ForPC: Force-Aware Whole-Body Control for Precise Humanoid Cart Pushing

Project Website: `https://roboforpc.github.io/`

Abstract

Humanoid robots are expected to take over strenuous material handling in manufacturing, logistics, and healthcare, where operating existing wheeled carts would let them move heavy loads without modifying the workplace. However, humanoid cart pushing remains challenging because the bimanual grasp closes a force-coupled loop in which the contact wrenches that drive the cart also disturb the robot's balance, and payload changes alter the coupled dynamics. We propose ForPC, a force-aware whole-body control framework that regulates cart velocity through two-stage reinforcement learning. A locomotion prior is first trained under varied hand loads and upper-body postures to provide balance support. A force latent and a state latent are then inferred from onboard observations, without external force sensors, and fed to a residual policy that coordinates bounded arm, waist, and leg adjustments with base-velocity commands. Multi-objective rewards penalize bimanual internal opposition, grasp slip, and actuator saturation, encouraging the policy to apply contact force on demand without human demonstrations. Sim-to-sim experiments show that ForPC reduces cart-velocity RMSE by up to 68.2% relative to a locomotion-only baseline (Loco-only) and cross-track RMSE by up to 74.2% on the multi-bend route. Trained solely in simulation, the policy also transfers to a physical Unitree G1 without real-world training or fine-tuning and, with no payload-specific adjustment, pushes a cart carrying up to 30 kg at 0.5 m/s. These results suggest that humanoid robots could eventually replace humans for efficient material handling in daily life.

Keywords: Humanoid robot, Cart pushing, Whole-body control, Dynamic coupling, Interaction-force estimation, Residual reinforcement learning

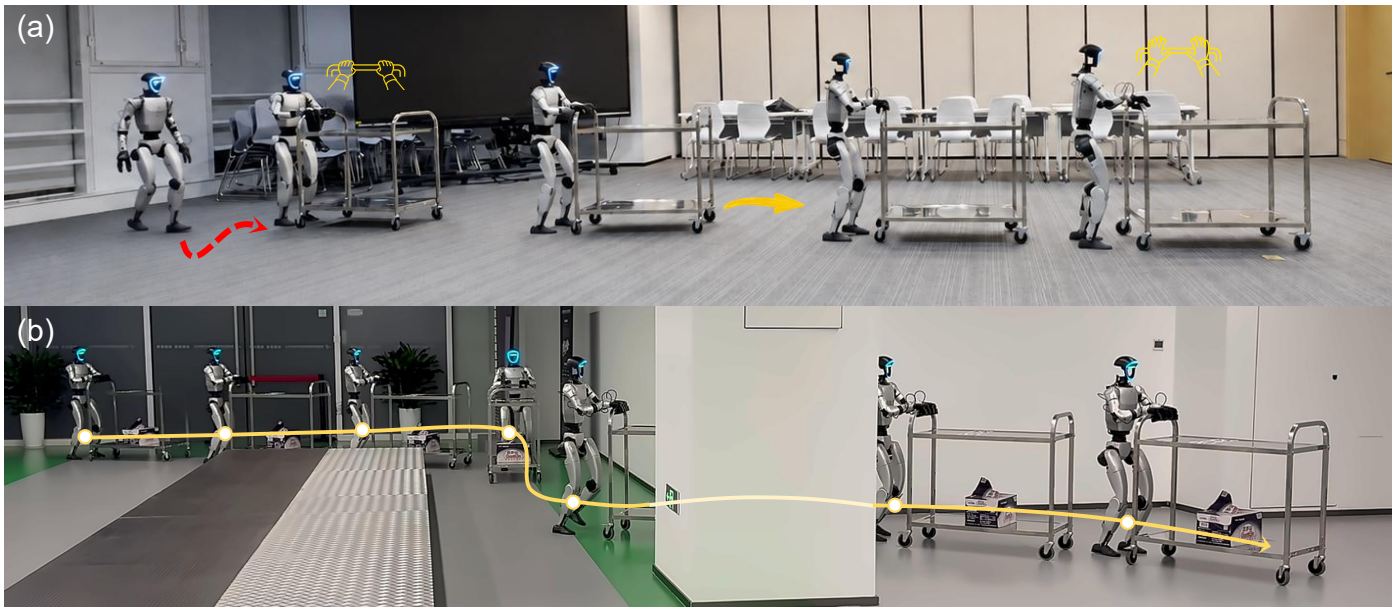


Figure 1: (a) Complete cart-pushing sequence performed by a Unitree G1 humanoid, comprising cart grasping, pushing, and release. (b) A Unitree G1 humanoid pushing the cart along a complex path.

1. Introduction

Humanoid robots possess an anthropomorphic morphology and whole-body dexterity, enabling them to directly operate human-centric equipment, including manual wheeled carts designed for human workers [1–3]. By autonomously taking over repetitive and strenuous hauling tasks, humanoid robots could enhance operational productivity and transport efficiency, and thus hold broad application prospects.

Despite early progress in humanoid cart transport [4, 5], achieving precise and dependable pushing remains challenging. Sustained contact establishes a bilaterally coupled physical system, in which hand forces drive the cart while its motion inertia and rolling resistance transmit reaction loads back to the robot and disrupt its locomotion balance. Under payload variations, the robot must accurately regulate both the magnitude and the direction of its contact effort to maintain a steady heading and forward cruise. Grasping the same handle makes

uncoordinated arms prone to working against each other. This mutual opposition produces internal forces that do not contribute to propulsion [6] and consumes the joint torque margin needed for heading corrections and steering maneuvers.

Model-based formulations coordinate motion and force tasks with balance objectives but require explicit models of the contact and the object [7–9], whereas learning-based methods obtain whole-body coordination from simulated interaction, demonstrations, or motion tracking [10–13]. Recent controllers further accommodate manipulation loads [14–16] or infer contact forces and object motion from the robot's own signals [17–21]. However, Portela et al. [17] and Zhi et al. [18] estimate contact forces without object motion, whereas HAIC [19] predicts object motion without an explicit hand-force estimate, so none of them closes the feedback loop on both the cart velocity and the bimanual hand wrenches. Without such feedback, a policy can neither assess the cart's actual dynamic response nor detect whether the two arms are fighting against each other on the handle. Although wrist force/torque sensors could provide direct feedback, such sensing is not always available for control, so recent methods estimate these forces from proprioception [17, 22]. Precise cart pushing under varying payloads, which requires feedback on both the cart motion and the hand forces without external force sensors, therefore remains an unresolved challenge.

Addressing this challenge requires treating the humanoid and the cart as a tightly coupled dynamic system rather than decoupled motion primitives. Effective payload transport depends on closing the feedback loop around the physical interaction, so the controller must perceive both the cart's real-time motion response and the contact loads at the hands. Because everyday carts have no built-in motion sensors and the hands provide no direct force measurement, these quantities must be inferred from signals available on the robot itself.

We propose ForPC, a force-aware whole-body control framework that uses two-stage reinforcement learning for dynamic cart pushing, as shown in Figure 3. In the first stage, an interaction-resilient locomotion prior is pre-trained under severe, randomized external hand disturbances and bounded posture offsets, and is then frozen so that the task policy can reuse its support response. In the second stage, a teacher–student estimator infers force and state latents from histories of joint kinematics, upper-body torques, and the visually observed relative cart heading, and the task policy uses these latents to output bounded waist–arm and leg joint residuals and modulate robot base velocity commands. Without relying on external force sensors or human demonstrations, our framework effectively coordinates bimanual effort and transfers zero-shot across varied payloads. We evaluate ForPC against a locomotion-only baseline (Loco-only) in Isaac Sim and MuJoCo on velocity tracking, force estimation, and route following, and then deploy the policy, trained solely in Isaac Sim, zero-shot on a physical Unitree G1 humanoid to accomplish the transport of parts and workpieces along complex paths (Figure 1).

The main contributions of this paper are threefold.

- We propose ForPC, a two-stage whole-body control framework for coupled humanoid cart pushing that combines a locomotion prior, interaction estimators, and a residual policy.
- We design a force-guided interaction inference module that infers a force latent from onboard signals and uses it to guide the estimation of a state latent, allowing the policy to perceive contact loading and cart motion without external force sensors.
- We transfer our policy zero-shot to a real Unitree G1 humanoid, which achieves precise cart-pushing control along complex paths under a 30 kg payload.

2. Related Work

2.1. Humanoid Pushing and Object Transport

Early humanoid pushing relied on dual-arm force control [23], controllers designed for a rolling cart [4], and joint planning of centre-of-mass motion and manipulation forces [24], which require force measurements or explicit object models. Learning-based systems have since substantially increased the load that a humanoid can move. Xiao et al. [16] push and pull a cart with a total load of 112.8 kg under commanded velocities and forces, Li et al. [25] use MPC-guided rewards to push a cart payload of up to 290 kg without runtime object feedback, and HULK [26] performs cart pushing and pulling at 150 kg. HAIC [19] predicts object motion from proprioception and reference motions to push and pull a cart, WholeBodyVLA [27] pushes a 50 kg cart within composite tasks, and other systems address visual goal pushing [28] and compliant transport [29]. However, these policies rarely estimate the cart motion and the hand forces together, although precise pushing depends on both. Payload mass and distribution change between transport tasks, so the mapping from hand forces to cart motion varies and is difficult to capture with a fixed object model. A controller must therefore adjust the magnitude and direction of its hand forces from online feedback.

2.2. Force Estimation and Force-Adaptive Control for Humanoid Robots

Recent humanoid controllers increasingly treat contact force as part of the control objective. FACET [30], SoftMimic [31], and GentleHumanoid [32] obtain compliant responses through a virtual mass-spring-damper system, augmented motion data, and impedance-based motion tracking, respectively. For intense interaction, FALCON [15] trains upper- and lower-body agents with a force curriculum, HAFO [33] couples them through a constrained residual action space, and Thor [34] adds a force-adaptive torso-tilt objective. Much of the sensorless force control originates from other legged platforms, whose estimation principles also apply to humanoids. Portela et al. [17] track commanded end-effector forces from proprioception on a quadruped manipulator, Zhi et al. [18] unify position and force control on quadrupedal and humanoid platforms, Zeng et al. [22] supply a hybrid force estimate as an explicit policy observation on a wheeled-legged robot, and CompliantWBC [35] brings force latent estimation to a heavy humanoid. However, these controllers target a commanded force, a compliant response, or

disturbance rejection, and they generally do not estimate how the manipulated object moves.

2.3. Residual Reinforcement Learning for Humanoid Control

General motion-tracking and locomotion policies provide reusable whole-body skills [11, 13, 14, 36–38], but they lack the precision and object awareness that loco-manipulation requires. Residual learning addresses this gap by refining an existing controller instead of learning the task from scratch [39]. Residual actions compensate for the dynamics mismatch of retargeted references [40] and adapt general trackers to new interfaces [41]. ResMimic [12] refines a general motion-tracking policy with an object-aware residual policy, and HDMI [42] co-tracks robot and object states from human videos through a residual action space. SEEC [43] and SteadyTray [44] instead cancel locomotion-induced disturbances at the end effectors and the tray, respectively, and Xiao et al. [16] add a delta-command policy over separately trained upper- and lower-body policies. However, the object-aware residuals depend on per-task reference motions and often on motion-capture object states, whereas the stabilizing residuals keep the end effectors or payload steady rather than drive an object. Neither suits sustained interaction with a moving cart, whose route has no reference motion and whose velocity is not measured onboard.

3. Problem Formulation

A humanoid robot grasps the cart handle with both hands and drives the unactuated cart through its walking and hand forces, as illustrated in Figure 2. The robot must regulate the planar cart velocity $\boldsymbol{v}_t^{\text{cart}}$ to track a commanded velocity $\boldsymbol{v}_t^{\text{cart,cmd}}$ while maintaining its balance and its grasp throughout the motion.

We split the state of the robot–cart system at control step t into an observable state $\boldsymbol{s}_t^{\text{obs}}$ and an unobservable state $\boldsymbol{s}_t^{\text{unobs}}$, so that the system $\mathcal{M}$ evolves as

$$\boldsymbol{s}_{t+1} = \mathcal{M}(\boldsymbol{s}_t^{\text{obs}}, \boldsymbol{s}_t^{\text{unobs}}, \boldsymbol{a}_t), \tag{1}$$

where $\boldsymbol{s}_t = (\boldsymbol{s}_t^{\text{obs}}, \boldsymbol{s}_t^{\text{unobs}})$ and $\boldsymbol{a}_t$ is the whole-body action. The observable state contains the signals available on the robot, namely proprioception, the relative cart heading θ_t^{cart} from the head camera, and the velocity command $\boldsymbol{v}_t^{\text{cart,cmd}}$. However, the hands carry no force sensors to measure the hand wrenches $\boldsymbol{F}_t^{\text{hand}}$, everyday carts carry no motion sensors to measure the cart velocity $\boldsymbol{v}_t^{\text{cart}}$, and the payload is unknown before the interaction begins, so the robot cannot directly perceive its interaction with the cart. This interaction information is critical because the four swivel casters impose no fixed-wheel constraint, which leaves the lateral and yaw motions of the cart primarily driven by the hand wrenches, while changes in the payload mass and distribution between transport tasks alter the mapping from hand wrenches to cart motion. Precise cart control thus depends on both $\boldsymbol{s}_t^{\text{obs}}$ and $\boldsymbol{s}_t^{\text{unobs}}$, and the latter must be inferred rather than measured.

In the system $\mathcal{M}$, the hand wrenches drive the cart while their reactions disturb the balance of the robot, so locomotion and manipulation are dynamically coupled. Directly training a policy on the system $\mathcal{M}$ is difficult because a single policy must learn

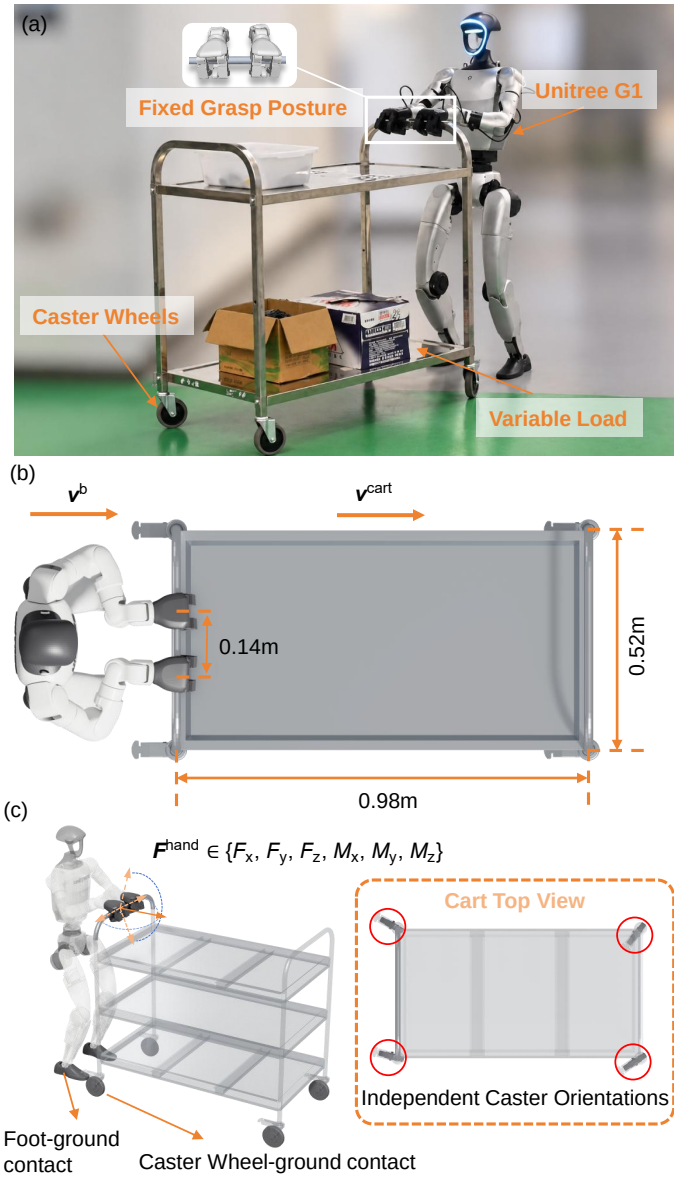


Figure 2: (a) A Unitree G1 grasps the cart handle with a fixed grasp posture, while the cart rolls on caster wheels and carries a variable payload. (b) Top view of the robot–cart system, indicating the bimanual grasp spacing, the cart dimensions, the robot base velocity $\boldsymbol{v}^{\text{b}}$, and the cart velocity $\boldsymbol{v}^{\text{cart}}$. (c) Contact interactions that govern the coupled robot–cart dynamics.

locomotion and manipulation at once, which leads to inefficient exploration [15], and it must do so under contact conditions that it cannot observe. We therefore decompose the system into two subsystems, $\mathcal{M}_1$ and $\mathcal{M}_2$, that are trained separately, while the physical system itself remains coupled. Because the policies act on the observable state, we describe each subsystem by the evolution of this state. From the perspective of locomotion, the cart affects the balance of the robot mainly through the reaction wrenches at the hands, which allows locomotion training to replace the cart with a hand wrench disturbance $\boldsymbol{F}_t^{\text{ext}}$ representing these reactions. This yields the first subsystem, in which the

robot learns to walk stably under the reactions of cart pushing, as

$$\boldsymbol{s}_{t+1}^{\mathrm{obs,r}} = \mathcal{M}_1(\boldsymbol{s}_t^{\mathrm{obs,r}}, \boldsymbol{F}_t^{\mathrm{ext}}, \boldsymbol{a}_t), \tag{2}$$

where $\boldsymbol{s}_t^{\mathrm{obs,r}}$ is the robot part of $\boldsymbol{s}_t^{\mathrm{obs}}$, and $\boldsymbol{a}_t$ comprises the leg action $\boldsymbol{a}_t^{\mathrm{loco}}$ of the locomotion policy π_{base} and bounded upper-body posture offsets. However, because $\mathcal{M}_1$ contains no cart, it can stabilize the walking of the robot but cannot make the cart track the commanded velocity, which calls for a second subsystem with the cart attached. In this subsystem, π_{base} is frozen because it has already learned to keep balance, and the task policy π_{task} only learns to control the cart through residual actions $\boldsymbol{a}_t^{\mathrm{res}}$ that act on top of π_{base}, without relearning walking. Unlike in $\mathcal{M}_1$, the hand wrenches in $\mathcal{M}_2$ are no longer applied disturbances but arise from the contact between the robot and the cart, and they depend on the payload and the cart motion. Because these unobservable states are not available to the policy, $\mathcal{M}_2$ forms a partially observable Markov decision process. Since they shape how proprioception evolves, they can be estimated from the proprioceptive history $\boldsymbol{H}_t$ through an estimator $\mathcal{E}$, as

$$\hat{\boldsymbol{s}}_t^{\mathrm{unobs}} = \mathcal{E}(\boldsymbol{H}_t). \tag{3}$$

With this estimate in place of the unobservable states, the second subsystem as seen by the task policy becomes

$$\boldsymbol{s}_{t+1}^{\mathrm{obs}} = \mathcal{M}_2(\boldsymbol{s}_t^{\mathrm{obs}}, \hat{\boldsymbol{s}}_t^{\mathrm{unobs}}, \boldsymbol{a}_t^{\mathrm{res}}). \tag{4}$$

The task policy is then trained to maximize the expected discounted return

$$J(\pi_{\mathrm{task}}) = \mathbb{E}\left[\sum_{t=0}^{T-1} \gamma^t r_t^{\mathrm{task}}\right], \tag{5}$$

where $\gamma \in (0, 1)$ is the discount factor, T is the episode horizon, and r_t^{task} is the task reward defined in Equation (14).

4. Method

For $\mathcal{M}_1$, in which hand wrench disturbances stand in for the cart, we pre-train a locomotion prior under randomized hand wrenches and bounded upper-body posture offsets in Section 4.1, so that the robot remains balanced under the reactions and postures that arise in cart pushing. For $\mathcal{M}_2$, we realize the estimator $\mathcal{E}$ in Section 4.2 with a force-guided interaction inference module, which infers force and state latents from the history $\boldsymbol{H}_t$ to represent the unobservable states. Using these latents, the task policy in Section 4.3 adds base-velocity command adjustments and arm, waist, and leg joint residuals to the frozen prior, coordinating the whole body to track the commanded cart velocity.

4.1. Locomotion Prior for Interactive Cart Pushing

The humanoid controls the cart's motion through coordinated walking and force application at the hands. During this process, the robot adjusts its arm and waist configuration to regulate hand-contact wrenches while experiencing reaction forces and moments from the cart. As formulated in Section 3, $\mathcal{M}_1$ requires the robot to maintain balance and track base-velocity commands under these combined effects. We therefore pre-train the locomotion prior under simultaneous upper-body posture variations and hand loading to provide support for subsequent cart manipulation (Figure 3(a)).

During pre-training, the robot tracks base-velocity commands without the cart attached. Three-axis external forces and moments are continuously applied at each palm, implementing the hand-wrench disturbances that replace the cart's reactions in the locomotion subsystem. Meanwhile, the arm and waist joint targets vary around the nominal pushing posture within bounds matched to the upper-body residual range in the second stage. These training conditions expose the locomotion prior to both the reaction loads and the configuration changes associated with cart manipulation, allowing locomotion support to be learned before introducing cart-motion control.

The locomotion prior is trained to track base-velocity commands while maintaining balance under the posture variations and hand loading described above. Its reward combines four groups of weighted terms as

$$r_t^{\mathrm{loco}} = r_t^{\mathrm{track}} + r_t^{\mathrm{balance}} + r_t^{\mathrm{gait}} - c_t^{\mathrm{reg}}. \tag{6}$$

The principal velocity-tracking term, which rewards accurate forward and lateral base velocities and yaw rate, is

$$r_t^{\mathrm{track}} = \sum_{i\in\{x,y,\omega\}} w_i \exp\left[-\frac{([\boldsymbol{v}_t^{\mathrm{b}}]_i - [\boldsymbol{v}_t^{\mathrm{cmd}}]_i)^2}{\sigma_i}\right], \tag{7}$$

where $\boldsymbol{v}_t^{\mathrm{b}} = [v_{x,t}^{\mathrm{b}}, v_{y,t}^{\mathrm{b}}, \omega_{z,t}^{\mathrm{b}}]^\top$ contains the measured base-motion components, $\boldsymbol{v}_t^{\mathrm{cmd}}$ denotes their commands, and $w_i > 0$ and $\sigma_i > 0$ specify the tracking weight and error width, respectively.

The balance terms regulate base height and penalize excessive tilt and undesired base motion. The gait terms regulate foot contact and swing behavior, including clearance and slip. The nonnegative regularization cost penalizes abrupt action changes, excessive joint effort, and joint-limit violations. Together, these terms encourage the robot to maintain commanded motion under loading while regulating its supporting posture, foot behavior, and actuation. Episodes terminate upon excessive body tilt or designated non-foot contacts.

After pre-training, the parameters of the locomotion prior are frozen. In $\mathcal{M}_2$, the task policy uses this prior to coordinate base-velocity commands and bounded joint residuals, learning cart-motion regulation on top of the acquired locomotion support.

4.2. Latent Interaction Dynamics Inference under Partial Observability

The humanoid transmits forces to the cart through its hand–handle contacts, driving the cart's translation and rotation. Together with the cart's inertia and wheel–ground interactions, these contact forces determine its acceleration and thereby influence the evolution of its velocity. Meanwhile, hand loading is reflected in the robot's joint torques and joint motion, while the history of the relative cart heading provides partial information about the motion response. These onboard signals contain information linking contact loading to the resulting motion, providing a basis for inferring interaction states without external force sensors.

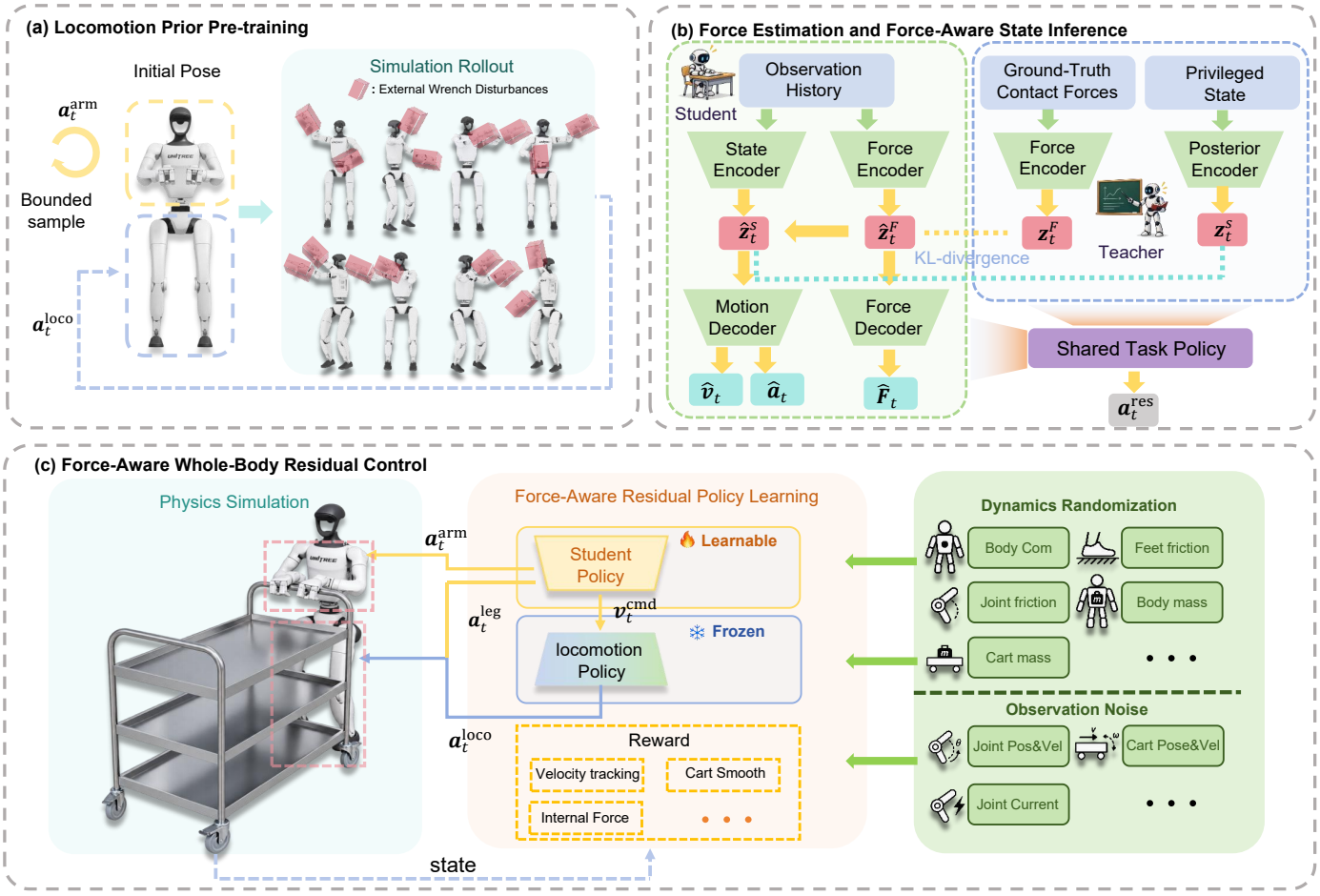


Figure 3: (a) The locomotion prior is pre-trained to track base-velocity commands under random palm wrenches and bounded arm and waist posture variations. (b) The teacher encodes privileged contact forces and states into force and state latents, whereas the student infers both latents from onboard observation histories and reconstructs the contact forces and the velocities and accelerations of the robot and the cart. (c) Using the inferred latents, the task policy outputs base-velocity command adjustments for the frozen prior together with arm, waist, and leg joint residuals, and it is trained under dynamics randomization and observation noise with rewards for cart-velocity tracking, cart-motion smoothness, and internal-force suppression.

Building on this relationship, we construct a teacher–student inference architecture that uses contact-force information for both whole-body coordination and motion-state estimation (Figure 3(b)). During training, the teacher encodes the unobservable states into a force latent and a state latent from privileged simulation information, and the task policy learns to act on these latents. This privileged information is unavailable at deployment, so the student predicts both latents from the observation history $\boldsymbol{H}_t$, thereby recovering the unobservable states from signals available on the robot. The student thus realizes the estimator $\mathcal{E}$ of Equation (3), and the predicted latents form the estimate $\hat{\boldsymbol{s}}_t^{\text{unobs}} = (\hat{\boldsymbol{z}}_t^F, \hat{\boldsymbol{z}}_t^s)$. The student first infers a force latent from observation histories containing joint motion, joint torques, and the relative cart heading. This latent is fed directly to the task policy to provide contact-loading information; it is also fused with the observation history to estimate the motion states of the robot and the cart and produce a state latent for the task policy.

During training, the teacher constructs separate force and state latents from privileged simulation information. The force encoder maps the reference contact forces $\boldsymbol{F}_t^\star$ to a force latent $\boldsymbol{z}_t^F$, while the state encoder maps true velocities and other privileged information $\boldsymbol{e}_t$ to a state latent $\boldsymbol{z}_t^s$. Explicit contact-force channels are excluded from $\boldsymbol{e}_t$. Both latents enter the task policy and provide supervision targets for the student, as

$$\boldsymbol{z}_t^F = E_F^T(\boldsymbol{F}_t^\star), \qquad \boldsymbol{z}_t^s = E_s^T(\boldsymbol{e}_t). \tag{8}$$

To recover these representations from onboard signals, the student first processes the observation history $\boldsymbol{H}_t$ through a causal temporal encoder, producing a sequence of force latents $\hat{\boldsymbol{Z}}_t^F$. The current force latent $\hat{\boldsymbol{z}}_t^F$ is aligned with the teacher's $\boldsymbol{z}_t^F$ and decoded through a shared decoder D_F to reconstruct the reference contact forces. These two objectives train the student to recover the teacher's force representation while retaining contact-loading information. The resulting force latent is supplied directly to the task policy and also used in subsequent motion-state inference.

To relate contact loading to the motion response, the student fuses the force-latent sequence with the corresponding observation frames. A temporal convolutional branch processes the

full 32-frame history, while a second branch processes the most recent six frames to retain recent observation details. The fused features $\boldsymbol{h}_t$ produce the state latent and auxiliary velocity and acceleration estimates, as

$$\hat{\boldsymbol{Z}}_t^F = E_F^S(\boldsymbol{H}_t), \quad \hat{\boldsymbol{F}}_t = D_F(\hat{\boldsymbol{z}}_t^F), \quad \boldsymbol{h}_t = G(\boldsymbol{H}_t, \hat{\boldsymbol{Z}}_t^F), \tag{9}$$

$$(\hat{\boldsymbol{z}}_t^s, \hat{\boldsymbol{v}}_t, \hat{\boldsymbol{a}}_t) = D_s(\boldsymbol{h}_t), \tag{10}$$

where D_s is the state decoder, whose output is partitioned into the estimated state latent $\hat{\boldsymbol{z}}_t^s$, the velocity estimate $\hat{\boldsymbol{v}}_t$, which stacks the estimated planar linear velocity and yaw rate of the cart and the planar base linear velocity of the robot, and the acceleration estimate $\hat{\boldsymbol{a}}_t$, which contains the corresponding accelerations.

The state latent $\hat{\boldsymbol{z}}_t^s$ is aligned with the teacher's $\boldsymbol{z}_t^s$, allowing it to replace the privileged representation used by the task policy during deployment. In parallel, the velocity and acceleration estimates are supervised by simulation reference values, requiring the shared features to explain explicit motion states. Latent alignment therefore trains the student to recover the policy's interaction representation, while motion supervision constrains the information retained in the shared features. Because these features depend on the force-latent sequence, motion-estimation errors also propagate through the force encoder. The force representation is thus trained both to reconstruct contact loading and to support motion-state inference. The task policy receives the force and state latents, while the velocity and acceleration estimates serve as auxiliary supervised outputs.

The student is optimized with the joint objective

$$\mathcal{L}_{\text{student}} = \mathcal{L}_{\text{align}} + \lambda_F \mathcal{L}_F + \lambda_v \mathcal{L}_v + \lambda_a \mathcal{L}_a, \tag{11}$$

$$\mathcal{L}_{\text{align}} = \left\| \hat{\boldsymbol{z}}_t^F - \text{sg}(\boldsymbol{z}_t^F) \right\|_2 + \left\| \hat{\boldsymbol{z}}_t^s - \text{sg}(\boldsymbol{z}_t^s) \right\|_2 . \tag{12}$$

Here, sg denotes the stop-gradient operator, keeping the teacher targets fixed during each student update. The force, velocity, and acceleration losses are mean-squared errors between normalized estimates and their simulation references, and $\lambda_F, \lambda_v, \lambda_a$ are positive weights. During teacher optimization, force reconstruction and variance regularization constrain the force latent, while a regularization term draws the state latent toward the student estimate to encourage recoverability from onboard observations. Training periodically uses student-estimated latents to collect rollouts, exposing the inference network to states encountered under its own estimation errors.

During training, the direct upper-body joint-torque channels in the task-policy observations are progressively suppressed, while these measurements remain available to the student inference network. Contact-force information therefore reaches the task policy through the force latent rather than through the raw torques, which makes the policy explicitly force-aware. At deployment, the privileged encoders are removed, and the shared task policy uses recent observations together with the estimated force and state latents for whole-body residual coordination.

4.3. Force-Aware Whole-Body Residual Coordination

The task policy uses recent onboard observations and the force and state latents inferred in Section 4.2 to coordinate the humanoid's motion for cart-velocity tracking (Figure 3(c)). The locomotion prior provides the underlying walking response, with its parameters frozen during task training. The task policy learns to adjust this response and the upper-body configuration according to the inferred interaction, so that locomotion and manipulation jointly serve the commanded cart motion. Whereas π_{base} tracks robot base-velocity commands in $\mathcal{M}_1$, π_{task} is optimized in $\mathcal{M}_2$ for the cart's forward and lateral velocities and yaw rate.

The task policy maps the recent observations and the inferred latents to the residual action

$$\boldsymbol{a}_t^{\text{res}} = \pi_{\text{task}}(\boldsymbol{O}_t, \hat{\boldsymbol{z}}_t^F, \hat{\boldsymbol{z}}_t^s), \quad \boldsymbol{a}_t^{\text{res}} = [\boldsymbol{a}_t^{\text{base}}, \boldsymbol{a}_t^{\text{arm}}, \boldsymbol{a}_t^{\text{leg}}]. \tag{13}$$

The policy produces 30 coordinated action components, namely three base-velocity command adjustments, 15 arm and waist joint residuals, and 12 leg joint residuals. The base-command adjustments modify the motion requested from the locomotion prior, allowing the robot's walking response to be adapted to the desired cart motion. The arm and waist residuals adjust the nominal pushing posture and thereby regulate hand-contact forces through joint control while maintaining the grasp. The leg residuals modify the prior's leg targets to accommodate task-specific support adjustments under contact loading. All three outputs are generated jointly, allowing base motion, force application, and leg support to be coordinated toward the same cart-motion objective.

The task reward encourages accurate and smooth cart motion while maintaining robot balance and hand contact, and it explicitly penalizes the internal load between the two hands. Its weighted terms are grouped as

$$r_t^{\text{task}} = r_t^{\text{cart}} - c_t^{\text{smooth}} - w_{\text{int}} J_t^{\text{int}} - c_t^{\text{grasp}} + r_t^{\text{support}} - c_t^{\text{reg}}. \tag{14}$$

The tracking reward encourages the cart's forward and lateral velocities and yaw rate to follow their references, and the smoothness costs suppress cart-velocity fluctuations and abrupt changes in contact forces. Because both hands hold the same handle, only part of the bimanual hand wrench moves the cart. We separate this driving part through $\boldsymbol{d}_t^{\text{dr}} = \boldsymbol{U}\boldsymbol{F}_t^{\text{hand}}$, where the constant matrix $\boldsymbol{U} \in \mathbb{R}^{4\times 12}$ maps the bimanual hand wrench to the push force of each hand, the sum of the lateral forces, and the sum of the yaw moments. The remaining internal wrench produces no cart motion and only consumes joint torque margin,

$$\boldsymbol{F}_t^{\text{int}} = \boldsymbol{F}_t^{\text{hand}} - \boldsymbol{\Sigma}\boldsymbol{U}^\top (\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^\top)^{-1} \boldsymbol{U}\boldsymbol{F}_t^{\text{hand}}, \tag{15}$$

and the internal-load cost penalizes it in proportion to the load capacity of each grasp direction,

$$J_t^{\text{int}} = \left\| \boldsymbol{\Sigma}^{-1/2} \boldsymbol{F}_t^{\text{int}} \right\|_2^2, \tag{16}$$

where $\boldsymbol{\Sigma}$ is a diagonal matrix of squared grasp capacities. The hand wrench is low-pass filtered before this decomposition, and the vertical load needed for frictional grip is not counted as internal. The grasp cost penalizes grasp slip, abrupt force changes, and wrist saturation, while the support and regularization terms promote balanced locomotion and smooth, moderate control actions.

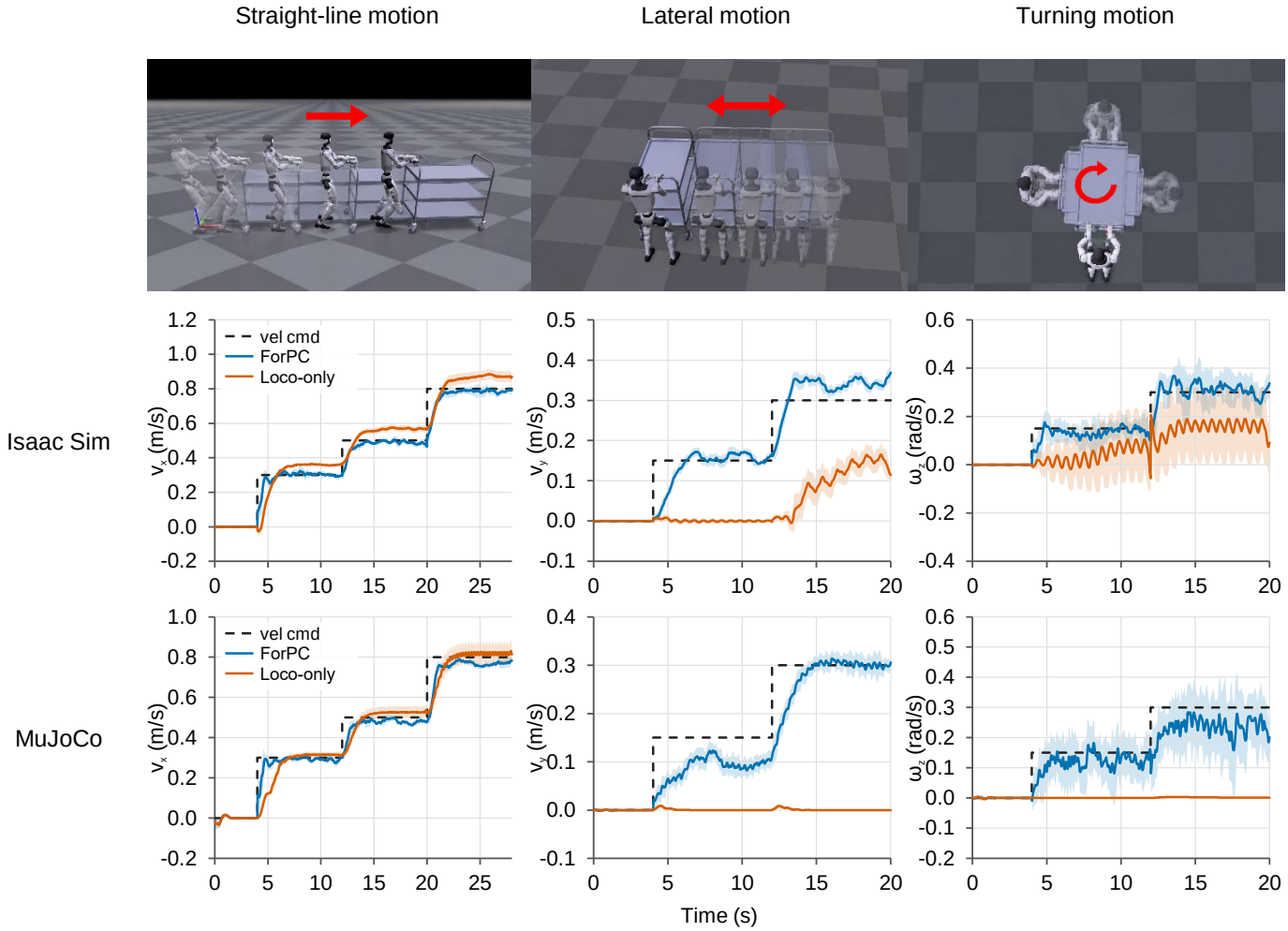


Figure 4: Cart-velocity tracking by ForPC and Loco-only in Isaac Sim and MuJoCo. Loco-only uses the Stage I policy without task-level corrections. Both policies are trained only in Isaac Sim, so the MuJoCo results serve as a sim-to-sim test; shaded bands denote standard deviations.

5. Experiments

We evaluate the cart-pushing performance of ForPC in simulation and on a physical Unitree G1 from three aspects. First, we assess whether ForPC tracks cart-velocity commands accurately with low velocity fluctuations in forward, lateral, and turning motions and maintains this accuracy over complete routes under different payloads (Section 5.2). Second, we analyze whether the inferred latents retain contact-loading and cart-motion information and whether force awareness directs the bimanual forces toward driving the cart rather than toward mutual opposition between the hands (Section 5.3). Third, we test whether the simulation-trained policy transfers to the physical robot and accomplishes real-world cart transport without real-world training or fine-tuning (Section 5.4).

5.1. Experimental Setup

The robot is a Unitree G1 humanoid with 12 leg joints, one waist yaw joint, and 14 arm joints, and the cart is a four-caster platform trolley with an empty mass of 12 kg. The robot carries three-fingered Dex3-1 hands held at a fixed grasp configuration, the waist roll and pitch joints are held at zero, and a RealSense camera in the head detects an AprilTag fixed to the cart to measure the relative cart heading. The payload is the mass that is added to the cart. The policies are trained in Isaac Sim with the Isaac Lab framework [45] and are evaluated with fixed weights in Isaac Sim and in MuJoCo [46]. The MuJoCo evaluation is therefore a sim-to-sim test in which the contact model, the solver, and the actuator integration differ from those of training. Both the policy and the simulators run at 50 Hz control rate, and the cart velocities are sampled at 50 Hz.

We compare ForPC with three configurations to evaluate the roles of task-level coordination, locomotion-prior training, and force awareness. The first, locomotion-only, uses the Stage I policy alone, taking the commanded cart velocity as its base-velocity command and maintaining the nominal bimanual contact posture without task-level corrections. The second retains force-aware whole-body residual coordination but uses a locomotion prior trained without hand-wrench disturbances or upper-body posture randomization. The third uses the same prior-training conditions as the second but additionally removes explicit force awareness from the residual coordination framework.

Table 1: Cart-velocity tracking RMSE and RMSF in Isaac Sim and MuJoCo at two payloads. Units are m/s for forward and lateral motion and rad/s for turning; ↓ indicates that lower is better, and bold marks the best value per simulator and payload.

Simulator	Payload (kg)	Method	Forward		Lateral		Turning	
			RMSE ↓	RMSF ↓	RMSE ↓	RMSF ↓	RMSE ↓	RMSF ↓
Isaac Sim	15	Loco-only	0.1768	0.1487	0.1564	0.1387	0.2989	0.2611
		Standard prior w Force-aware	0.0816	0.0612	0.0908	0.0613	0.1067	**0.1103**
		Standard prior w/o Force-aware	0.0984	0.0758	0.1421	0.0612	0.1311	0.1221
		ForPC	**0.0801**	**0.0590**	**0.0892**	**0.0591**	**0.1010**	0.1203
	25	Loco-only	0.2003	0.1449	0.1612	0.1320	0.2958	0.2453
		Standard prior w Force-aware	0.0856	0.0546	0.1064	0.0523	0.1048	0.1031
		Standard prior w/o Force-aware	0.1026	0.0680	0.1548	0.0516	0.1305	0.1061
		ForPC	**0.0851**	**0.0501**	**0.1022**	**0.0515**	**0.0941**	**0.0959**
MuJoCo	15	Loco-only	0.1393	0.1417	0.1494	0.1281	0.2830	0.2162
		Standard prior w Force-aware	0.0736	0.0606	0.0799	0.0691	0.1149	**0.1059**
		Standard prior w/o Force-aware	0.0882	0.0683	0.1159	0.0573	0.1286	0.1240
		ForPC	**0.0707**	**0.0573**	**0.0739**	**0.0551**	**0.1087**	0.1109
	25	Loco-only	0.1578	0.1383	0.1541	0.1206	0.2805	0.2046
		Standard prior w Force-aware	0.0773	0.0539	0.0964	0.0526	0.1164	**0.0958**
		Standard prior w/o Force-aware	0.0928	0.0632	0.1246	0.0596	0.1365	0.1074
		ForPC	**0.0731**	**0.0498**	**0.0834**	**0.0502**	**0.1032**	0.0986

Table 2: Route-tracking results of four configurations at two payloads under matched speed limits. ↓ and ↑ indicate that lower and higher are better, respectively; bold marks the best value per payload.

Payload (kg)	Method	RMSE ↓		Completion time (s) ↓	Success rate (%) ↑
		Cross-track (m)	Heading (deg)		
15	Loco-only	0.2153	23.83	62.00	89.1
	Standard prior w Force-aware	0.0658	5.70	42.49	93.8
	Standard prior w/o Force-aware	0.1494	14.22	45.81	90.6
	ForPC	**0.0648**	**5.63**	**41.49**	**100**
25	Loco-only	0.2652	25.70	71.07	76.6
	Standard prior w Force-aware	0.0735	6.36	43.47	95.3
	Standard prior w/o Force-aware	0.1641	16.55	47.98	89.1
	ForPC	**0.0685**	**5.98**	**42.45**	**100**

Full ForPC combines the locomotion prior trained under hand loading and upper-body posture variations with force-aware coordination of base-velocity command adjustments and arm, waist, and leg joint residuals. All configurations are evaluated using the same commands, cart, and environmental conditions.

5.2. Humanoid Cart-Pushing Control

We evaluate humanoid cart-pushing control through velocity tracking and complete-route tracking.

5.2.1. Cart-Velocity Tracking

To evaluate tracking accuracy and velocity fluctuations during humanoid cart pushing, we compare full ForPC with the three configurations introduced in Section 5.1 through forward, lateral, and turning velocity-tracking tests at payloads of 15 kg and 25 kg. At each payload, all configurations use identical velocity commands and resistance settings. We apply sequences of constant commands separately for forward translation (0.3, 0.5, and 0.8 m/s), lateral translation (0.15 and 0.3 m/s), and rotation about the vertical axis (0.15 and 0.3 rad/s), holding each command for 8 s. Tracking errors and fluctuations in the cart's forward velocity, lateral velocity, and yaw rate are compared to assess the control performance of each configuration and its variation with payload.

Evaluation metrics. We use the root-mean-square error (RMSE) to evaluate velocity-tracking accuracy and RMS velocity fluctuation (RMSF) to quantify velocity variations within constant-command plateaus. Let T denote the number of samples used to compute the metrics. For each evaluated component, let v_t and v_t^{cmd} denote the actual and commanded velocities in

the same coordinate frame, and let $e_t = v_t - v_t^{\mathrm{cmd}}$ denote their difference; actual velocities are unfiltered, and uncommanded components have a zero reference. The following scalar formulas are applied separately to forward velocity, lateral velocity, and yaw rate.

(1) RMSE. For a scalar error e_t over T samples, the RMSE is

$$\mathrm{RMSE} = \sqrt{\frac{1}{T}\sum_{t=1}^{T} e_t^2}, \tag{17}$$

which we use for all tracking and estimation errors in this section; for velocity tracking, e_t is the velocity error defined above.

(2) RMSF. RMSF measures the root-mean-square deviation of velocity from its own command-plateau mean, excluding the plateau mean tracking error and changes between plateau means. With the plateau $k(t)$ that contains sample t and the mean velocity μ_k over the evaluated samples of plateau k, it is

$$\mathrm{RMSF} = \sqrt{\frac{1}{T}\sum_{t=1}^{T} (v_t - \mu_{k(t)})^2}. \tag{18}$$

For every plateau, the squared RMSE equals the squared RMSF plus the squared mean error, and this metric includes the slow drift within a plateau because no frequency separation is applied. All configurations use identical plateaus, and lower values indicate smaller tracking errors or weaker fluctuations.

Results.

Figure 4 shows that ForPC responds to forward, lateral, and yaw commands in both simulators. Once the forward motion is established, Loco-only follows the straight-line speed changes, but its lateral and turning responses remain weak in Isaac Sim and are nearly absent in MuJoCo. These responses show that locomotion capability alone is insufficient for effective lateral and turning control of the dynamically coupled cart. These maneuvers require precise regulation of the hand-contact forces and moments to drive the cart against the resistance while the robot maintains its balance under the resulting reactions. By coordinating this interaction with whole-body support, ForPC follows the lateral and yaw commands more closely and extends the effective cart control beyond forward pushing.

Table 1 shows that force-aware whole-body coordination enables precise and smooth cart control that locomotion alone cannot provide. ForPC attains the lowest RMSE in all twelve combinations of simulator, payload, and motion, reducing RMSE by up to 68.2% and RMSF by up to 65.4% relative to Loco-only. The gap is largest in turning, where the cart must be rotated by a bimanual yaw moment that walking alone cannot generate.

Force awareness allows the policy to apply the hand forces that the cart actually requires. With the same standard prior, adding force awareness lowers RMSE by up to 36.1%, most strongly in lateral motion, which is driven mainly by lateral hand forces. Perceiving the contact loading lets the policy correct the steady tracking offset caused by unmeasured resistance.

Training the locomotion prior under hand loading and posture variations further lowers RMSE by up to 13.5%. Because this prior already keeps the robot balanced under hand reactions, the residual policy can concentrate on cart control, and the benefit grows at 25 kg and in MuJoCo, where the reactions are larger and the dynamics differ from training. Overall, these results demonstrate that ForPC achieves accurate cart-velocity tracking under different payloads.

5.2.2. *Route Tracking*

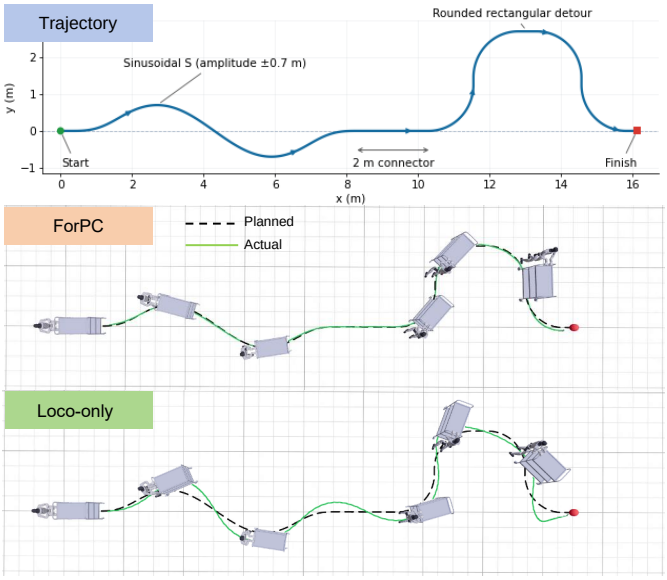


Figure 5: Route tracking by ForPC and Loco-only in simulation. Cart trajectories are overlaid on robot–cart snapshots.

To evaluate how velocity control supports continuous cart transport, we test route tracking through changes in curvature and a final stop. The route comprises an S-bend, a straight connector, and a rounded rectangular detour, ending at a target stopping pose (Figure 5). We compare full ForPC with the three configurations in Section 5.1 at payloads of 15 kg and 25 kg, using cross-track RMSE, heading RMSE, completion time, and success rate. Completion requires reaching the target within the time limit and the prescribed pose tolerances while the robot maintains its balance and the hand contact. All configurations use the same reference route, route tracker, and speed limits.

Evaluation metrics. We evaluate route tracking using cross-track RMSE, heading RMSE, completion time, and success rate, with T denoting the number of samples in the route-tracking interval.

(1) Cross-track RMSE. The cross-track RMSE, reported in metres, applies Equation (17) to the signed lateral distance ε_t from the cart to the reference route at sample t.

(2) Heading RMSE. The heading RMSE applies Equation (17) to the heading error $\mathrm{wrap}(\psi_t^{\mathrm{cart}} - \psi_t^{\mathrm{ref}})$, where ψ_t^{cart} is the cart heading in the world frame, ψ_t^{ref} is the reference route heading at the nearest route point in the same frame, and $\mathrm{wrap}(\cdot)$ maps each angular difference to $(-\pi, \pi]$. The error is computed in radians, and the resulting RMSE is converted to degrees for reporting.

(3) Completion time. Completion time is the elapsed time, in seconds, from the start of route tracking until the cart first satisfies the endpoint completion criterion defined above. Lower

cross-track and heading RMSE indicate more accurate position and orientation tracking, respectively, whereas a shorter completion time indicates faster route completion under the matched speed limits.

(4) Success rate. The success rate is the percentage of trials that satisfy the completion criterion defined above, computed over 640 trials for each configuration and payload, so a trial fails if the robot loses its balance or the hand contact, or does not reach the target pose within the time limit.

Results. Figure 5 shows that ForPC follows the planned bends more closely, whereas Loco-only deviates from the route, particularly around the rounded rectangular detour. This comparison illustrates how the ability to control lateral and turning motion supports tracking through changes in path curvature.

Table 2 shows that the velocity-tracking advantage of ForPC carries over to complete routes. ForPC achieves the best value in every metric at both payloads; relative to Loco-only, it reduces the cross-track RMSE by up to 74.2% and the heading RMSE by up to 76.7%, shortens the completion time by up to 40.3%, and succeeds in every trial. Accurate lateral and yaw control keeps the cart close to the route through successive bends, so the robot spends less time correcting deviations and completes the route sooner.

Force awareness remains the main contributor. With the same standard prior, it reduces the cross-track RMSE by up to 56.0% and the heading RMSE by up to 61.6%, showing that the precise cart control enabled by force awareness allows ForPC to handle the lateral and heading errors that arise along curved routes.

Training the locomotion prior under hand loading further lowers the cross-track RMSE by up to 6.8% and raises the success rate from 93.8% and 95.3% to 100%, indicating that its balance support keeps the robot from failing over long transports.

As the payload increases from 15 kg to 25 kg, the cross-track RMSE of Loco-only rises by 23.2% and its success rate drops from 89.1% to 76.6%, whereas the cross-track RMSE of ForPC changes only slightly and its success rate remains 100%. These results demonstrate that ForPC achieves accurate and reliable route transport under different payloads.

5.3. Force-Latent Analysis and Bimanual Coordination

Section 5.2 shows that ForPC tracks cart velocities and routes more precisely than the compared configurations. This subsection examines how fusing the force latent contributes to this precision. We first decode the inferred latents into hand forces and cart velocities to test whether they retain contact-loading and cart-motion information, and then examine whether force awareness directs the bimanual forces toward driving the cart rather than toward mutual opposition between the hands.

5.3.1. Information Encoded in the Inferred Latents

We test whether the student's latents retain contact-loading and cart-motion information during cart pushing in Isaac Sim, evaluating the force estimates over cart-pushing runs under various operating conditions and the cart-velocity estimates at a payload of 25 kg. The evaluated outputs are the three contact-force components of each hand and the cart's forward velocity, lateral velocity, and yaw rate. Predictions and simulation references use matching coordinate frames, signs, and timestamps. Hand forces are decoded from the force latent through D_F in Equation (9), and cart velocities are taken from the state decoder D_s in Equation (10), whose input features depend on the force-latent sequence.

Table 3: ForPC estimation accuracy in Isaac Sim, with force components expressed in the cart-heading frame.

Quantity	RMSE (N) ↓	R^2 ↑
Left-hand forward force	7.54	0.97
Left-hand lateral force	4.32	0.78
Left-hand vertical force	4.32	0.95
Right-hand forward force	7.36	0.97
Right-hand lateral force	4.28	0.78
Right-hand vertical force	4.28	0.95
Quantity	**RMSE (m/s, rad/s) ↓**	
Cart forward velocity	0.035	
Cart lateral velocity	0.035	
Cart yaw rate	0.050	

Evaluation metrics. For each force or velocity component, the estimation RMSE applies Equation (17) to the error $\hat{y}_t - y_t$ between the predicted value $\hat{y}_t$ and its simulation reference y_t at sample t. This metric compares an estimate with its actual reference, whereas the velocity-tracking RMSE in Section 5.2.1 compares actual motion with the commanded motion. For the force components, we also report the coefficient of determination

$$R^2 = 1 - \frac{\sum_{t=1}^{T}(y_t - \hat{y}_t)^2}{\sum_{t=1}^{T}(y_t - \bar{y})^2}, \tag{19}$$

where $\bar{y}$ is the mean of the reference over the T samples; R^2 measures the fraction of the reference variation explained by the estimate, and values closer to one indicate closer agreement.

Results. Table 3 reports the decoding accuracy. The component-wise force RMSE ranges from 4.32 N to 7.54 N for the left hand and from 4.28 N to 7.36 N for the right hand, with R^2 values of 0.97 for the forward forces, 0.95 for the vertical forces, and 0.78 for the lateral forces, indicating that the force latent encodes the contact loading of each hand. At a payload of 25 kg, the cart-velocity RMSE is 0.035 m/s for both linear components and 0.050 rad/s for the yaw rate, indicating that the fused features also capture the cart's motion response.

5.3.2. Bimanual Force Utilization

We compare the four configurations in Section 5.1 in matched Isaac Sim pushing trials at payloads of 15 kg and 25 kg. All configurations use the same cart-velocity command sequences, initial conditions, and evaluation duration for each motion type in Section 5.2.1. For every configuration, including those without a force estimator, the hand loads are taken from the simulation-reference hand wrenches.

Evaluation metrics. Only the forward force, lateral force, and yaw moment of each hand act in the plane of cart motion. The vertical forces and the remaining moments mainly sustain the

Table 4: Internal-force ratio ρ^{int} in Isaac Sim; lower is better.

Payload (kg)	Method	Forward (%)	Lateral (%)	Turning (%)
15	Loco-only	83.6	83.7	78.2
	Standard prior w Force-aware	84.7	61.5	57.3
	Standard prior w/o Force-aware	96.9	85.3	89.2
	ForPC	**81.8**	**61.3**	**55.6**
25	Loco-only	81.7	81.5	75.2
	Standard prior w Force-aware	78.7	57.0	53.3
	Standard prior w/o Force-aware	96.1	82.9	87.9
	ForPC	**78.4**	**56.9**	**51.9**

grasp and are transmitted to the ground through the cart wheels; they are therefore excluded from this evaluation. Let $\boldsymbol{P} \in \mathbb{R}^{6\times 12}$ select these six components from $\boldsymbol{F}_t^{\text{hand}}$. Because $\boldsymbol{\Sigma}$ is diagonal and $\boldsymbol{U}$ acts only on these components, Equation (15) splits the capacity-normalized horizontal load into two orthogonal parts: a driving part that moves the cart and an internal part that cancels between the hands. The internal-force ratio over T samples is

$$\rho^{\text{int}} = \frac{\sum_{t=1}^{T} \left\| \boldsymbol{P}\boldsymbol{\Sigma}^{-1/2}\boldsymbol{F}_t^{\text{int}} \right\|_2^2}{\sum_{t=1}^{T} \left\| \boldsymbol{P}\boldsymbol{\Sigma}^{-1/2}\boldsymbol{F}_t^{\text{hand}} \right\|_2^2}, \tag{20}$$

where identical filtering and grasp capacities are applied to all configurations. This ratio lies in $[0, 1]$, and $1 - \rho^{\text{int}}$ is the fraction of the applied horizontal hand load that drives the cart. A lower value indicates that less bimanual effort is lost to mutual opposition between the hands. The ratio is computed separately for forward, lateral, and turning trials and is interpreted together with the tracking errors in Table 1 to verify that the cart remains accurately controlled.

Results. Table 4 shows that force awareness determines whether the bimanual effort drives the cart or is lost to mutual opposition between the hands. Without force awareness, the standard prior yields the highest internal-force ratio in every case, up to 14.4 percentage points above Loco-only, because its upper-body residuals act without perceiving the opposition they create. Adding force awareness lowers the ratio by up to 34.6 percentage points and, in lateral and turning motion, increases the fraction of the horizontal hand load that drives the cart by up to 4.0 times. The effort lost to opposition without force awareness is consistent with the larger lateral RMSE of this configuration in Table 1.

ForPC remains within 3.0 percentage points of the standard prior with force awareness, indicating that the force distribution is set mainly by force awareness, whereas the trained prior contributes through the tracking accuracy and success rate reported in Section 5.2.

In forward motion, all configurations show high ratios with small differences. A plausible explanation is that steady forward pushing requires only a small driving force against rolling resistance, so the inward grip forces that cancel between the hands dominate the horizontal load. For every configuration, the ratio also decreases from 15 kg to 25 kg, because the heavier cart requires larger driving forces.

These results demonstrate that force awareness enables ForPC to direct its bimanual effort toward driving the cart.

5.4. Real-World Deployment

We deploy ForPC on a physical Unitree G1 to examine whether the simulation-trained controller transfers to hardware. The deployed networks, comprising the normalizers, the student inference network, and the task policy, are taken directly from simulation training without real-world training or fine-tuning. On hardware, the policy relies only on proprioception, joint torques, and the relative cart heading obtained from the AprilTag; neither the hand forces nor the cart velocity is measured, and no payload information is provided.

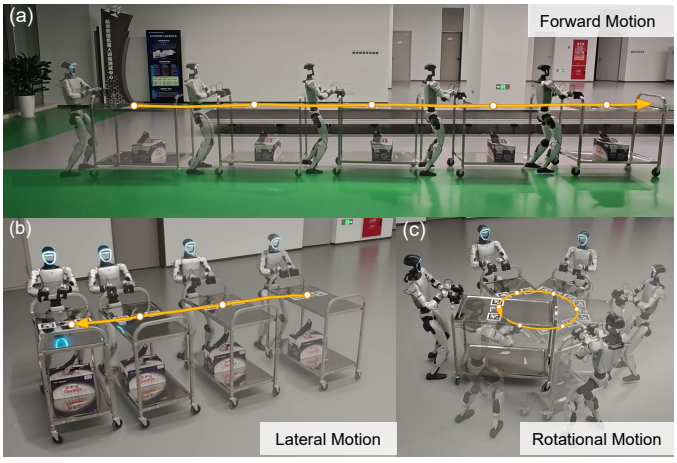


Figure 6: (a) ForPC deployed on a physical Unitree G1 pushes the cart forward along a straight line. (b) The deployed policy moves the cart laterally. (c) The deployed policy rotates the cart.

Figure 6 shows three basic motions on a flat indoor floor. In forward motion, the robot walks behind the cart and pushes it along an approximately straight line through a long corridor. In lateral motion, the robot steps sideways with the cart, which translates across the floor while both hands remain on the handle. In rotational motion, the robot walks around the cart while the cart turns about a nearly fixed point. In lateral and rotational motion, the cart is driven mainly by lateral hand forces or a yaw moment rather than by forward pushing, while the legs reposition the body around the cart. These motions therefore require

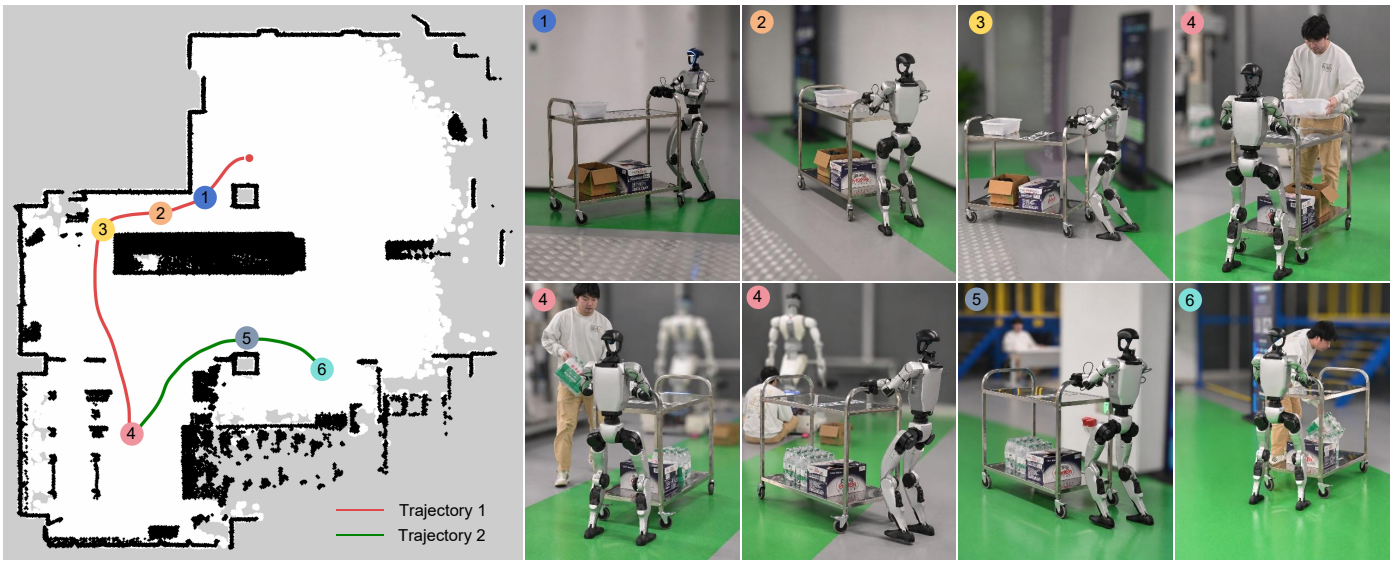


Figure 7: A physical Unitree G1 running ForPC within a navigation system transports materials. The map shows Trajectory 1 (red) and Trajectory 2 (green), and the numbered markers indicate where the corresponding snapshots were taken. (1) The robot turns into a narrow passage. (2) The robot moves straight through the narrow passage. (3) The robot turns onto a straight segment. (4) A worker removes materials from the cart, a worker places materials onto the cart, and the robot turns in place. (5) Under the same policy, the robot pushes the cart with a different payload to complete the material transport. (6) A worker removes the materials from the cart.

the coordinated arm, waist, and leg adjustments described in Section 4.3.

We further integrate ForPC into a navigation system for material transport, a common logistics task, as shown in Figure 7. The navigation system plans the route on a prebuilt map and issues cart-velocity commands, which ForPC executes through whole-body motion. Along Trajectory 1, the robot pushes the cart into a narrow passage, moves straight through it, turns onto a long straight segment, and reaches the transfer station. At this station, a worker unloads the transported materials and places a different payload on the cart, after which the robot turns the cart in place. The robot then pushes the reloaded cart along Trajectory 2 to the destination, where the materials are unloaded. The same policy executes both trajectories consecutively, without resetting the controller or adjusting any parameter after the payload change.

These experiments show that the simulation-trained policy transfers to the physical robot without fine-tuning, sustains coordinated pushing in forward, lateral, and rotational motion, and, in a real transport task with payload changes, pushes a cart carrying up to 30 kg at 0.5 m/s.

6. Conclusion

This paper presented ForPC, a force-aware whole-body control framework for precise humanoid cart pushing. A locomotion prior trained under hand loading and upper-body posture variations provides balance support, a force-guided interaction inference module infers force and state latents from onboard signals without external force sensors, and a residual policy uses these latents to coordinate the arms, waist, and legs. Sim-to-sim experiments show that ForPC reduces cart-velocity RMSE by up to 68.2% and cross-track RMSE by up to 74.2% relative to Loco-only, succeeds in every route trial, and directs more of the bimanual effort toward driving the cart. Trained solely in simulation, the policy transfers to a physical Unitree G1 and pushes a cart carrying up to 30 kg at 0.5 m/s. Future work will extend ForPC to other wheeled objects and contact modes and incorporate richer exteroception for transport in cluttered environments.

References

[1] Y. Tong, H. Liu, Z. Zhang, Advancements in humanoid robots: A comprehensive review and future prospects, IEEE/CAA Journal of Automatica Sinica 11 (2) (2024) 301–328. doi:10.1109/jas.2023.124140.

[2] Z. Gu, J. Li, W. Shen, W. Yu, Z. Xie, S. McCrory, X. Cheng, A. Shamsah, R. Griffin, C. K. Liu, A. Kheddar, X. B. Peng, Y. Zhu, G. Shi, Q. Nguyen, G. Cheng, H. Gao, Y. Zhao, Humanoid locomotion and manipulation: Current progress and challenges in control, planning, and learning, IEEE/ASME Transactions on Mechatronics 31 (2) (2026) 2300–2330. doi:10.1109/tmech.2025.3579247.

[3] A. Kheddar, M. A. Roa, P.-B. Wieber, F. Chaumette, F. Spindler, G. Oriolo, L. Lanari, A. Escande, K. Chappellet, F. Kanehiro, P. Rabate, S. Caron, P. Gergondet, A. Comport, A. Tanguy, C. Ott, B. Henze, G. Mesesan, J. Englsberger, Humanoid robots in aircraft manufacturing: The Airbus use cases, IEEE Robotics & Automation Magazine 26 (4) (2019) 30–45. doi:10.1109/mra.2019.2943395.

[4] L. Hawley, W. Suleiman, Control strategy and implementation for a humanoid robot pushing a heavy load on a rolling cart, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017, pp. 4997–5002. doi:10.1109/iros.2017.8206382.

[5] A. Rioux, W. Suleiman, Autonomous SLAM based humanoid navigation in a cluttered environment while transporting a heavy load, Robotics and Autonomous Systems 99 (2018) 50–62. doi:10.1016/j.robot.2017.10.001.

[6] T. Yoshikawa, K. Nagai, Manipulating and grasping forces in manipulation by multifingered robot hands, IEEE Transactions on Robotics and Automation 7 (1) (1991) 67–77. doi:10.1109/70.68071.

[7] L. Sentis, O. Khatib, Synthesis of whole-body behaviors through hierarchical control of behavioral primitives, International Journal of Humanoid Robotics 2 (4) (2005) 505–518. doi:10.1142/s0219843605000594.

[8] P. M. Wensing, M. Posa, Y. Hu, A. Escande, N. Mansard, A. D. Prete, Optimization-based control for dynamic legged robots, IEEE Transactions on Robotics 40 (2024) 43–63. doi:10.1109/tro.2023.3324580.

[9] J.-P. Sleiman, F. Farshidian, M. V. Minniti, M. Hutter, A unified MPC framework for whole-body dynamic locomotion and manipulation, IEEE Robotics and Automation Letters 6 (3) (2021) 4688–4695. doi:10.1109/lra.2021.3068908.

[10] J. Dao, H. Duan, A. Fern, Sim-to-real learning for humanoid box loco-manipulation, in: 2024 IEEE International Conference on Robotics and Automation (ICRA), 2024, pp. 16930–16936. doi:10.1109/icra57147.2024.10610977.

[11] Y. Ze, Z. Chen, J. P. Araújo, Z.-a. Cao, X. B. Peng, J. Wu, C. K. Liu, TWIST: Teleoperated whole-body imitation system, in: Proceedings of The 9th Conference on Robot Learning, Vol. 305 of Proceedings of Machine Learning Research, PMLR, 2025, pp. 2143–2154.
URL https://proceedings.mlr.press/v305/ze25a.html

[12] S. Zhao, Y. Ze, Y. Wang, C. K. Liu, P. Abbeel, G. Shi, R. Duan, ResMimic: From general motion tracking to humanoid whole-body loco-manipulation via residual learning, arXiv preprint arXiv:2510.05070 (2025). doi:10.48550/arXiv.2510.05070.

[13] Q. Liao, T. E. Truong, X. Huang, Y. Gao, G. Tevet, K. Sreenath, C. K. Liu, BeyondMimic: From motion tracking to versatile humanoid control via guided diffusion, Science Robotics 11 (117) (2026). doi:10.1126/scirobotics.adx8924.

[14] Q. Ben, F. Jia, J. Zeng, J. Dong, D. Lin, J. Pang, HOMIE: Humanoid loco-manipulation with isomorphic exoskeleton cockpit, in: Robotics: Science and Systems XXI, 2025. doi:10.15607/rss.2025.xxi.070.

[15] Y. Zhang, Y. Yuan, P. Gurunath, I. Gupta, S. Omidshafiei, A.-a. Aghamohammadi, M. Vazquez-Chanlatte, L. Pedersen, T. He, G. Shi, FALCON: Learning force-adaptive humanoid loco-manipulation, in: Proceedings of The 8th Annual Learning for Dynamics and Control Conference, Vol. 331 of Proceedings of Machine Learning Research, PMLR, 2026, pp. 265–281.
URL https://proceedings.mlr.press/v331/zhang26a.html

[16] K. Xiao, Z. Xu, C. Zhe, C. Liu, Q. Chen, Kinematics-aware multi-policy reinforcement learning for force-capable humanoid loco-manipulation, arXiv preprint arXiv:2511.21169 (2025).
URL https://arxiv.org/abs/2511.21169

[17] T. Portela, G. B. Margolis, Y. Ji, P. Agrawal, Learning force control for legged manipulation, in: 2024 IEEE International Conference on Robotics and Automation (ICRA), 2024, pp. 15366–15372. doi:10.1109/icra57147.2024.10611066.

[18] P. Zhi, P. Li, J. Yin, B. Jia, S. Huang, Learning a unified policy for position and force control in legged loco-manipulation, in: Proceedings of The 9th Conference on Robot Learning, Vol. 305 of Proceedings of Machine Learning Research, PMLR, 2025, pp. 652–669.
URL https://proceedings.mlr.press/v305/zhi25a.html

[19] D. Li, X. Chen, Q. Wu, B. Chen, S. Wu, H. Wu, G. Zhang, L. Li, M. Zhou, D. Xiang, J. Ma, Q. Zhang, R. Xu, HAIC: Humanoid agile object interaction control via dynamics-aware world model, in: Robotics: Science and Systems XXII, 2026. doi:10.15607/rss.2026.xxii.013.

[20] X. Liu, Y. Dong, J. Wan, L. Deng, F. Hua, Y. Shen, M. Yu, G. Ma, C. Cheng, H. Song, H. Ding, Y. Yuan, DexAnyTwist: Learning general dexterous twisting with hybrid manipulation system identification, National Science Review (2026) nwag351doi:10.1093/nsr/nwag351.

[21] J. Wan, X. Liu, Y. Dong, DexReMoE: In-hand reorientation of general object via mixtures of experts, arXiv preprint arXiv:2508.01695 (2025). doi:10.48550/arXiv.2508.01695.

[22] X. Zeng, J. Wang, T. Zhang, L. Zhang, B. Xu, W. Xia, Z. Li, Y.-H. Liu, Force-aware reinforcement learning with hybrid sensorless force estimation for wheeled-legged loco-manipulation, arXiv preprint arXiv:2609.13779 (2026).
URL https://arxiv.org/abs/2609.13779

[23] S. Nozawa, Y. Kakiuchi, K. Okada, M. Inaba, Controlling the planar motion of a heavy object by pushing with a humanoid robot using dual-arm force control, in: 2012 IEEE International Conference on Robotics and Automation (ICRA), 2012, pp. 1428–1435. doi:10.1109/ICRA.2012.6224884.

[24] M. Murooka, K. Chappellet, A. Tanguy, M. Benallegue, I. Kumagai, M. Morisawa, F. Kanehiro, A. Kheddar, Humanoid loco-manipulations pattern generation and stabilization control, IEEE Robotics and Automation Letters 6 (3) (2021) 5597–5604. doi:10.1109/LRA.2021.3077858.

[25] J. Li, L. Wu, S. A. Esteban, L. Yang, J. Drgoňa, A. D. Ames, Accelerating and scaling MPC-guided reinforcement learning for humanoid locomotion and manipulation, arXiv preprint arXiv:2606.05687 (2026). doi:10.48550/arXiv.2606.05687.

[26] A. Dang, A. Flores Alvarez, Y.-M. Chen, C. Mc Gartoll, H. Sun, A. Ames, N. Fazeli, M. Nambi, HULK: Learning whole-body forceful loco-manipulation for humanoids, arXiv preprint arXiv:2610.08970 (2026). doi:10.48550/arXiv.2610.08970.

[27] H. Jiang, J. Chen, Q. Bu, L. Chen, M. Shi, Y. Zhang, D. Li, C. Suo, C. Wang, Z. Peng, H. Li, WholeBodyVLA: Towards unified latent VLA for whole-body loco-manipulation control, in: The Fourteenth International Conference on Learning Representations, 2026.
URL https://arxiv.org/abs/2512.11047

[28] Z. Hu, Z. Xu, D. Chang, H. Yin, L. Tran, R. Martín-Martín, P. Stone, J. Qiao, J. Biswas, VOFA: Visual object goal pushing with force-adaptive control for humanoids, in: 2026 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2026.
URL https://www.cs.utexas.edu/~AustinVilla/papers/zichao_hu_iros_2026.pdf

[29] H. Zhang, Y. Tseng, D. Zhao, H. E. Tseng, Interaction-aware whole-body control for compliant object transport, arXiv preprint arXiv:2603.03751 (2026). doi:10.48550/arXiv.2603.03751.

[30] B. Xu, H. Weng, Q. Lu, Y. Gao, H. Xu, FACET: Force-adaptive control via impedance reference tracking for legged robots, arXiv preprint arXiv:2505.06883 (2025). doi:10.48550/arXiv.2505.06883.

[31] G. B. Margolis, M. Wang, N. Fey, P. Agrawal, SoftMimic: Learning compliant whole-body control from examples, arXiv preprint arXiv:2510.17792 (2025). doi:10.48550/arXiv.2510.17792.

[32] Q. Lu, Y. Feng, B. Shi, M. Piseno, Z. Bao, C. K. Liu, GentleHumanoid: Learning upper-body compliance for contact-rich human and object interaction, IEEE Robotics and Automation Letters (2026) 1–8doi:10.1109/lra.2026.3747618.

[33] C. Dong, H. Xu, W. Feng, Z. Wang, Y. Zhou, Y. Zhao, B. He, HAFO: A force-adaptive control framework for humanoid robots in intense interaction environments, arXiv preprint arXiv:2511.20275 (2025). doi:10.48550/arXiv.2511.20275.

[34] G. Li, H. Shi, Q. Shi, Y. Hu, C. Ma, Z. Wang, X. Wang, S. Luo, Thor: Towards human-inspired whole-body reactions for intense contact-rich environments, arXiv preprint arXiv:2510.26280 (2025). doi:10.48550/arXiv.2510.26280.

[35] T.-D. Do, C. T. Trinh, T. D. Phuong, C. Le, T. Ly, V. A. Ngo, A. T. Le, CompliantWBC: Whole-body compliance for heavy humanoids via force latent estimation and residual impedance targets, arXiv preprint arXiv:2609.33310 (2026). doi:10.48550/arXiv.2609.33310.

[36] T. He, Z. Luo, W. Xiao, C. Zhang, K. Kitani, C. Liu, G. Shi, Learning human-to-humanoid real-time whole-body teleoperation, in: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2024, pp. 8944–8951. doi:10.1109/iros58592.2024.10801984.

[37] T. He, Z. Luo, X. He, W. Xiao, C. Zhang, W. Zhang, K. M. Kitani, C. Liu, G. Shi, OmniH2O: Universal and dexterous human-to-humanoid whole-body teleoperation and learning, in: Proceedings of The 8th Conference on Robot Learning, Vol. 270 of Proceedings of Machine Learning Research, PMLR, 2025, pp. 1516–1540.
URL https://proceedings.mlr.press/v270/he25b.html

[38] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, C. Finn, HumanPlus: Humanoid shadowing and imitation from humans, in: Proceedings of The 8th Conference on Robot Learning, Vol. 270 of Proceedings of Machine Learning Research, PMLR, 2025, pp. 2828–2844.
URL https://proceedings.mlr.press/v270/fu25a.html

[39] T. Johannink, S. Bahl, A. Nair, J. Luo, A. Kumar, M. Loskyll, J. A. Ojea, E. Solowjow, S. Levine, Residual reinforcement learning for robot control, in: 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 6023–6029. doi:10.1109/icra.2019.8794127.

[40] Z. Sun, Y. Peng, Y. Meng, X. Li, B.-S. Huang, Z. Bing, X. Wang, A. Knoll, RobotDancing: Residual-action reinforcement learning enables robust long-horizon humanoid motion tracking, IEEE Robotics and Automation Letters 11 (10) (2026) 11267–11274. doi:10.1109/lra.2026.

3723309.

[41] Z. Sun, B.-S. Huang, Y. Peng, X. Li, J. Ma, Y. Sun, Z. Li, H. Jiang, B. Gao, Z. Bing, X. Wang, A. Knoll, MOSAIC: Bridging the sim-to-real gap in generalist humanoid motion tracking and teleoperation with rapid residual adaptation, arXiv preprint arXiv:2602.08594 (2026). doi:10.48550/arXiv.2602.08594.

[42] H. Weng, Y. Li, N. Sobanbabu, Z. Wang, Z. Luo, T. He, D. Ramanan, G. Shi, HDMI: Learning interactive humanoid whole-body control from human videos, arXiv preprint arXiv:2509.16757 (2025). doi:10.48550/arXiv.2509.16757.

[43] J. Jang, Z. Wang, Z. Zhou, F. Wu, Y. Zhao, SEEC: Stable end-effector control with model-enhanced residual learning for humanoid loco-manipulation, arXiv preprint arXiv:2509.21231 (2025).
URL https://arxiv.org/abs/2509.21231

[44] A. Huang, Z. Wu, S. Atar, Y. Zhi, M. Yip, SteadyTray: Learning object balancing tasks in humanoid tray transport via residual reinforcement learning, arXiv preprint arXiv:2603.10306 (2026). doi:10.48550/arXiv.2603.10306.

[45] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, A. Garg, Orbit: A unified simulation framework for interactive robot learning environments, IEEE Robotics and Automation Letters 8 (6) (2023) 3740–3747. doi:10.1109/lra.2023.3270034.

[46] E. Todorov, T. Erez, Y. Tassa, Mujoco: A physics engine for model-based control, in: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012, pp. 5026–5033. doi:10.1109/iros.2012.6386109.