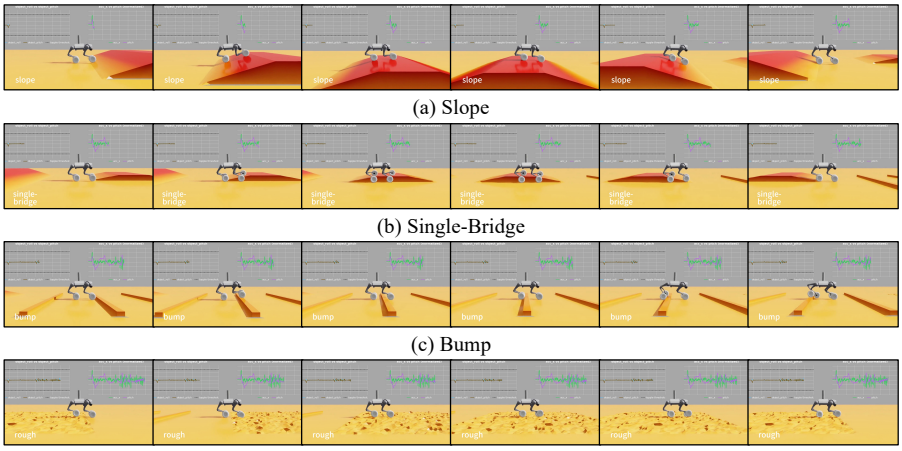


Figure 9: Forward velocity tracking and pitch and roll responses during simulated terrain leveling.

Figure 10 presents successive snapshots of payload transport under different support conditions. On slopes, the front and rear wheels pass through the slope entry, crest, and descent in sequence, while the leg configuration adjusts to changing support heights and the platform remains approximately level. The one-sided bridge creates a lateral difference in support height, requiring an asymmetric leg configuration to regulate roll. In both scenes, the payload remains upright throughout the displayed traversal.

Bumps and rough terrain illustrate responses to repeated changes in contact conditions. Crossing a bump produces distinct support transitions as the front and rear wheels pass over the obstacle, whereas rough terrain introduces continuing irregular disturbances. The inset torso acceleration and pitch traces fluctuate during these maneuvers, while payload roll and pitch remain within the indicated toppling thresholds. These sequences complement the aggregate results in Table 1 by illustrating how LocoWM accommodates local support changes while retaining an upright payload.



(a) Slope

(b) Single-Bridge

(c) Bump

(d) Rough

Figure 10: Simulated terrain leveling with an upright payload.

C.1.2 Acceleration Compensation

We examine how LocoWM adjusts platform pitch during acceleration and braking on flat ground using step and triangular-wave velocity commands. In Figure 11, the commanded forward velocity

follows the sequence $0 \to 1 \to -1 \to 1 \to 0$ m/s, and the measured velocity reaches each plateau after a brief transition. Each command change produces an acceleration transient accompanied by a pitch response of the same sign. Pitch returns close to zero as acceleration subsides, including during steady backward motion. This response links platform tilt to acceleration while maintaining an approximately level posture at constant velocity.

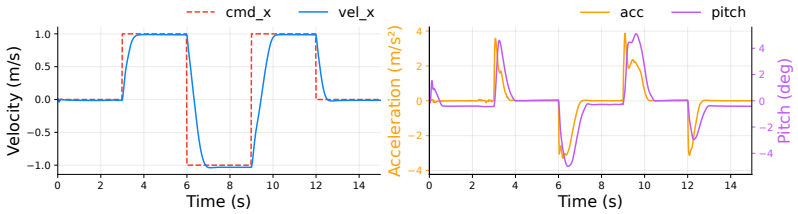


Figure 11: Simulated acceleration compensation under step velocity commands.

Figure 12 shows the response to triangular-wave commands spanning -1 to 1 m/s. The measured velocity follows the rising and falling ramps with a delay near their turning points. During each ramp, sustained acceleration is accompanied by sustained platform tilt, with larger pitch magnitudes on the steeper ramps. When the command slope changes sign, acceleration reverses and pitch subsequently transitions to the corresponding sign.

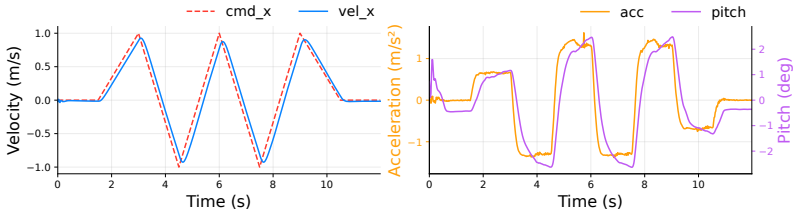
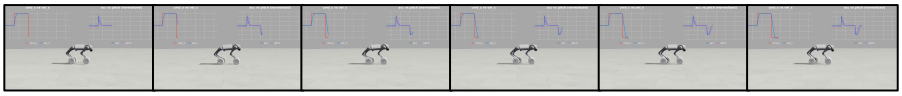
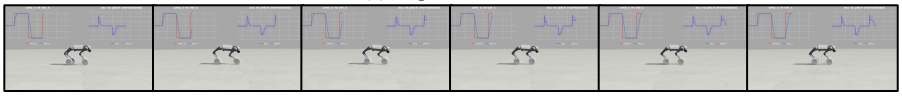


Figure 12: Simulated acceleration compensation under triangular-wave velocity commands.

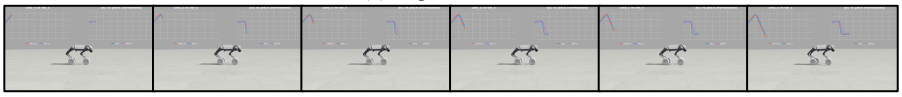
The snapshots in Figure 13 show the leg configurations and torso poses associated with these responses. Under step commands, the torso tilts during velocity transitions and returns toward level as velocity settles. Under triangular-wave commands, the torso maintains a tilt during the ramps and changes its orientation as acceleration reverses. Together, the curves and snapshots illustrate transient and sustained pitch adjustments for acceleration compensation.



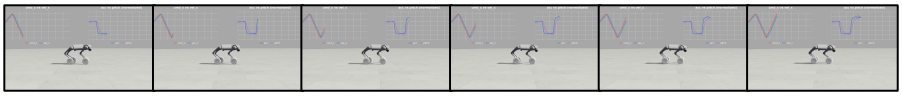
(a) Step-acceleration



(b) Step-deceleration



(c) Triangle-acceleration



(d) Triangle-deceleration

Figure 13: Simulated posture sequences during acceleration compensation under step commands in (a,b) and triangular-wave commands in (c,d). Insets show velocity tracking and normalized acceleration and pitch traces.

C.1.3 PUSH RECOVERY

Figure 14 shows LocoWM's velocity and payload-orientation responses to repeated external pushes during forward locomotion. After startup, the forward velocity command remains at 0.5 m/s, while the disturbances produce transient increases and decreases in the measured velocity. Velocity returns close to the command after each disturbance. Payload pitch and roll undergo brief excursions of a few degrees and settle near zero between pushes, showing recovery of the payload's upright orientation alongside command tracking.

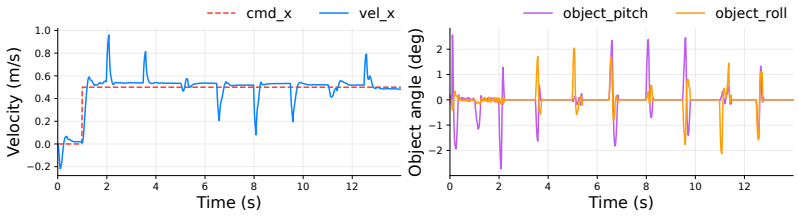
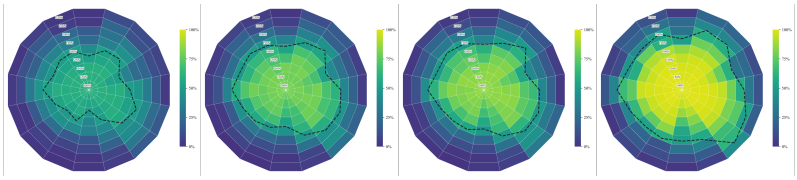


Figure 14: Simulated push recovery: forward velocity tracking (left) and payload pitch and roll (right).

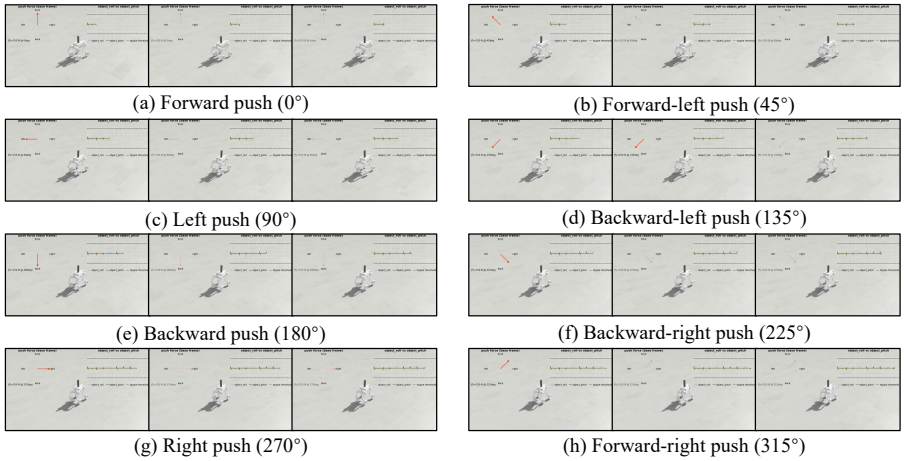
Figure 15 compares payload-retention success across push directions and force magnitudes for End-to-End, React, Recon, and LocoWM. These maps complement the recovery sequences by showing how payload retention changes as disturbance strength and direction vary.



(a) E2E (b) React (c) Recon (d) LocoWM

Figure 15: Payload-retention success-rate maps for (a) End-to-End (E2E), (b) React, (c) Recon, and (d) LocoWM. Polar angle denotes push direction, radius denotes force magnitude, and color denotes success rate. Dashed black contours mark a 50% success rate.

The sequences in Figure 16 show recovery from 150 N pushes applied in eight directions, spaced at 45° intervals in the body frame. Each row follows force application and the subsequent recovery, with changes in the robot's posture and leg configuration as it responds to the disturbance. The payload remains on the platform in all eight displayed sequences, and the inset orientation traces remain within the indicated toppling thresholds.

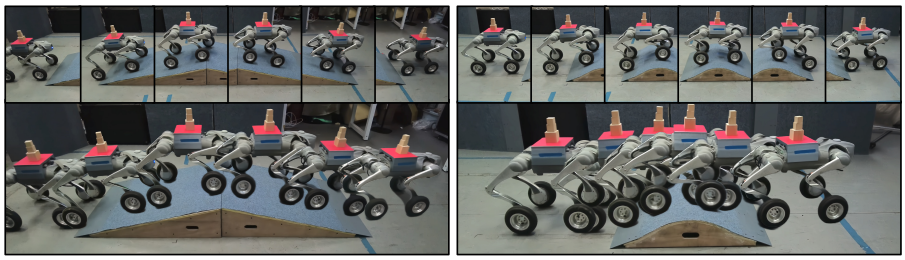


(a) Forward push (0°) (b) Forward-left push (45°)
(c) Left push (90°) (d) Backward-left push (135°)
(e) Backward push (180°) (f) Backward-right push (225°)
(g) Right push (270°) (h) Forward-right push (315°)

Figure 16: Simulated push recovery under 150 N forces applied in eight body-frame directions. Frames progress from left to right through force application and recovery. Arrows indicate push direction, and inset traces show payload roll and pitch with the toppling thresholds.

C.2 Real-World Demonstrations

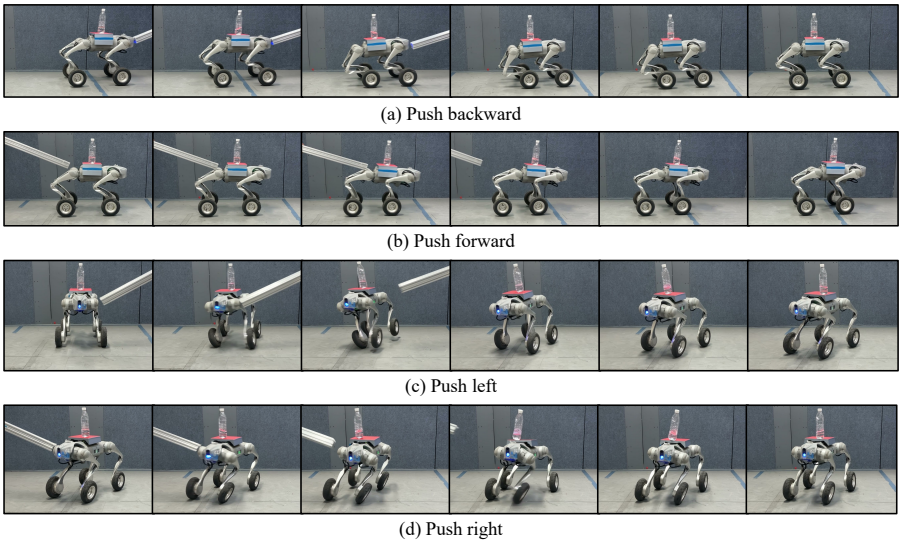
Terrain Leveling. Complementing the one-sided bridge and bump experiments in the main text, Figure 17 shows transport of unsecured stacked blocks over a slope and a wave-shaped obstacle. LocoWM adjusts the leg configuration to changing support conditions, keeping the back-mounted platform approximately level in the displayed sequences. Near the crest and obstacle exits, the front and rear wheels contact different terrain segments while the platform orientation remains distinct from the local ground inclination. The successive video frames record the traversal, and the overlaid views show how wheel–leg configurations and payload orientation evolve along the path. The block stack remains upright in both sequences, illustrating hardware leveling and payload retention under changing support heights.



(a) Slope (b) Wave

Figure 17: Real-world terrain leveling with stacked blocks on (a) a slope and (b) a wave-shaped obstacle. Each panel shows successive video frames above and robot cutouts overlaid on a common background below.

Push Recovery. Figure 18 shows responses to backward, forward, leftward, and rightward pushes applied to the torso with a pole while an unsecured bottle rests on the platform. The side and frontal views capture the resulting pitch and roll disturbances, respectively. LocoWM adjusts body posture and leg configuration during the disturbance and brings the platform back toward level after the push. The bottle tilts transiently but remains on the platform without toppling in all four displayed sequences.



(a) Push backward

(b) Push forward

(c) Push left

(d) Push right

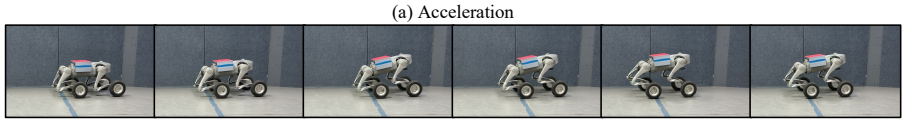
Figure 18: Real-world push recovery under (a) backward, (b) forward, (c) leftward, and (d) rightward pushes. Each row progresses from force application to recovery, with the unsecured bottle retained on the platform.

Acceleration Compensation. Figure 19 supplements the hardware demonstrations in the main text with separate posture sequences during acceleration and deceleration on flat ground. During acceleration, the torso adjusts its pitch as the motion changes and returns toward level in the later frames. During deceleration, changes in the front and rear leg configurations accompany a corresponding

adjustment in platform tilt. Because the platform is rigidly attached to the torso, these orientation changes are achieved through the robot's own wheel–leg motion. Both sequences illustrate active platform tilting during changes in longitudinal motion, consistent with the acceleration compensation behavior observed in simulation.



(a) Acceleration



(b) Deceleration

Figure 19: Real-world acceleration compensation during (a) acceleration and (b) deceleration on flat ground. Successive frames from left to right show changes in leg configuration and platform pitch.

D Humanoid Transport

D.1 Experimental Setup

Following the SteadyTray task definition (Huang et al., 2026), we evaluate *bimanual humanoid tray transport* in Isaac Lab with a Unitree G1 humanoid. The G1 has 29 actuated joints, and both hands support a tray carrying an unsecured cylindrical payload. The tray measures 25.4×35.2 cm, while the payload radius and height are randomized within $[2.1, 4.5]$ cm and $[7.5, 20]$ cm, respectively.

Task. We evaluate three scenarios: *Command Track*, *Push Robot*, and *Push Object*. The velocity command is zero during the first second, switches to a sampled target during 1–10 s, and switches once more at 10 s. For the two push scenarios, a constant world-frame XY force is applied for 0.1 s at 5, 10, and 15 s. The nominal per-axis velocity-change bound is 0.5 m/s for Push Robot and 0.3 m/s for Push Object, while Command Track has no external push.

Baselines. The evaluation compares three executable controller paths: Base-Policy, ReST-RL, and LocoWM. Base-Policy denotes the deterministic output of the frozen whole-body locomotion policy. ReST-RL uses the deployable residual branch from the public checkpoint. The following table therefore reports a reconstruction, rather than an exact reproduction of every method row in the original Table II in (Huang et al., 2026).

Metrics. TrackLinErr is the L_2 error between commanded and measured horizontal velocity in the torso frame. Grav-XY is the norm of the XY components of unit gravity expressed in the object frame. A trial is successful only if the object remains above 0.7 m and its tilt remains below 0.7 rad throughout the full horizon.

D.2 Comparison Results

Table 6 shows that both ReST-RL and LocoWM achieve higher Success Rate than Base-Policy across all three tasks. LocoWM achieves the highest Success Rate of 94.1%, 82.8%, and 92.6% in Command Track, Push Robot, and Push Object, exceeding ReST-RL by 3.1, 1.9, and 2.4 percentage points, respectively. LocoWM also achieves the lowest mean TrackLinErr and Grav-XY in both Command Track and Push Object. In Push Robot, ReST-RL attains slightly lower mean TrackLinErr and Grav-XY, whereas LocoWM achieves a higher Success Rate, increasing it from 12.5% with Base-Policy to 82.8%. These results show that LocoWM achieves higher transport success across the evaluated scenarios, although it does not attain the lowest mean tracking error and payload tilt metric in every scenario.

D.3 Visualization

Figure 20 shows two walking intervals separated by a stop, with the forward velocity command switching between 0 and 1 m/s. The measured velocity rises after each start command, oscillates

Task	Method	TrackLinErr (m/s) ↓		Grav-XY ↓		Success Rate (%) ↑
		mean	std	mean	std	
Command Track	Base-Policy	0.128	0.096	0.498	0.431	52.7
	ReST-RL	0.123	0.108	0.083	0.212	91.0
	LocoWM	**0.120**	0.102	**0.063**	0.165	**94.1**
Push Robot	Base-Policy	0.161	0.162	0.751	0.388	12.5
	ReST-RL	**0.150**	0.148	**0.128**	0.281	80.9
	LocoWM	0.155	0.165	0.129	0.280	**82.8**
Push Object	Base-Policy	0.130	0.105	0.567	0.435	44.1
	ReST-RL	0.123	0.107	0.082	0.209	90.2
	LocoWM	**0.122**	0.112	**0.064**	0.163	**92.6**

Table 6: Simulation results for humanoid tray transport. Bold entries mark the best mean error or success rate within each task.

during walking, and returns close to zero after each stop command. Payload pitch and roll also exhibit periodic oscillations during walking. These oscillations subside after stopping as the payload settles.

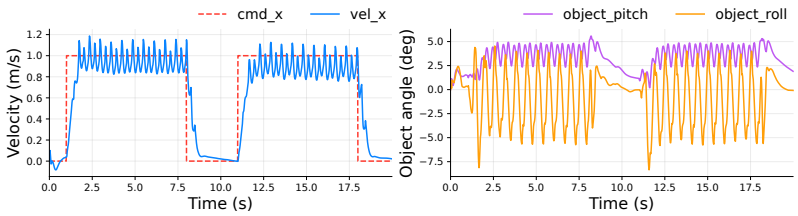
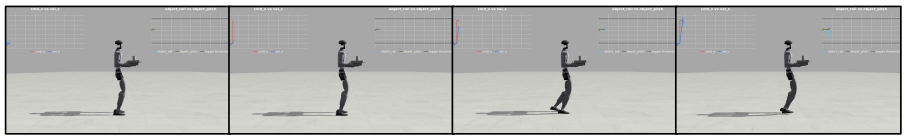
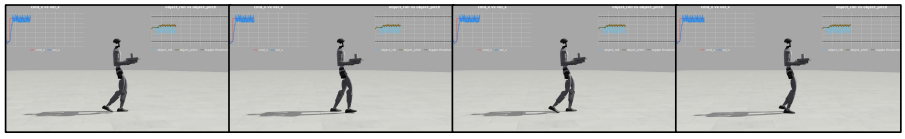


Figure 20: Simulated humanoid transport during starting, walking, and stopping.

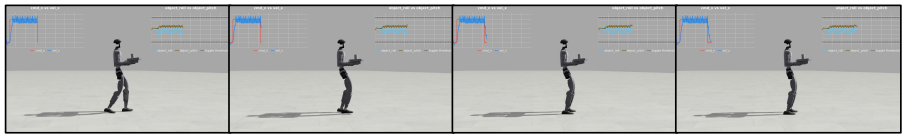
The snapshots in Figure 21 show the corresponding transitions from standing to stepping, continued forward walking, and a return to standing. The robot alternates leg support while holding the tray in front of its torso, exposing the payload to changing body motion and contact conditions. The payload remains on the tray without toppling throughout the displayed sequences. This example complements the Go2-W experiments by applying preactive residual control to end-effector stabilization during bipedal locomotion.



(a) Start



(b) Forward



(c) Stop

Figure 21: Simulated humanoid tray transport during (a) starting, (b) forward walking, and (c) stopping. Frames progress from left to right, with insets showing velocity tracking and payload orientation.