ADP: Adversarial Dynamics Priors for Physically Grounded Humanoid Locomotion

Seokju Lee[1*], Jeongtae Lee[1*], Jeonghyeok Lim[1], Jeonguk Kang[2], Byungwook Lee[1], Seungho Han[3], Keun Ha Choi[1], Dongil Park[4,†], and Kyung-Soo Kim[1,†]

***Abstract*— In this paper, we propose Adversarial Dynamics Priors (ADP) for perturbation-resilient humanoid locomotion control. Existing motion prior-based methods induce natural motion styles by imitating kinematic motion features, but they do not directly regularize dynamics features, such as CoM motion, centroidal momentum, contact forces, and contact states. To address this limitation, we replace kinematic motion-style feature with selected dynamics features extracted from locomotion trajectories as the target of adversarial regularization. To this end, we use trajectory optimization to construct a reference dataset and train a discriminator to evaluate whether policy-induced temporal windows are consistent with the resulting reference distribution. Without explicit motion tracking, ADP encourages policy rollouts to remain close to the reference support, even after perturbations. Experimental results show that, compared with AMP, the strongest baseline in our evaluation, ADP improves the 80%-success impulse threshold (J_{80}) by 16.7%, while reducing direction-averaged recovery time and velocity tracking error by 47.9% and 35.4%, respectively.**

I. INTRODUCTION

Humanoid robots have recently attracted increasing attention as robotic platforms capable of operating in human-centric environments and performing tasks on behalf of humans [1]–[3]. For controlling such high-dimensional systems, learning-based control has become a dominant paradigm, building on its remarkable success in quadrupedal locomotion [4]–[6]. Nevertheless, humanoid locomotion remains substantially more challenging than quadrupedal locomotion due to the increased number of degrees of freedom, the underactuated floating-base dynamics, and the need to maintain balance with intermittent contacts. Consequently, recent approaches have moved beyond purely hand-designed reward functions and have instead leveraged motion datasets to guide policy learning, either through explicit reference-tracking rewards or through motion priors [7]–[10].

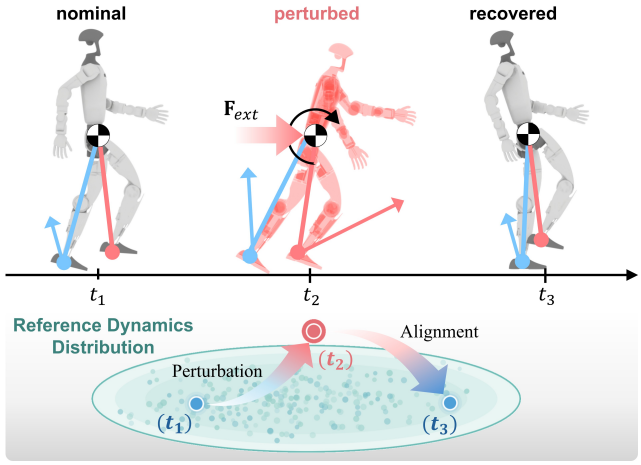


Fig. 1. Dynamics-feature alignment for perturbation recovery. An external force at t_2 perturbs the humanoid and drives the generated dynamics features away from the reference distribution. ADP encourages recovery by regularizing the perturbed windows toward this distribution, resulting in stable locomotion at t_3.

A representative approach for leveraging motion datasets is to design an imitation reward that directly tracks a reference motion. For example, DeepMimic [11] trains a policy to follow a given reference motion using tracking terms such as pose, velocity, end-effector, and root-state matching. Although this tracking-based formulation is effective for generating natural motions, its reliance on phase variables or target poses can limit policy flexibility when external perturbations drive the robot away from the reference trajectory. To mitigate this limitation, AMP [12] formulates motion imitation as a distribution matching problem rather than explicit tracking, and uses a discriminator-based style reward to encourage the policy to generate a kinematic motion-style distribution similar to the motion dataset. However, the AMP prior is still based on kinematic motion features such as joint pose, velocity, and end-effector state, and therefore does not directly regularize dynamics-level recovery behaviors required after perturbations, such as momentum regulation, contact-force modulation, and contact timing.

In this paper, we propose Adversarial Dynamics Priors (ADP), which leverages dynamics features obtained from Trajectory Optimization (TO) to address the limitations of motion-style priors. Instead of directly imitating the kinematic style of reference motions, ADP provides an adversar-

*Equal contribution.

This work was supported by R&D Project of Korea Institute of Machinery and Materials (No. NK261A). *(Corresponding author: Dongil Park and Kyung-Soo Kim)*

[1]Seokju Lee, Jeongtae Lee, Jeonghyeok Lim, Byungwook Lee, Keun Ha Choi, Kyung-Soo Kim are with the Mechatronics, Systems and Control Lab (MSC Lab), Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Yuseong-gu, Daejeon 34141, Republic of Korea (e-mail: {dltjrwn0322, us03316, henricus0973, bwlee605, choiha99, kyungsookim}@kaist.ac.kr)

[2]Jeonguk Kang is with the Samsung Electronics, Future Robotics AI Group, Seoul, Republic of Korea. (e-mail: kju8765@gmail.com)

[3]Seungho Han is with the School of Electrical Engineering, Hanyang University, Ansan 15588, Republic of Korea. (email: seungho-han@hanyang.ac.kr)

[4]Dongil Park is with the Advanced Robotics Research Center, Korea Institute of Machinery & Materials (KIMM), Daejeon 34103, Republic of Korea. (e-mail: parkstar@kimm.re.kr)

Paper website: `https://seokju-lee.github.io/adp`

ial reward that keeps policy-induced dynamics features close to a reference distribution. We construct $\mathscr{D}_{\text{dyn}}$ by solving a Single-Rigid-Body-Dynamics (SRBD)-based TO problem and extracting dynamics features from the resulting locomotion trajectories. The resulting samples satisfy SRBD-level dynamics and contact constraints, forming the reference set used for adversarial training. During policy training, the same feature type is computed from rollouts, and a discriminator is trained to distinguish reference windows from policy-generated windows. Conversely, the policy is optimized to maximize both the task reward and the discriminator-based dynamics prior reward, thereby learning recovery behaviors that remain close to the reference distribution even after external perturbations, as shown in Fig. 1. The main contributions of this paper are as follows:

- **Adversarial Dynamics Priors:** We propose ADP, which replaces kinematic motion-style features with selected dynamics features for adversarial regularization, including CoM motion, centroidal momentum, contact forces, and contact states, without reference pose, phase, or end-effector tracking.
- **Perturbation-Sensitive Dynamics Representation:** We formulate a temporal dynamics-feature representation that exposes post-push transients in CoM motion, centroidal momentum, contact forces, and contact timing more directly than joint-level kinematic features.
- **Perturbation Recovery Evaluation:** We show through simulation comparisons with Vanilla RL, AMP, and direct dynamics-feature rewards that ADP improves humanoid perturbation recovery, and provide qualitative hardware demonstrations on a real humanoid robot under external perturbations.

The remainder of this paper is organized as follows. Section II reviews prior studies on learning policy networks using motion datasets and dynamics datasets. Section III describes the proposed method in detail. Section IV presents the experimental setup and results. Finally, Section V concludes the paper and discusses future directions.

II. RELATED WORK

A. Motion Imitation for Robot Control

Recent robot control has widely adopted motion imitation methods to learn natural behaviors for complex articulated systems. In particular, for legged robots, designing hand-designed rewards alone is often insufficient to induce natural locomotion, and many studies have leveraged motion datasets from animals or humans to assist policy learning. Escontrela *et al.* [13] replaced complex locomotion reward design with an AMP-based method that learns an adversarial style reward from a motion dataset, demonstrating natural and energy-efficient locomotion on a quadruped robot. Wu *et al.* [14] further used a motion dataset generated by SRBD-based TO to learn an AMP style reward, and combined it with task and regularization rewards to achieve robust and agile quadruped locomotion over diverse terrains.

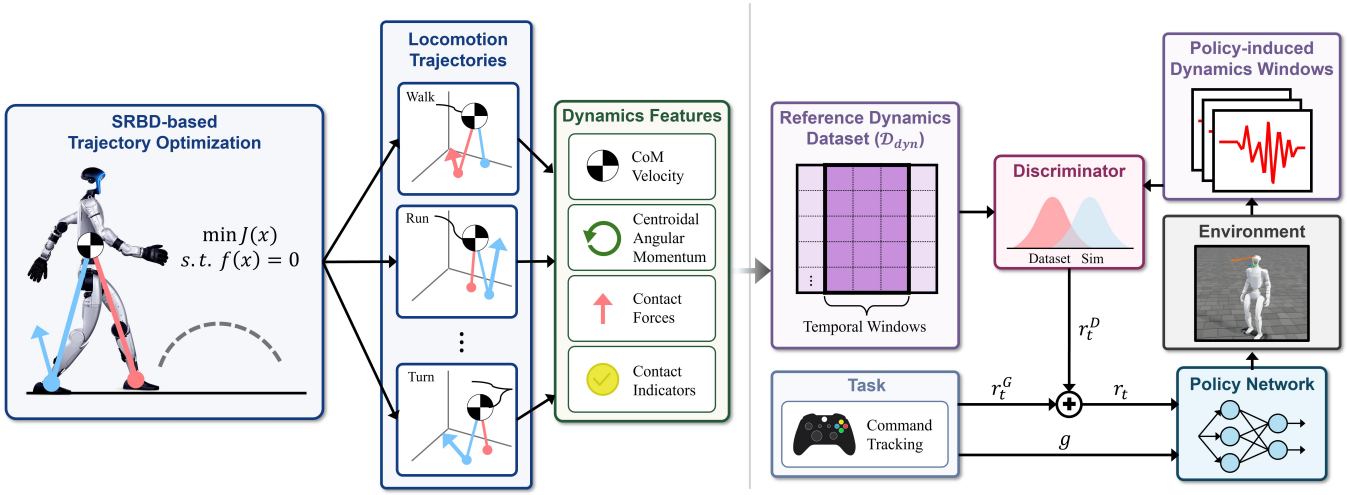
These motion imitation approaches have also become an important direction for humanoid robot control. Since humanoids have high degrees of freedom and unstable floating-base dynamics, it is difficult to learn natural and stable whole-body behaviors using simple task rewards alone. Accordingly, many studies explicitly track kinematic features of reference motions, such as joint positions, joint velocities, root states, and end-effector positions, or use adversarial imitation learning to match kinematic motion-style distributions [15]. While these methods are effective in generating human-like natural behaviors, their priors are still largely based on kinematic motion features. However, perturbation recovery requires not only kinematic motion-style similarity but also appropriate momentum regulation and contact interaction for restoring balance. This motivates replacing kinematic motion-style matching with dynamics-level adversarial regularization.

To address this limitation, ADP replaces kinematic motion imitation with adversarial dynamics-feature regularization, encouraging policy-generated dynamics features to remain close to a reference dynamics-feature distribution even after external perturbations.

B. Dynamics-Guided Locomotion Learning

Dynamics-based control has long provided a physical basis for stable legged locomotion. Centroidal dynamics represents complex multibody motion through CoM motion and centroidal momentum, and has been widely used for humanoid motion planning and balance control [16]. Building on this representation, WBC, MPC, and TO methods incorporate full-body kinematics, contact constraints, and friction constraints to generate stable and dynamic motions for humanoid and quadruped robots [17]–[23]. Related approaches have further used DCM or ZMP reference adaptation for push and stumble recovery, compliant walking-pattern generation, and whole-body motion generation under inertia constraints [24]–[26]. However, these methods often depend on explicit dynamics models, contact-state estimation, prescribed contact schedules, online optimization, or separate tracking controllers, which can limit rapid feedback adaptation under model mismatch or unexpected contact changes.

To mitigate these limitations, recent studies have attempted to incorporate the physical structure of model-based dynamics into learning-based locomotion. For example, TO-generated demonstrations have been used to alleviate the initial exploration problem in RL [27], and LIP-dynamics-based footstep guidance has been incorporated into RL policy learning [28]. Other approaches have used contact-consistent whole-body trajectories generated from full-order dynamics-based TO as imitation targets [2], or embedded a differentiable simulator into a generative model to produce physically consistent state-action trajectories [29]. Although DynaFlow ensures the dynamic feasibility of generated trajectories, its dynamics model is embedded within the generative process to synthesize feasible motions, rather than used as a prior for evaluating the closed-loop dynamics produced by an RL policy. More recently, APT-RL demonstrated that large-scale motion and torque datasets generated by simplified-dynamics-based TO can provide effective priors for learning



(a) Offline reference dynamics dataset generation using trajectory optimization

(b) Adversarial policy learning using dynamics priors

Fig. 2. Overview of the proposed Adversarial Dynamics Priors (ADP) framework. (a) SRBD-based trajectory optimization generates reference locomotion trajectories, from which dynamics features are extracted to construct $\mathcal{D}_{\text{dyn}}$. (b) During policy training, policy-generated windows are evaluated by a discriminator trained with $\mathcal{D}_{\text{dyn}}$, producing the dynamics prior reward r_t^D. The policy is trained with the task reward r_t^G, dynamics prior reward r_t^D, and regularization reward r_t^R.

reusable locomotion skills and improving the efficiency of downstream RL [30]. Motivated by this observation, ADP similarly constructs a TO-derived dataset, but uses the resulting dynamics distribution as a discriminator-defined prior for evaluating policy-generated temporal features, rather than as a trajectory reference, action decoder, footstep guide, or planning constraint.

III. METHOD

A. Problem Formulation

We train the humanoid locomotion policy with an asymmetric actor-critic formulation. The actor receives proprioceptive observations, including IMU signals, projected gravity, joint states, previous actions, and the velocity command. It outputs target position offset actions for the active joints. Base linear velocity is excluded from the actor input for real-world deployability, but is provided to the critic as privileged information during training.

The policy is trained to perform the locomotion task while regularizing its policy-induced features toward the TO-derived reference distribution. To this end, the total reward is defined as

$$r_t = w_G r_t^G + w_D r_t^D + w_R r_t^R, \tag{1}$$

where r_t^G is the task reward, r_t^D is the proposed adversarial dynamics prior reward, and r_t^R denotes regularization terms.

B. Reference Trajectory Generation

To construct the reference dataset for ADP training, we perform SRBD-based TO. We use SRBD as a low-dimensional TO model that represents the robot by its CoM dynamics and foot contact forces. The SRBD equations are defined as

$$m\ddot{\mathbf{p}}_t^{\text{com}} = m\mathbf{g} + \sum_{i\in\{L,R\}} \mathbf{f}_t^i, \tag{2a}$$

$$\dot{\mathbf{L}}_t^{\text{com}} = \sum_{i\in\{L,R\}} \left(\mathbf{p}_t^i - \mathbf{p}_t^{\text{com}}\right) \times \mathbf{f}_t^i, \tag{2b}$$

where m is the total robot mass, $\mathbf{p}_t^{\text{com}}$ is the CoM position, $\mathbf{L}_t^{\text{com}}$ is the centroidal angular momentum, $\mathbf{p}_t^i$ is the contact position of foot i, $\mathbf{f}_t^i$ is the corresponding contact force, and $i \in \{L,R\}$ denote the left and right foot contact points.

TO is used to generate reference locomotion trajectories from the SRBD model. Rather than serving as full-body motion references, the optimized trajectories provide SRBD-consistent feature samples over a range of commands. The TO problem is formulated as

$$\begin{aligned}
\min_{\mathbf{x}_{0:T},\mathbf{u}_{0:T}} \quad & \sum_{t=0}^{T} \left\| \mathbf{v}_t^{xy,\text{com}} - \mathbf{v}^{xy,\text{cmd}} \right\|_{\mathbf{Q}_v}^2 + \|\mathbf{L}_t^{\text{com}}\|_{\mathbf{Q}_L}^2 + \sum_{i\in\{L,R\}} \left\|\mathbf{f}_t^i\right\|_{\mathbf{R}_f}^2 \\
\text{s.t.} \quad & m\ddot{\mathbf{p}}_t^{\text{com}} = m\mathbf{g} + \sum_{i\in\{L,R\}} \mathbf{f}_t^i, \\
& \dot{\mathbf{L}}_t^{\text{com}} = \sum_{i\in\{L,R\}} \left(\mathbf{p}_t^i - \mathbf{p}_t^{\text{com}}\right) \times \mathbf{f}_t^i, \\
& 0 \le f_t^{i,z} \le F_{\max}\ \rho_t^i, \quad \sqrt{(f_t^{i,x})^2 + (f_t^{i,y})^2} \le \mu f_t^{i,z}, \\
& 0 \le \rho_t^i \le c_t^i, \quad c_t^i \in \{0,1\}.
\end{aligned} \tag{3}$$

Here, $\mathbf{x}_t$ includes the CoM state and centroidal momentum, while $\mathbf{u}_t$ includes the contact forces. The objective encourages the horizontal CoM velocity to follow the commanded velocity while penalizing excessive centroidal angular momentum and contact-force magnitudes. The constraints enforce SRBD consistency, contact-force activation, and friction-cone limits, producing trajectories that are physically grounded at the SRBD level.

For turning motions, the commanded yaw rate ω_z^{cmd} is integrated into a time-varying heading frame in which the horizontal CoM velocity objective is evaluated; thus, no separate yaw-rate tracking term is required in (3). The TO trajectories are not used as full-body tracking targets; they only provide the reference set against which policy-generated windows are evaluated.

The dataset consists of multiple locomotion modes, including forward running, walking, turning, lateral stepping, and backward walking. Each motion is generated as an SRBD-consistent locomotion trajectory under a fixed linear and yaw velocity command, from which we extract dynamics quantities such as CoM motion, centroidal momentum, contact forces, and contact indicators. The contact schedule $\{c_t^L, c_t^R\}$ follows a per-motion prescribed gait schedule and is used as the binary contact indicator in each reference window. The continuous variable ρ_t^i in (3) only serves as a soft contact activation for scaling normal forces during optimization. Thus, $\mathscr{D}_{dyn}$ stores dynamics features rather than motion-tracking targets.

C. Dynamics Feature Representation

ADP compares TO-derived reference trajectories and policy rollouts in a shared feature space. To this end, we construct a dynamics feature at each timestep that includes CoM motion, centroidal momentum, contact forces, and contact states. This representation is separated from the actor observation and is used only as the discriminator input, not as the actor input.

The timestep-level dynamics feature is defined as

$$\mathbf{z}_t = [\mathbf{v}_{t,h}^{\text{com}},\ \mathbf{L}_{t,h}^{\text{com}},\ \dot{\bar{\mathbf{L}}}_{t,h}^{xy,\text{com}},\ \hat{\mathbf{f}}_{t,h}^{L},\ \hat{\mathbf{f}}_{t,h}^{R},\ c_t^L,\ c_t^R]. \tag{4}$$

Here, the subscript h denotes the heading frame. $\mathbf{v}_{t,h}^{\text{com}}$ denotes the heading-frame CoM velocity, $\mathbf{L}_{t,h}^{\text{com}}$ denotes the centroidal angular momentum, and $\dot{\mathbf{L}}_{t,h}^{xy,\text{com}}$ denotes the planar angular momentum rate. In addition, $\hat{\mathbf{f}}_{t,h}^{L}$ and $\hat{\mathbf{f}}_{t,h}^{R}$ denote the left and right foot contact forces normalized by the robot weight, and c_t^L, c_t^R denote binary contact indicators.

For policy rollouts, the centroidal angular momentum is computed by summing the link-wise spin momentum and the orbital momentum about the system CoM over all links of the articulated humanoid model:

$$\mathbf{L}_t^{\text{com}} = \sum_i \left(\mathbf{I}_{i,w}\boldsymbol{\omega}_i + (\mathbf{p}_i - \mathbf{p}^{\text{com}}) \times m_i(\mathbf{v}_i - \mathbf{v}^{\text{com}})\right). \tag{5}$$

This captures whole-body momentum redistribution in policy rollouts, while SRBD-based reference trajectories are converted into the same feature interface.

Because the same dynamics feature can correspond to different behaviors under different velocity commands, the discriminator input is conditioned on both the current command and the instantaneous tracking error. These quantities enter as conditioning variables that align the reference distribution with the current command, rather than as tracking-error reward terms. Given the command $\mathbf{v}_t^{\text{cmd}} = [v_{x,t}^{\text{cmd}}, v_{y,t}^{\text{cmd}}, \omega_{z,t}^{\text{cmd}}]$, the command-conditioned dynamics feature is defined as

$$\xi_t = \left[\mathbf{z}_t,\ \mathbf{v}_t^{\text{cmd}},\ \mathbf{e}_t\right]. \tag{6}$$

Here, $\mathbf{e}_t$ denotes the planar velocity and yaw-rate tracking error:

$$\mathbf{e}_t = \left[v_{x,t}^{\text{com}} - v_{x,t}^{\text{cmd}},\ v_{y,t}^{\text{com}} - v_{y,t}^{\text{cmd}},\ \omega_{z,t} - \omega_{z,t}^{\text{cmd}}\right]. \tag{7}$$

Contact switching, force modulation, and momentum recovery cannot be sufficiently represented by a single timestep. Therefore, ADP constructs a short temporal window:

$$\Xi_t = \left[\bar{\xi}_{t-K+1}, \bar{\xi}_{t-K+2}, \cdots, \bar{\xi}_t\right], \tag{8}$$

where $\bar{\xi}_t$ denotes the normalized dynamics feature and K is the window length. The discriminator receives the flattened vector $\text{vec}(\Xi_t)$ as input.

Using this interface, TO-derived trajectories and policy rollouts are converted into reference and policy-generated windows, respectively. The reference dataset is defined as $\mathscr{D}_{\text{dyn}} = \left\{\Xi_j^{\text{ref}}\right\}_{j=1}^{N}$. The corresponding rollout window at timestep t is denoted by Ξ_t^{π}. Although the reference windows are obtained from SRBD-based TO and the policy windows from articulated humanoid rollouts, both are represented in the same feature space. Therefore, ADP can compare the reference and policy-generated window distributions without using reference joint poses, motion phases, or end-effector trajectories. This comparison is intentionally defined in the selected feature space: the SRBD-derived reference provides SRBD-consistent features, but does not impose full-body joint-level kinematic targets or guarantee full-order dynamic feasibility of every articulated configuration.

D. Training

We jointly perform PPO [31]-based actor-critic learning and adversarial dynamics discriminator learning. During training, rollout windows Ξ_t^{π} are computed from policy rollouts and stored in a replay buffer $\mathscr{B}_{\pi}$. The discriminator is trained to distinguish rollout windows sampled from $\mathscr{B}_{\pi}$ from reference windows sampled from $\mathscr{D}_{\text{dyn}}$.

The discriminator uses normalized windows as input, rather than single state transitions. It is optimized with the following least-squares adversarial objective:

$$\begin{aligned}\mathscr{L}_D(\phi) =& \frac{1}{2}\mathbb{E}_{\Xi^{\text{ref}}\sim\mathscr{D}_{\text{dyn}}}\left[\left(D_\phi(\Xi^{\text{ref}}) - 1\right)^2\right] \\ &+ \frac{1}{2}\mathbb{E}_{\Xi^{\pi}\sim\mathscr{B}_{\pi}}\left[\left(D_\phi(\Xi^{\pi}) + 1\right)^2\right] \\ &+ \lambda_{\text{gp}}\mathbb{E}_{\Xi^{\text{ref}}\sim\mathscr{D}_{\text{dyn}}}\left[\left\|\nabla_{\Xi} D_\phi(\Xi^{\text{ref}})\right\|^2\right].\end{aligned} \tag{9}$$

The first two terms classify reference and policy-generated windows as +1 and -1, respectively, while the gradient penalty stabilizes training around the reference distribution.

The policy receives the ADP reward from the discriminator output. For a policy-generated window Ξ_t^{π}, the reward is defined as

$$r_t^D = c_D \max\left[0,\ 1 - \frac{1}{4}\left(D_\phi(\Xi_t^{\pi}) - 1\right)^2\right], \tag{10}$$

where c_D is the ADP reward scale. This reward increases when the policy-generated window is classified as close to the reference distribution.

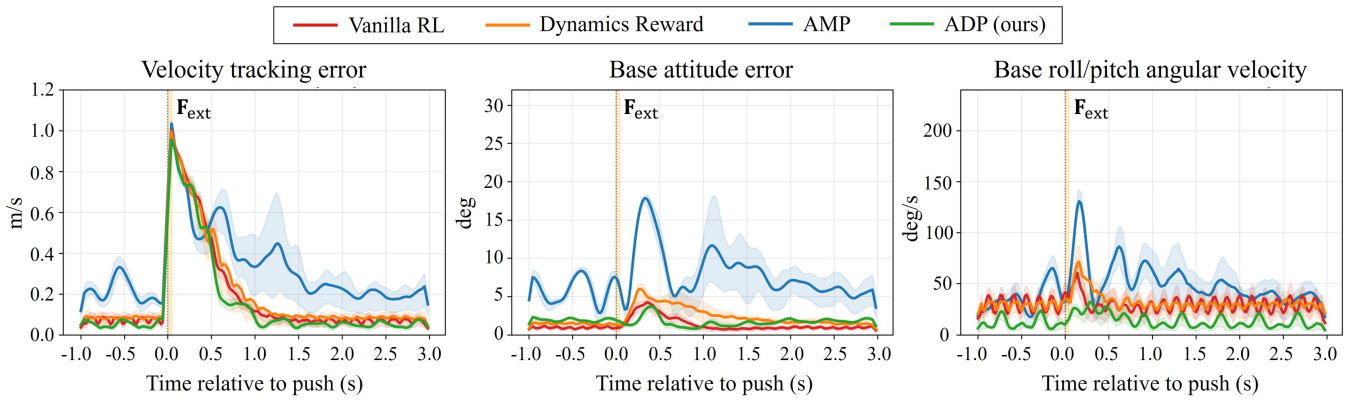


Fig. 3. Time-series perturbation recovery under a lateral push. Red, orange, blue, and green curves denote Vanilla RL, Dynamics Reward, AMP, and ADP, respectively. Lines show the mean and translucent bands show the Interquartile Range (IQR) over 32 environments; the panels show velocity tracking error, base attitude error, and base roll/pitch angular velocity.

The actor and critic are updated using PPO with the total reward defined in (1). In contrast, the discriminator is updated using $\mathscr{L}_D$ with samples from $\mathscr{D}_{\text{dyn}}$ and $\mathscr{B}_\pi$. The discriminator loss is not directly backpropagated through the simulator dynamics into the actor. Instead, the actor is affected by the dynamics prior only through the reward r_t^D. Thus, ADP affects the actor only through the reward r_t^D, without directly tracking reference joint poses, velocities, or end-effector trajectories.

IV. EXPERIMENTS

We quantitatively evaluate ADP in simulation and provide qualitative hardware demonstrations on the Unitree G1 [32], a 29-DoF humanoid robot. The supplementary video provides side-by-side hardware comparison between ADP and AMP under manually applied external pushes. These hardware trials are intended as qualitative transfer validation rather than as a controlled quantitative evaluation. All methods share the same PPO implementation, architecture, observations, actions, command distribution, domain randomization, and perturbation curriculum. Auxiliary reward scales for non-vanilla methods are tuned using the same preliminary perturbation-recovery criterion and then fixed for all reported evaluations. The source code and experiment configurations will be released on the paper website. We compare ADP against Vanilla RL, which is trained only with task rewards without using a dataset; AMP [12], which learns a kinematic motion prior from reference data generated under the same locomotion set; and a Dynamics Reward baseline, which directly uses dynamics-feature matching rewards without an adversarial discriminator. The Dynamics Reward baseline uses the same feature set and reference data as ADP, but replaces the discriminator with a cluster-based matching reward

$$r_t^{\text{DR}} = \exp\left(-\tau \sum_k w_k(\mathbf{v}_t^{\text{cmd}}) \left\| (\Xi_t^\pi - \mu_k) \oslash \sigma_k \right\|_2^2\right), \quad (11)$$

where $\{\mu_k, \sigma_k\}$ denote the per-feature mean and standard deviation of the k-th reference motion cluster, $\oslash$ denotes element-wise division, and $w_k(\mathbf{v}_t^{\text{cmd}}) \propto \exp(-\|(\mathbf{v}_t^{\text{cmd}} - \mathbf{c}_k) \oslash \sigma_{\text{cmd}}\|^2)$ weights the k-th reference motion cluster according to its command-space proximity to $\mathbf{v}_t^{\text{cmd}}$. Here, $\mathbf{c}_k$ is the command center of the k-th cluster, and τ is a temperature. The Dynamics Reward thus measures a point-wise Gaussian-style distance from the policy window Ξ_t^π to the reference cluster centers, whereas ADP uses an adversarial discriminator to provide a distribution-level reward over reference windows. For a fair comparison, AMP and ADP use reference data generated from the same locomotion trajectories. This design controls the reference-source variable, so that the comparison isolates the effect of the prior representation rather than differences in demonstration quality. For AMP, following the TO-derived motion source used by Wu *et al.* [14], these trajectories are converted into full-body kinematic motions through inverse kinematics, and the AMP discriminator receives joint-level kinematic features, including joint pose, joint velocity, and end-effector transforms. In contrast, ADP uses only command-conditioned windows extracted from the same trajectory sources, so that the structural difference between the two methods is the prior feature space rather than the reference source. The evaluation focuses on the following three questions:

- **Q1:** Does ADP improve perturbation recovery performance after external disturbances?
- **Q2:** Does the proposed representation expose perturbation-induced transients more directly than a kinematic representation?
- **Q3:** Which dynamics features and temporal representations contribute to the performance improvement of ADP?

A. Perturbation Recovery under External Disturbances

We apply an instantaneous velocity impulse to the floating base in four directions (lateral $\pm y$, forward $+x$, backward $-x$) to assess generalization across disturbance directions. J_{80} is determined by sweeping Δv from 1.5 m/s to 4.5 m/s at 0.5 m/s intervals over the four push directions. Per-direction success rate, recovery time, and velocity tracking error are then measured under a unified push magnitude of

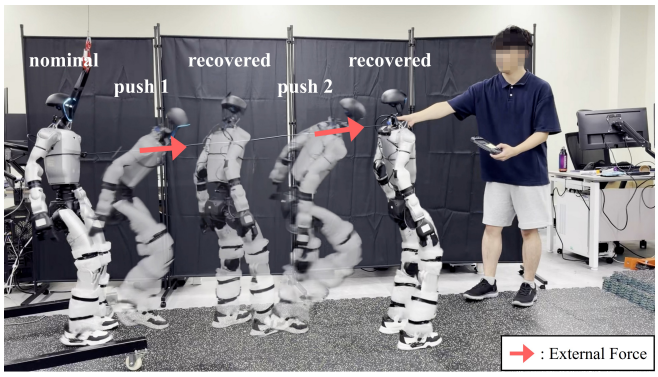


Fig. 4. Qualitative hardware demonstration of ADP on the real robot under repeated external pushes. The overlaid snapshots show the robot transitioning from nominal walking to a perturbed state and recovering under subsequent perturbations.

$\Delta v = 3.0$ m/s (impulse $= 99$ N·s), and Table I reports the corresponding direction-averaged metrics together with J_{80}.

The metrics in Table I are defined as follows. **Success Rate** is the fraction of trials without fall termination during the post-push window, where a fall is declared if any of the simulator's termination conditions is triggered: contact at a designated termination body (the torso), base height below 0.30 m, or a lateral projected-gravity component (along the body x- or y-axis) exceeding 0.8, which corresponds to an absolute pitch or roll of about 53°. **Recovery Time** measures how quickly the robot returns below fixed velocity and attitude error thresholds after the push, and **Velocity Error** is the post-push RMS planar velocity tracking error. Fallen trials are assigned fixed penalties for recovery time and velocity error. These metrics are averaged over four push directions. J_{80} (the 80%-success impulse threshold) is the largest impulse $J = m\Delta v$ for which the direction-averaged success rate remains above 80%, identified by sweeping Δv in 0.5 m/s increments.

Fig. 3 compares the recovery behaviors of the four policies under a representative lateral $\Delta v_y = 3.0$ m/s push. In this representative lateral-push case, ADP rapidly attenuates the transient velocity and attitude errors within approximately 1–1.5 s, stabilizing the velocity error below 0.2 m/s and the attitude error below 3°, while the formal direction-averaged recovery time is reported in Table I. AMP recovers velocity tracking, but its base attitude oscillates near 10–15° for almost 2 s, with angular velocity peaks exceeding 100°/s. This suggests that kinematic-style matching can recover the visible posture in this case, but may leave under-damped residual rotational dynamics.

Table I quantitatively summarizes the perturbation recovery performance. ADP achieves the highest J_{80} of 115.5 N·s and the highest direction-averaged success rate of 91.4% across the four push directions. It also reduces direction-averaged recovery time by 47.9% compared with AMP and by 74.5% compared with Dynamics Reward, while achieving the lowest velocity-error, indicating that the recovery improvement holds across push directions rather than being specialized to a single direction.

TABLE I

QUANTITATIVE PERTURBATION RECOVERY PERFORMANCE UNDER FOUR-DIRECTION EXTERNAL PUSHES.

Method	Success Rate (%) ↑	J_{80} (N·s) ↑	Recovery Time (s) ↓	Vel. Error (m/s) ↓
Vanilla RL	5.5	49.5	10.52	2.89
Dynamics Reward	14.1	66.0	9.73	2.68
AMP [12]	73.4	99.0	4.76	1.30
ADP (ours)	**91.4**	**115.5**	**2.48**	**0.84**

The comparison with AMP highlights the benefit of regularizing selected dynamics features rather than only matching kinematic motion style, yielding a 48% reduction in direction-averaged recovery time and a 35% reduction in velocity error. The comparison with the Dynamics Reward baseline further isolates the effect of replacing point-wise feature matching with distribution-level adversarial regularization over the same feature set, suggesting that the adversarial formulation provides a more effective signal under this setup. Fig. 4 qualitatively demonstrates repeated external-push recovery on the real robot.

B. Representation Sensitivity to External Perturbations

We analyze representation sensitivity to external perturbations by comparing how strongly the same post-push transient appears in the proposed and kinematic representations.

For the same perturbed rollouts, we compute normalized deviations from each representation's own reference distribution. $d_{\mathrm{dyn}}(t)$ is computed from the dynamics feature z_t in (4), excluding the command and tracking-error variables in (6). $d_{\mathrm{kin}}(t)$ is computed from the joint pose and joint velocity component of the AMP discriminator input. Each curve is normalized by its pre-push mean and computed from pooled non-fallen rollout episodes of all four methods under the same lateral-push condition.

Fig. 5 and Table II show that the proposed representation responds earlier and more strongly to the push: its deviation reaches 6.0× the pre-push baseline within 20 ms, whereas the kinematic representation reaches 4.3× with a 160 ms rise time. The higher post-push Area Under the Curve (AUC) and pre/post separability further indicate that the push transient is more salient in the proposed representation. This is consistent with the fact that the impulse directly excites floating-base momentum and contact forces before its effect is expressed through joint-level kinematics.

This analysis is not a task-level performance metric and does not imply that raw deviation is monotonic with the discriminator reward in (10). Fig. 5 and Table II show that perturbation-induced transients are exposed earlier and more strongly in the proposed representation. These results indicate that impulse-induced recovery transients are more directly observable in the proposed feature space than in joint-level kinematic features.

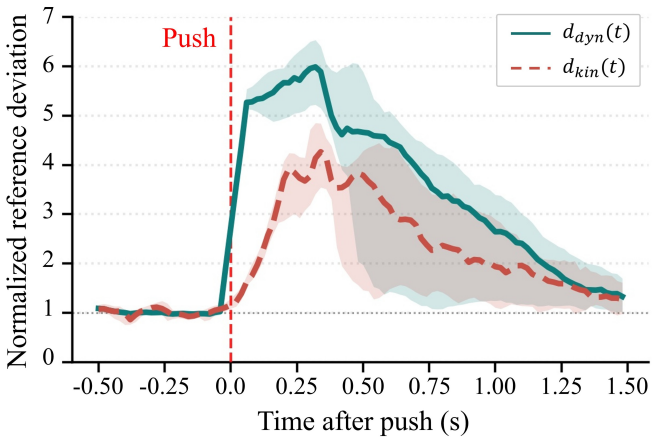


Fig. 5. Normalized reference deviation in dynamics-feature and kinematic-feature representations under a lateral $\Delta v_y = 3.0$ m/s push. Lines and bands denote medians and IQRs over 106 pooled non-fallen rollout episodes.

TABLE II

PERTURBATION SENSITIVITY OF THE DYNAMICS-FEATURE AND KINEMATIC-FEATURE REPRESENTATIONS.

Representation	Peak fold change ↑	50%-rise time (ms) ↓	Post-push AUC ↑	Pre/post sep. AUC ↑
Kinematic	4.3×	160	2.2	0.98
Dynamics	**6.0**×	**20**	**3.9**	**1.00**

TABLE III

SUMMARY OF DYNAMICS FEATURE COMPONENT ABLATION RESULTS.

Method	Success Rate (%) ↑	Vel. Error (m/s) ↓	Dyn. Dist. ↓	Recovery Time (s) ↓
ADP	**91.4**	**0.84**	**1.13**	**2.48**
ADP w/o Momentum	45.3	1.92	3.04	6.74
ADP w/o Contact Force	82.0	1.06	1.51	3.32
ADP w/o Contact Indicator	30.5	2.36	3.82	8.82

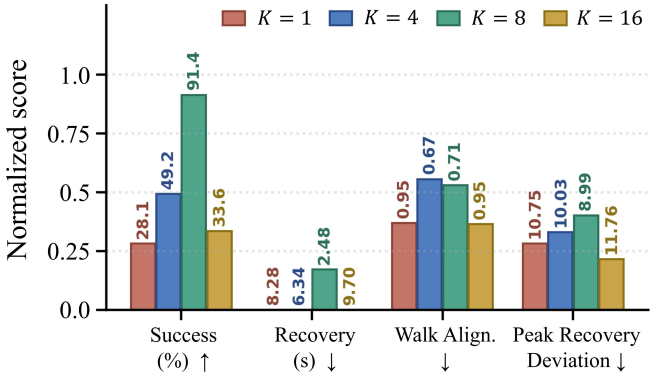


Fig. 6. Effect of temporal dynamics-feature window length K on normalized recovery metrics. Larger K provides temporal context for contact switching, force modulation, and momentum recovery, but overly long windows can degrade training stability. Lower-is-better metrics are inverted after normalization, so higher scores indicate better performance.

C. Ablation Study

We next ablate the discriminator input while keeping the PPO setting, reward weights, domain randomization, and reference dataset fixed. All variants are evaluated under four-direction $\Delta v = 3.0$ m/s pushes. In Table III, Dyn. Dist. denotes the post-push average $d_{\text{dyn}}(t)$ from Sec. IV-B, with fallen episodes assigned $Z_{\text{pen}} = 5$.

Table III summarizes the component ablation results. Full ADP achieves the best performance across all metrics (91.4% success). Removing the centroidal angular momentum reduces the success rate to 45.3% and raises the recovery time from 2.48 s to 6.74 s, while removing the contact force feature drops the success rate more modestly to 82.0%. This indicates that centroidal momentum and foot contact forces both contribute to post-perturbation recovery, with momentum playing the larger role under direction-averaged disturbances. The largest degradation arises from removing the binary contact indicator, which drops the success rate to 30.5% and increases the recovery time to 8.82 s, confirming that swing/stance transitions and contact timing are the most critical signals for alignment with the reference distribution.

Fig. 6 shows the performance variation with respect to the temporal window length K, with all metrics direction-averaged. When $K = 1$, the discriminator observes only single-timestep features and cannot sufficiently capture temporal patterns such as contact switching, force modulation, and momentum recovery, resulting in a direction-averaged success rate of only 28.1%. A moderate window length improves recovery by providing local temporal context, whereas an excessively large K increases the discriminator input dimension and can overemphasize timing differences between SRBD-based reference trajectories and feedback-driven articulated rollouts. Although $K = 4$ slightly improves walking-time alignment, it does not generalize consistently across push directions, resulting in a 49.2% direction-averaged success rate, whereas $K = 16$ further degrades to 33.6% due to long-horizon timing mismatch and the increased discriminator input dimension. In contrast, $K = 8$ achieves 91.4% direction-averaged success with the lowest recovery time (2.48 s) and the smallest peak recovery deviation, providing the best trade-off between temporal context and training stability. We therefore adopt $K = 8$ as the default temporal window for ADP.

V. CONCLUSIONS

In this work, we proposed Adversarial Dynamics Priors (ADP) for physically grounded humanoid locomotion control. ADP provides an adversarial reward over TO-derived features instead of directly tracking reference motions, encouraging the policy to learn stable perturbation recovery behaviors. Experiments show that ADP reduces direction-averaged recovery time and velocity tracking error compared with AMP and direct dynamics-reward baselines. A complementary representation-sensitivity analysis further shows that the proposed feature space exposes perturbation-induced transients earlier and more strongly than a joint-level kinematic space. Since ADP does not explicitly enforce joint-level motion naturalness, future work will investigate combining dynamics and kinematic priors to improve both robustness and naturalness. We also note that ADP is

compared against an AMP baseline using the same TO-derived reference source for fairness, rather than against a high-quality mocap-based AMP baseline; characterizing the gap between TO-derived and high-quality mocap-based kinematic priors is left to future work.

REFERENCES

[1] Y. Ze, Z. Chen, W. Wang, T. Chen, X. He, Y. Yuan, X. B. Peng, and J. Wu, "Generalizable humanoid manipulation with 3d diffusion policies," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2873–2880, IEEE, 2025.

[2] F. Liu, Z. Gu, Y. Cai, Z. Zhou, H. Jung, J. Jang, S. Zhao, S. Ha, Y. Chen, D. Xu, *et al.*, "Opt2skill: Imitating dynamically-feasible whole-body trajectories for versatile humanoid loco-manipulation," *IEEE Robotics and Automation Letters*, 2025.

[3] Y. Fu, F. Xie, C. Xu, J. Xiong, H. Yuan, and Z. Lu, "Demohlm: From one demonstration to generalizable humanoid loco-manipulation," *IEEE Robotics and Automation Letters*, 2026.

[4] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, p. eabc5986, 2020.

[5] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science robotics*, vol. 7, no. 62, p. eabk2822, 2022.

[6] S. Choi, G. Ji, J. Park, H. Kim, J. Mun, J. H. Lee, and J. Hwangbo, "Learning quadrupedal locomotion on deformable terrain," *Science Robotics*, vol. 8, no. 74, p. eade2256, 2023.

[7] T. He, J. Gao, W. Xiao, Y. Zhang, Z. Wang, J. Wang, Z. Luo, G. He, N. Sobanbabu, C. Pan, Z. Yi, G. Qu, K. Kitani, J. K. Hodgins, L. Fan, Y. Zhu, C. Liu, and G. Shi, "ASAP: Aligning Simulation and Real-World Physics for Learning Agile Humanoid Whole-Body Skills," in *Proceedings of Robotics: Science and Systems*, (LosAngeles, CA, USA), June 2025.

[8] A. Allshire, H. Choi, J. Zhang, D. McAllister, A. Zhang, C. M. Kim, T. Darrell, P. Abbeel, J. Malik, and A. Kanazawa, "Visual imitation enables contextual humanoid control," in *Conference on Robot Learning*, pp. 794–815, PMLR, 2025.

[9] Q. Liao, T. E. Truong, X. Huang, Y. Gao, G. Tevet, K. Sreenath, and C. K. Liu, "Beyondmimic: From motion tracking to versatile humanoid control via guided diffusion," *arXiv preprint arXiv:2508.08241*, 2025.

[10] J. P. Sleiman, H. Li, A. Adu-Bredu, R. Deits, A. Kumar, K. Bergamin, M. Bhardwaj, S. Biddlestone, N. Burger, M. A. Estrada, *et al.*, "Zest: Zero-shot embodied skill transfer for athletic robot control," *arXiv preprint arXiv:2602.00401*, 2026.

[11] X. B. Peng, P. Abbeel, S. Levine, and M. Van de Panne, "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Transactions On Graphics (TOG)*, vol. 37, no. 4, pp. 1–14, 2018.

[12] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa, "Amp: Adversarial motion priors for stylized physics-based character control," *ACM Transactions on Graphics (ToG)*, vol. 40, no. 4, pp. 1–20, 2021.

[13] A. Escontrela, X. B. Peng, W. Yu, T. Zhang, A. Iscen, K. Goldberg, and P. Abbeel, "Adversarial motion priors make good substitutes for complex reward functions," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 25–32, IEEE, 2022.

[14] J. Wu, G. Xin, C. Qi, and Y. Xue, "Learning robust and agile legged locomotion using adversarial motion priors," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 4975–4982, 2023.

[15] A. Tang, T. Hiraoka, N. Hiraoka, F. Shi, K. Kawaharazuka, K. Kojima, K. Okada, and M. Inaba, "Humanmimic: Learning natural locomotion and transitions for humanoid robot via wasserstein adversarial imitation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13107–13114, IEEE, 2024.

[16] D. E. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous robots*, vol. 35, no. 2, pp. 161–176, 2013.

[17] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *2014 IEEE-RAS International Conference on Humanoid Robots*, pp. 295–302, IEEE, 2014.

[18] S. Dafarra, G. Romualdi, and D. Pucci, "Dynamic complementarity conditions and whole-body trajectory optimization for humanoid robot locomotion," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3414–3433, 2022.

[19] A. W. Winkler, F. Farshidian, D. Pardo, M. Neunert, and J. Buchli, "Fast trajectory optimization for legged robots using vertex-based zmp constraints," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2201–2208, 2017.

[20] F. Farshidian, E. Jelavić, A. W. Winkler, and J. Buchli, "Robust whole-body motion control of legged robots," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4589–4596, IEEE, 2017.

[21] M. Neunert, F. Farshidian, A. W. Winkler, and J. Buchli, "Trajectory optimization through contacts and automatic gait discovery for quadrupeds," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1502–1509, 2017.

[22] D. Kim, J. Lee, J. Ahn, O. Campbell, H. Hwang, and L. Sentis, "Computationally-robust and efficient prioritized whole-body controller with contact constraints," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–8, IEEE, 2018.

[23] D. Kim, J. Di Carlo, B. Katz, G. Bledt, and S. Kim, "Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control," *arXiv preprint arXiv:1909.06586*, 2019.

[24] G. Mesesan, J. Englsberger, and C. Ott, "Online dcm trajectory adaptation for push and stumble recovery during humanoid locomotion," in *2021 IEEE international conference on robotics and automation (ICRA)*, pp. 12780–12786, IEEE, 2021.

[25] M. Kim, D. Lim, and J. Park, "Online walking pattern generation for humanoid robot with compliant motion control," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 1417–1422, IEEE, 2019.

[26] G. Ficht and S. Behnke, "Fast whole-body motion control of humanoid robots with inertia constraints," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6597–6603, IEEE, 2020.

[27] M. Bogdanovic, M. Khadiv, and L. Righetti, "Model-free reinforcement learning for robust locomotion using demonstrations from trajectory optimization," *Frontiers in Robotics and AI*, vol. 9, p. 854212, 2022.

[28] H. J. Lee, S. Hong, and S. Kim, "Integrating model-based footstep planning with model-free reinforcement learning for dynamic legged locomotion," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11248–11255, IEEE, 2024.

[29] S. Lee, D. Kang, J. Park, and H.-W. Park, "Dynaflow: Dynamics-embedded flow matching for physically consistent motion generation from state-only demonstrations," *arXiv preprint arXiv:2509.19804*, 2025.

[30] J.-G. Kang, J. Park, T.-G. Song, J.-H. Kim, S. Hong, and H.-W. Park, "Agile perceptive multiskill locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 11, no. 116, p. eadz7397, 2026.

[31] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[32] Unitree Robotics, "Unitree G1 Humanoid Robot." `https://www.unitree.com/g1/`.