2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Learning Motion Skills with Adaptive Assistive Curriculum Force in Humanoid Robots

Zhanxiang Cao[1,2], Yang Zhang[1], Buqing Nie[1], Huangxuan Lin[1], Haoyang Li[1,2], Yizhi Chen[3,2], Xiaokang Yang[1], and Yue Gao[1,2†]

***Abstract*— Learning policies for complex humanoid tasks remains both challenging and compelling. Inspired by how infants and athletes rely on external support—such as parental walkers or coach-applied guidance—to acquire skills like walking, dancing, and performing acrobatic flips, we propose A2CF: *Adaptive Assistive Curriculum Force* for humanoid motion learning. A2CF trains a dual-agent system, in which a dedicated *assistive force agent* applies state-dependent forces to guide the robot through difficult initial motions and gradually reduces assistance as the robot's proficiency improves. Across three benchmarks—bipedal walking, choreographed dancing, and backflips—A2CF achieves convergence 30% faster than baseline methods, lowers failure rates by over 40%, and ultimately produces robust, support-free policies. Real-world experiments further demonstrate that adaptively applied assistive forces significantly accelerate the acquisition of complex skills in high-dimensional robotic control.**

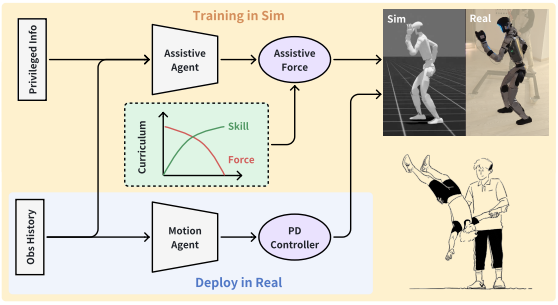

Fig. 1. **Algorithm Framework.** The assistive agent applies forces during simulation to accelerate learning. A curriculum gradually reduces the force as the policy improves. In real-world deployment, the learned policy operates without assistance.

I. INTRODUCTION

The development of humanoid robots capable of learning complex motion skills, such as dancing and acrobatic movements, remains a significant challenge [1], [2]. Despite recent advancements in reinforcement learning (RL) [3]–[5] and imitation learning (IL) [6]–[9], robots still struggle to acquire such skills effectively, particularly in terms of the stability and efficiency of the learning process. A key challenge in this domain is the balance between exploration and exploitation, which often results in slow learning and suboptimal performance [10], [11]. These limitations highlight the need for more effective learning strategies that can improve both the speed and performance of skill acquisition, especially for high-dimensional humanoid control tasks.

During human development, external assistance plays a crucial role in learning motion skills [12]. Infants, for example, often rely on parental support during their first steps, with walkers or direct physical assistance to help them gain the confidence and balance needed for independent locomotion [13], [14]. Similarly, in the case of highly complex movements like backflips, experienced coaches provide physical guidance, supporting the learner's back and applying upward forces to prevent falls and promote proper technique [15]. Studies indicate that such external aids not only expedite the learning process but also help prevent learners from adopting ineffective or unsafe strategies [16]. The assistance prevents learners from getting trapped in local optima, enabling them to discover more efficient and stable movement strategies. This observation serves as a basis for our proposed framework, which incorporates assistive forces during robotic learning.

However, robotic systems face unique challenges in acquiring complex motion strategies. During early training, inefficient exploration can cause robots to miss effective behaviors or converge to suboptimal solutions [1]. Unlike humans, robots lack instinctive physical support , making their learning process more susceptible to instability. Balancing and maintaining robustness further complicate skill acquisition. To address these issues, we draw inspiration from human learning and incorporate assistive forces into the RL framework. These adaptive forces support the robot in the early stages of training, improving stability and accelerating skill acquisition.

In this paper, we propose a RL framework that integrates adaptive assistive forces to facilitate humanoid motion learning. Initially, strong guidance is provided to stabilize and guide the robot, with the assistance gradually reduced as learning progresses. This human-inspired strategy and enables the robot to ultimately perform tasks independently. We validate our method on three tasks—walking, dancing, and backflips—and show that adaptive assistance significantly improves both learning efficiency and final performance in real-world scenarios.

Our primary contributions are as follows:

1) We introduce **A2CF**, a RL framework incorporating **A**daptive **A**ssistive **C**urriculum **F**orce to accelerate the

This work was supported by the National Natural Science Foundation of China (Grant No. 62373242 and No. 92248303).

†Corresponding author, Email: yuegao@sjtu.edu.cn.
[1]MoE Key Lab of Artificial Intelligence and AI Institute, Shanghai Jiao Tong University, Shanghai, China.
[2]Shanghai Innovation Institute, Shanghai, China.
[3]Tongji University, Shanghai, China.

learning of complex humanoid motions. An assistive force agent is jointly trained with the motion agent to provide state-dependent guidance.

2) We enhance the assistive learning paradigm by integrating privileged information, tailored initial state distributions, and random masking, inspired by human motor learning, to improve generalization and prevent over-reliance on external support.
3) We conduct both simulation and real-world experiments, demonstrating that A2CF yields notable improvements in learning speed and task performance, and successfully transfers to physical humanoid robots.

II. RELATED WORKS

A. Learning-Based Locomotion in Humanoid

Recent work has advanced humanoid locomotion through learning-based methods. Radosavovic et al. [3] proposed a transformer-based RL policy for zero-shot deployment across terrains. HiLo [17] combined motion tracking with domain randomization for robust movement. Gu et al. [18] introduced the DWL framework for mastering complex terrains via sim-to-real transfer.

Furthermore, HoST [19] and HumanUp [20] focus on training humanoid robots to recover from postures such as lying down and crawling to standing. In HoST, an externally applied vertical force accelerates the learning of the recovery motion, but this force is not adaptive and is independent of the robot's state.

These methods often rely on reward shaping, domain randomization, or heuristic interventions. In contrast, A2CF provides a learnable, state-aware assistive force that accelerates training without modifying task objectives or assuming specific reward structures.

B. Imitation Learning for Humanoid Whole-Body Control

Imitation learning has become a powerful method for teaching humanoid robots complex whole-body movements by mimicking human demonstrations. Zhang et al. [21] applied AMP [22] to humanoid robots, introducing human walking trajectory priors to make the robot's gait more human-like. Similarly, OmniH2O [7] and ExBody2 [8] achieve dexterous whole-body imitation by retargeting human motion trajectories to specific robots. ASAP [6] also follows this paradigm, aligning sim and real physics to enable robust imitation-based control.

The TRILL system [23] utilizes deep imitation learning for loco-manipulation tasks, while Matsuura et al. [24] developed a whole-body imitation learning system for a biped and bi-armed humanoid robot. AMO [25] further explores adaptive motion decomposition for hyper-dexterous loco-manipulation skills.

While prior studies achieve impressive performance, they often depend on curated motion priors, tracking supervision, or phase annotations. In contrast, A2CF requires no additional expert demonstrations or handcrafted rewards beyond the original task specifications. Instead, it accelerates policy convergence and enhances performance by providing adaptive physical guidance during training, complementing existing approaches under sparse or unstable reward conditions.

III. METHODOLOGY

A. Problem Statement

In this work, we model the task of learning motion skills as a Partially Observable Markov Decision Process (POMDP), denoted by the tuple $M = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \Omega, \mathcal{O}, \gamma)$. Specifically, $\mathcal{S}$ denotes the state space, which consists of the set of states s_t at each time step t. The state includes the observation o_t, and privileged information o_t^{priv} available during simulation. The observation contains robot-accessible sensory data:

$$o_t = [w_t, g_t, q_t, \dot{q}_t, a_{t-1}, c_t],$$

where w_t represents the angular velocity, g_t is the gravity projection vector of the pose, q_t and $\dot{q}_t$ denote the joint positions and velocities, respectively, a_{t-1} is the previous action, and c_t is the current command.

The action space $\mathcal{A}$ is the set of actions a_t that the agent can execute. The environment evolves according to the state transition function $\mathcal{T}(s_{t+1}|s_t, a_t)$, which defines the probability distribution of the next state s_{t+1} given the current state s_t and action a_t. The agent receives a reward $r_t \sim \mathcal{R}(s_t, a_t)$ and observes $o_t \sim \mathcal{O}(o_t|s_{t+1}, a_t)$.

The goal of the agent is to learn an optimal policy π^* that maximizes the expected cumulative reward $J(\pi)$ over an infinite time horizon. This is formalized as:

$$J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) \right],$$

where $\gamma \in [0, 1)$ is the discount factor that determines the weight given to future rewards.

B. Learning with Adaptive Assistive Spatial Force (*A2CF*)

1) Expanded Action Space: In addition to the motion policy agent, an assistive force agent is trained to assist the robot. The close collaboration between the two agents is facilitated by joint action learners (JAL) [26], [27]. The action a_t^{motion} of the motion policy agent corresponds to the desired joint position q_t^{des}, which is subsequently controlled by a PD controller. Based on this, the action space is expanded with the assistive force agent's action a_t^{assi}, which consists of a 6-D spatial force $F_t = [f_t, m_t]$, where f_t represents the linear force and m_t represents the torque. For convenience, this force is applied to the robot's base link, specifically the pelvis of the humanoid robot. Thus, the expanded action space becomes $a_t = [a_t^{\text{motion}}, a_t^{\text{assi}}]$, enabling simultaneous adjustment of joint positions and assistive forces during training.

2) Assistive Force Curriculum: In human motor learning—such as walking, backflipping, or dancing—external assistance from parents or coaches is typically provided during the early learning stages and gradually removed as the learner gains proficiency. Inspired by this observation, we introduce a curriculum-based mechanism for regulating

assistive forces in robotic motion learning. Specifically, we define a bounded action space for the assistive force agent as a 6D hypercube centered at the origin:

$$\mathcal{B}_k = \left\{ F \in \mathbb{R}^6 \mid -\eta_{k,i} \leq F_i \leq \eta_{k,i}, \quad \forall i \in \{1, \ldots, 6\} \right\},$$

where $\eta_k \in \mathbb{R}^+$ denotes the half-width of the hypercube at training iteration k, and each dimension corresponds to a linear or torque component of the spatial assistive force.

To ensure a gradual reduction in assistance, the bound η_k is adaptively updated based on the magnitude of the applied force F_k and a skill acquisition indicator. The update rule is described in Algorithm 1. When the normalized magnitude $\|F_k\|/\|\eta_k\|$ falls below a threshold, the system infers that the assistive contribution is diminishing and decreases η_k accordingly. If the motion policy has completed skill acquisition, a decay is also enforced to eliminate external support during deployment.

Algorithm 1 Hypercube-Based Assistive Force Curriculum

Input: Initial bound η_0, applied force F_k, skill acquisition flag `isSkillLearned`
Output: Updated bound η_k
for each training iteration $k = 1, 2, \ldots$ **do**
  if $\|F_k\| < (1 - \epsilon) \cdot \|\eta_k\|$ **then**
    $\eta_k \leftarrow (1 - \delta) \cdot \eta_k$
  else if $\|F_k\| > (1 + \epsilon) \cdot \|\eta_k\|$ **then**
    $\eta_k \leftarrow (1 + \delta) \cdot \eta_k$
  end if
  if `isSkillLearned` **then**
    $\eta_k \leftarrow (1 - \delta) \cdot \eta_k$
  end if
end for

*3) Initial Distribution of Spatial Force (**ID**):* An appropriately designed initial bound of the assistive force hypercube can provide the assistive force agent with a strong prior, thereby improving learning efficiency and stability. For example, in the training of walking skills, assistive forces are predominantly required in the lateral xy-plane, while the demand along the vertical z-axis is minimal. In contrast, more dynamic tasks such as backflips exhibit phase-dependent variations in assistive force requirements: during the *Stand* and *Land* phases, lateral assistance in the xy-plane is more critical, whereas the *Jump* and *Air* phases necessitate stronger support in the z-direction to generate and control vertical momentum.

To encode such human-inspired priors into the learning process, we initialize the spatial force bound η_0 in a task-dependent manner. These initial bounds guide the early behavior of the assistive force agent by shaping the feasible region of the applied forces. The specific initialization bounds for each task are described in detail in Section IV-A.

*4) Assistive Force Agent with Privileged Information (**PI**):* Since the assistive force agent only operates in simulation, it can fully utilize privileged information available in the simulation, as demonstrated by Lee et al. [28]. During the training of walking skills, the assistive force agent can leverage terrain information to provide assistive forces that align with the current terrain conditions. This enables the motion policy agent to overcome difficulties more efficiently, helping it escape potentially conservative local optima and more quickly navigate challenging terrains.

In addition to terrain information, privileged information such as the current assistive force bound, robot linear velocity, and domain randomization parameters (e.g., ground friction coefficient, disturbance speed, and mass) are also used. This mirrors how a parent assists a child by observing details the child cannot perceive, providing more informed support.

*5) Random Mask of Spatial Force (**RM**):* When assistive forces are large and persistent, the motion policy agent may become overly dependent on them. To mitigate this, we introduce a random masking mechanism that intermittently disables assistive forces, encouraging the agent to learn independent control.

Specifically, during each training iteration, a random mask is applied to selectively disable certain components of the assistive force with probability ζ. The final assistive force F_t^{assi} applied to the robot is expressed as:

$$F_t^{\text{assi}} = M_t \odot F_t$$

where $M_t \in \{0, 1\}$ is a randomly generated mask, with $P(M_t = 0) = \zeta$ and $P(M_t = 1) = 1 - \zeta$, and F_t represents the output of the assistive force agent. By employing this method, the robot is occasionally required to complete tasks without the aid of assistive forces, similar to how a parent occasionally lets go when teaching a child to walk, allowing the child to develop the ability to walk independently.

This reduces reliance on assistive forces and accelerates the transition to autonomous execution.

C. Overall Training Architecture

For the POMDP problem, an Asymmetric Actor-Critic (AAC) structure [29] is employed for training. The motion policy agent receives only the observation o_t and historical information o_t^H, while the Critic receives the full state s_t, including privileged information o_t^{priv}, available only in simulation. To better leverage o_t^H, the architecture from [30], [31] is adopted, using a Variational Autoencoder (VAE) [32], [33] to compress historical data and estimate the robot's velocity v_t. It predicts the next observation o_{t+1} by minimizing MSE loss and KL divergence to a standard Gaussian $\mathcal{N}(0, I)$.

Both the motion policy agent and the assistive force agent share the compressed latent representation z_t, along with the estimated velocity v_t and observation o_t as inputs. The assistive agent also receives selected privileged information (see Section III-B.4). The PPO algorithm [34] is used to optimize the policy.

D. Task-Specific Training Frameworks

This section outlines the training setup for three representative locomotion tasks, including reward design and environment configuration, all based on original task-specific

RL formulations without requiring additional engineering effort specific to our method. The overall reward function consists of two components: the motion reward r^{motion} and the assistive force reward r^{force}. The former encourages the acquisition of desired motion skills and includes both task-relevant terms and regularization terms, while the latter penalizes the excessive use of assistive forces to ensure minimal reliance on external support. Notably, A2CF does *not* rely on manually designed task-specific state machines. Any task structuring (e.g., phase sequencing) is inherent to the task definition and remains independent of our framework. The assistive force reward r^{force} is only introduced once the motion agent achieves 80% of the skill acquisition target.

1) Walking: In the walking task, a locomotion policy is trained to follow a command velocity vector $\mathbf{c}_t = [v_x^{\text{cmd}}, v_y^{\text{cmd}}, \omega_z^{\text{cmd}}]$, representing the desired forward and lateral linear velocities and the angular velocity of the yaw, respectively. The task reward is primarily based on exponential tracking rewards for velocity errors. Additionally, a cost of transport term, adapted from [35], is introduced to optimize the energy efficiency of the gait. The detailed reward components are summarized in Table I.

TABLE I
REWARD TERMS FOR WALKING TASK.

Type	Name	Equation	Weight
Task	Lin. Vel. Track	$\exp[-4(v_{xy} - v_{xy}^{\text{cmd}})^2]$	5.0
	Ang. Vel. Track	$\exp[-4(\omega_z - \omega_z^{\text{cmd}})^2]$	2.0
Reg.	Feet Air Time	$\sum_i \mathbb{1}_{\text{air},i} \cdot t_{\text{air},i}$	1.0
	Feet Slide	$\sum_i \mathbb{1}_{\text{contact},i} \cdot \|v_{foot,i}\|$	-1.0
	Cost of Transport	$\frac{\|\dot{q}\|\|\tau\|}{9.81\|\mathbf{v}_{xy}\|}$	-0.01
	Stand Still	$\mathbb{1}_{\|\mathbf{c}\|<0.1} \cdot \|q - q_{\text{default}}\|$	-0.1
	Action Rate	$\|a_t - a_{t-1}\|^2$	-0.01
	DOF Acc.	$\|\ddot{q}\|^2$	-2.5e-7
	DOF Pos. Limit	$\sum_j (\|q_j\| - q_j^{\text{lim}})$	-1.0
	Orientation	$\|g\|$	-1.0
Force	Less Assi. Force	$\exp[-2\|F\|/\|\eta\|]$	2.0

2) Backflip: For complex motion tasks such as backflip, a phase-based training approach is adopted following the method in [36]. The environment maintains a finite state machine (FSM) with five discrete phases: *Stand*, *Crouch*, *Jump*, *Air*, and *Land*, as illustrated in Fig. 2. Transitions between phases are primarily determined by the robot's height and foot contact states.

The task command $\mathbf{c}_t$ is defined as a simple start trigger $\mathbf{c}_t = \mathbb{1}_{\text{start}}$, which also acts as the condition for transitioning from the *Stand* phase to *Crouch*. Based on the current phase, the reward function is redesigned to guide learning at each phase. In particular, additional vertical velocity rewards are introduced during the *Crouch* and *Jump* phases to encourage rapid body movement. The complete reward design is summarized in Table II.

3) Dancing: To learn expressive whole-body motions, we utilize the dancing subset from the LAFAN1 motion dataset [37], retargeted to the Unitree G1 robot by Unitree Robotics Company. A total of 8 motion clips are selected as

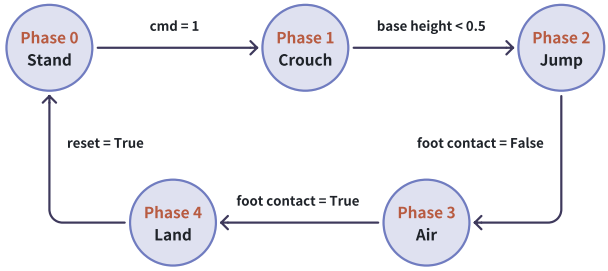

Fig. 2. **FSM for Backflip.** The figure shows the five states of the backflip task, with arrows indicating state transitions. The conditions for these transitions are shown along the horizontal lines.

TABLE II
REWARD TERMS FOR BACKFLIP TASK.

Type	Name	Equation	Weight
Task	Alive (Phase 0)	$\mathbb{1}_{\text{phase}_0}$	2.0
	Down. Vel. (Phase 1)	$\mathbb{1}_{\text{phase}_1} \cdot \mathbb{1}_{\text{contact}} \cdot -v_z$	2.0
	Up. Vel. (Phase 2)	$\mathbb{1}_{\text{phase}_2} \cdot v_z$	2.0
	Ang. Vel. (Phase 2)	$\mathbb{1}_{\text{phase}_2} \cdot -w_y$	0.5
	Ang. Vel. (Phase 3)	$\mathbb{1}_{\text{phase}_3} \cdot -w_y$	2.0
	Land (Phase 4)	$\mathbb{1}_{\text{phase}_4} \cdot exp[-10\|v\|]$	20.0
Reg.	Balance(Phase 0,1,4)	$\mathbb{1}_{\text{phase}_{0,1,4}} \cdot \angle \mathbf{z}_{\text{base}}, \mathbf{z}_{\text{world}}$	-2.0
	Balance(Phase 2,3)	$\mathbb{1}_{\text{phase}_{2,3}} \cdot \angle \mathbf{y}_{\text{base}}, \mathbf{y}_{\text{world}}$	-2.0
	Yaw Ang. Vel.	$\|w_z\|$	-0.1
	Action Rate	$\|a_t - a_{t-1}\|^2$	-0.01
	Joint Acc.	$\|\ddot{q}\|^2$	-2.5e-7
	DOF Pos. Limit	$\sum_j (\|q_j\| - q_j^{\text{lim}})$	-1.0
	DOF Pos. Deviation	$\sum_i \mathbb{1}_{\text{phase}_i} \cdot \|q - q_{\text{phase}_i}^{\text{des}}\|$	-0.2
Force	Less Assi. Force	$\exp[-2\|F\|/\|\eta\|]$	1.0

expert demonstrations. These trajectories are interpolated to match the policy control frequency of 50 Hz.

The task command c_t is defined as $[\Delta\mathbf{p}^{\text{cmd}}, \Delta\mathbf{q}^{\text{cmd}}, q^{\text{cmd}}]$, where $\Delta\mathbf{p}^{\text{cmd}}$ denotes the desired displacement of the robot's base position, $\Delta\mathbf{q}^{\text{cmd}}$ denotes the quaternion-based rotational difference of the base between adjacent timesteps, and q^{cmd} represents the desired joint positions. The policy is trained to follow these references using exponential tracking rewards based on the respective errors. The complete reward terms are listed in Table III.

TABLE III
REWARD TERMS FOR DANCING TASK.

Type	Name	Equation	Weight
Task	Base Pos. Track	$\exp[-2500(\Delta\mathbf{p} - \Delta\mathbf{p}^{\text{cmd}})^2]$	10.0
	Base Quat. Track	$\exp[-100(\Delta\mathbf{q} - \Delta\mathbf{q}^{\text{cmd}})^2]$	5.0
	DOF Pos. Track	$\exp[-0.25(q - q^{\text{cmd}})^2]$	20.0
Reg.	Feet Slide	$\sum_i \mathbb{1}_{\text{contact},i} \cdot \|v_{foot,i}\|$	-1.0
	Action Rate	$\|a_t - a_{t-1}\|^2$	-0.01
	DOF Acc.	$\|\ddot{q}\|^2$	-2.5e-7
	DOF Torque	$\|\tau\|^2$	-1e-5
Force	Less Assi. Force	$\exp[-2\|F\|/\|\eta\|]$	2.0

IV. EXPERIMENTAL RESULTS

To validate the effectiveness and impact of the A2CF method, it is applied to three tasks: walking, backflip, and dancing. For both the walking and dancing tasks, real-world transfer experiments are conducted to verify the performance of the trained policies.

A. Experimental Setup

1) Simulation Training: Training is conducted on the Unitree G1 EDU humanoid robot, which consists of a total of 29 degrees of freedom (DoF): 7 DoF for each arm, 3 DoF for the waist, and 6 DoF for each leg. Isaac Lab [38] is employed as the simulation platform, and the Proximal Policy Optimization (PPO) algorithm [34], [39] is used for policy optimization. The training is performed in parallel across 4096 environments on an NVIDIA GeForce 4090 GPU. The simulation time step is set to $\Delta t = 0.001$ s, and the environment step is set to $\Delta t_{\text{env}} = 0.02$ s, resulting in an execution frequency of 50 Hz for both the motion policy agent and the assistive force agent. For the walking and backflip tasks, each iteration takes approximately 5 seconds, while the dancing task, due to the need to sample expert data, requires approximately 8 seconds per iteration.

In the assistive force curriculum learning (Alg. 1), the hyperparameters are set to $\epsilon = 0.5$ and $\delta = 0.2$. The initial distribution of the assistive force constraints in Section III-B.3 is shown in Table IV. In Section III-B.5, the probability of random dropout for the mask is set to $\zeta = 0.2$.

TABLE IV
INITIAL DISTRIBUTION OF SPATIAL FORCE

Task		f_t Initial Bound	m_t Initial Bound
Walking		[40, 40, 10]	[40, 40, 40]
Backflip	Phase 0	[40, 40, 10]	[30, 30, 10]
	Phase 1	[40, 40, 10]	[30, 30, 10]
	Phase 2	[40, 40, 100]	[30, 30, 30]
	Phase 3	[40, 40, 100]	[30, 30, 30]
	Phase 4	[40, 40, 10]	[30, 30, 10]
Dancing		[40, 40, 40]	[40, 40, 40]

2) Real-World Deployment: The learned policy is directly transferred to the real robot without fine-tuning for the walking and dancing tasks. The robot operates solely on the basis of proprioceptive sensors, which include joint angles, velocities, body orientation, and angular velocity, without the use of external sensor inputs. The control policy is executed at a frequency of 50 Hz on a personal computer equipped with an Intel Core i9-12900H CPU.

B. Compared Methods

To demonstrate the effectiveness of the proposed method in accelerating learning and achieving higher performance, the training curves of the Baseline and A2CF are compared. Additionally, a series of ablation experiments are conducted to validate the effectiveness of the PI, ID, and RM components, which are detailed in Section III-B.4, Section III-B.3, and Section III-B.5, respectively. The methods compared are as follows:

1) **Baseline**: The DreamWaQ framework implemented on the humanoid robot.
2) **A2CF**: Our proposed method.
3) **A2CF w/o PI**: A2CF without the use of privileged information (PI) for the assistive force agent.
4) **A2CF w/o ID**: A2CF without the initial distribution of spatial force (ID). The spatial force is initialized with identical bounds across the x, y, and z dimensions.
5) **A2CF w/o RM**: A2CF without the random mask (RM) mechanism for the assistive force.

To ensure fairness, the reward function used in all methods, except for the assistive force-related terms, is kept consistent between A2CF and Baseline. Furthermore, all methods are trained with the same curriculum learning, domain randomization parameters, seed, and network architecture.

C. Walking Task

The walking policy is trained on rough terrain, which includes flat surfaces, slopes, rough terrains, discretized terrains, and stairs, organized into a 10-level difficulty curriculum. The highest stair height is 23 cm. Since the terrain level is randomly reassigned to any difficulty once a robot reaches the maximum level, the time-averaged terrain level across all environments is approximately 6. The speed commands include a maximum forward velocity of 1.2 m/s, a maximum lateral velocity of 0.8 m/s, and a maximum angular velocity of 1.5 rad/s. Additionally, domain randomization is applied for friction coefficients, load, random velocity disturbances, and the position of the center of mass. The robot's proficiency in the task is determined by whether it reaches the maximum terrain level, which then triggers the assistive force limit curriculum.

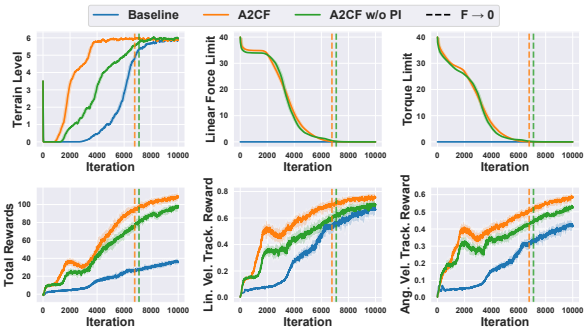

Fig. 4. **Training Curves for the Walking Task.** The figure shows the terrain level, force limit curriculum, total rewards, and raw velocity tracking reward (ranging from 0 to 1). Vertical dashed lines indicate the point at which the assistive force becomes negligible.

In this task, the A2CF algorithm is compared to the Baseline and A2CF without Privileged Information (w/o PI) during training. The results, shown in Figure 4, demonstrate that A2CF learns walking capabilities much faster than the Baseline. Specifically, A2CF reaches the maximum terrain

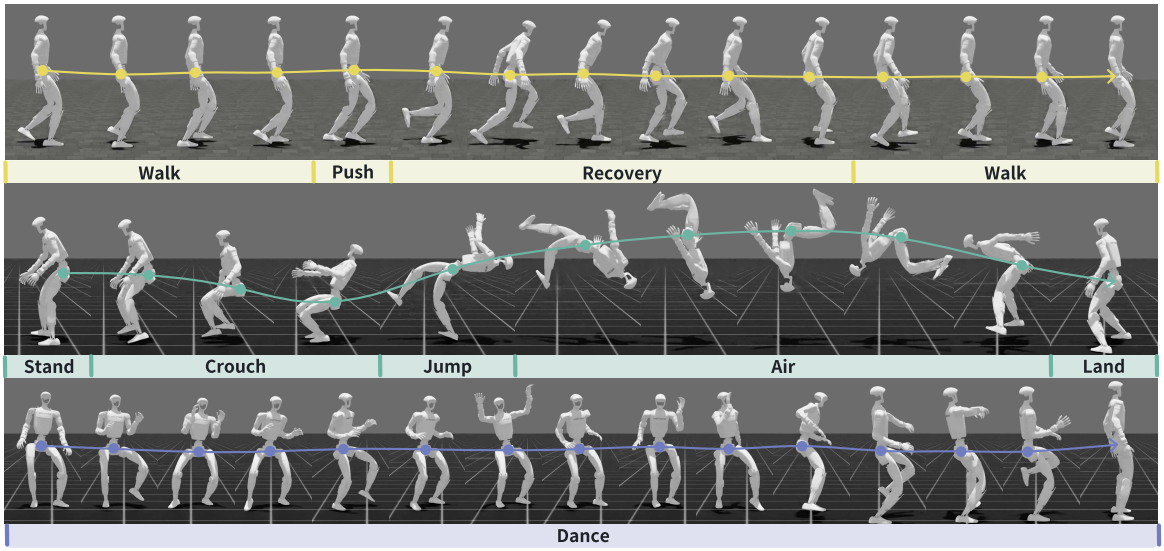

Fig. 3. **Simulation of Walking, backflip, and Dancing Tasks.** The figure illustrates the execution of three tasks—walking, backflip, and dancing—in the simulation. The walking and backflip tasks have a 0.1 s interval between frames, while the dancing task has a 1 s interval. The walking sequence shows the robot recovering from a push disturbance. The backflip sequence highlights the robot's movement through five phases of the backflip.

level around 4k iterations, while the Baseline requires approximately 10k iterations. Furthermore, the magnitude of the assistive force in A2CF falls below a negligible threshold of 0.1 N—corresponding to an insignificant value for a 35 kg robot—around 7k iterations, effectively enabling the emergence of an autonomous walking policy. Additionally, A2CF outperforms the Baseline in tracking both forward speed and angular velocity errors.

The ablation study on A2CF w/o PI highlights the critical role of privileged information, such as terrain perception, in enhancing the assistive force agent. The inclusion of PI accelerates learning, causes the assistive force to decay to zero more rapidly, and improves velocity tracking performance.

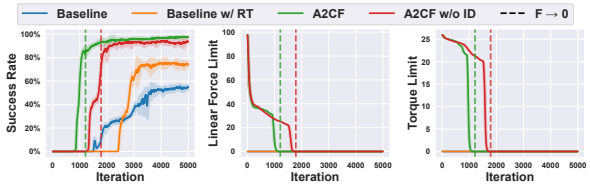

Fig. 5. **Training Curves for the Backflip Task.** The figure shows the success rate of the backflip task during training and the corresponding assistive force limit curriculum. Vertical dashed lines indicate the point at which the assistive force becomes negligible.

D. Backflip Task

The backflip policy is trained on flat terrain with randomized friction coefficients and payload masses. Task success is determined by whether the robot reaches the land phase and remains stable, which also triggers the assistive force limit curriculum.

In this task, the training success rates of Baseline, A2CF, and the ablated variant A2CF w/o ID are compared. Except for the additional assistive force reward, all other reward terms and weights remain consistent between Baseline and A2CF. Considering the reward sensitivity of the backflip task, an additional version of the Baseline is trained with manually tuned reward parameters, referred to as *Baseline with Reward Tuning (Baseline w/ RT)*. The comparative results are shown in Figure 5.

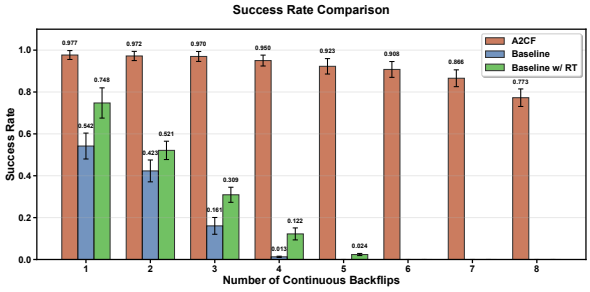

Fig. 6. **Success rate comparison across varying numbers of continuous backflips.** A2CF (brown) shows robust performance across 1–8 flips, while baseline methods degrade rapidly. Error bars indicate standard deviation over 5 seeds.

To further evaluate A2CF, we introduce a harder variant: the *continuous backflip task*, where the robot performs multiple backflips in succession. All methods use the same reward structure for fair comparison. As shown in Fig. 6, A2CF maintains high success rates even for 8 flips, while baseline variants fail beyond 4. These results show that A2CF not only accelerates learning but also scales to tasks that standard methods cannot solve.

Experimental results demonstrate that A2CF, benefiting from adaptive assistive force, learns the backflip policy more efficiently and achieves a success rate exceeding 90%. The ablation study on A2CF w/o ID further validates the

importance of a well-designed initial distribution of assistive force. In the backflip task, assistive force requirements vary significantly across different motion phases. Using uniform initial limits across all phases may reduce the diversity and effectiveness of the assistive force, ultimately hindering learning performance.

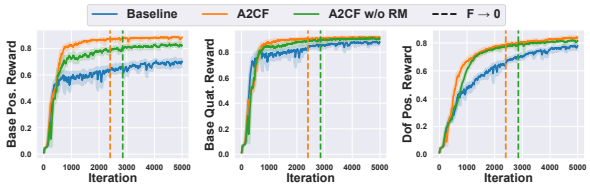

Fig. 7. **Training Curves for the Dancing Task.** The figure shows the raw tracking rewards for base position, base orientation, and joint positions, without applying reward weights. Vertical dashed lines indicate the point at which the assistive force becomes negligible.

E. Dancing Task

In the dancing task, training is conducted on flat terrain with randomized friction coefficients, payload masses, and center-of-mass positions. Task success is determined by whether the joint position tracking error reaches a threshold of 0.16, which triggers the assistive force limit curriculum.

In this task, the training reward curves for base position, base orientation, and joint position tracking are compared across Baseline, A2CF, and A2CF w/o RM. The experimental results, shown in Figure 7, indicate that A2CF achieves faster learning and better performance in joint position tracking accuracy compared to Baseline. While there is no significant improvement in the speed of base position and orientation tracking, A2CF demonstrates a substantial increase in tracking accuracy.

The ablation study on A2CF w/o RM highlights the importance of the random masking mechanism. By intermittently masking the assistive force, the method accelerates the agent's transition from reliance on assistive force to independent task execution, with assistive force decaying to zero earlier, enabling the agent to complete the task autonomously at an earlier stage.

F. Sim2Sim2Real Results

To evaluate the effectiveness of the trained policies in real-world conditions, a Sim2Sim2Real transfer study was conducted. For the walking task, the policy was evaluated across diverse real-world terrains, including flat surfaces, grass, and uneven cobblestone paths, as demonstrated in the accompanying video. The results confirm that the walking policy generalizes well to real-world environments, as illustrated in Figure 8.

For the backflip task, real-world testing was not conducted due to limitations in funding and available space. However, the policy was transferred and validated in the Genesis simulator [40], where similar results were achieved. This validation process further supports the applicability of the trained policies in real-world scenarios. The Sim2Sim results for the backflip task are shown in Figure 8.

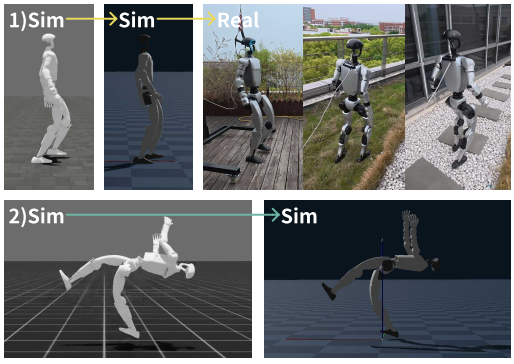

Fig. 8. **Sim2Sim2Real Transfer for Walking and Sim2Sim for Backflip.** 1) Walking: transfer from IsaacLab (training) to Genesis (validation) to the real world. 2) Backflip: transfer from IsaacLab to Genesis.

Fig. 9. **Snapshots of the learned dancing motion using A2CF on the real robot.** Each subfigure includes the corresponding target motion from the LAFAN1 dataset (bottom right) for side-by-side comparison, highlighting the high-fidelity reproduction of complex body dynamics.

Sim2Sim2Real transfer was also performed for the dancing task. The real-world results are presented in Figure 9, where the robot performs the dance routine with high accuracy, demonstrating its ability to execute predefined motion sequences in a real-world environment.

These results demonstrate A2CF's ability to transfer learned policies across different domains, which is crucial for practical deployment.

V. CONCLUSIONS AND DISCUSSION

In this work, we proposed the A2CF method to enhance learning efficiency and task performance across robotic motion tasks, including walking, backflips, and dancing. By incorporating adaptive assistive forces, privileged information, initial force distribution, and random masking, A2CF accelerates the learning process and facilitates more autonomous task execution. Experimental results demonstrate that A2CF outperforms the Baseline in terms of learning speed and task success rates. Specifically, in the walking task, A2CF achieves faster learning and transitions to autonomy more efficiently. In the backflip task, A2CF achieves a success

rate exceeding 90%, while in the dancing task, it improves joint position tracking accuracy.

In this work, assistive forces are intentionally applied only to the base link (pelvis), as stabilizing the base is essential for learning dynamic whole-body locomotion. While this design simplifies control, it may limit extension to loco-manipulation tasks requiring upper-limb coordination. The framework, however, can be naturally extended to apply forces to other links, such as the arms or hands, by expanding the actuation space. Moreover, interaction with external objects, such as walls, could allow the robot to autonomously obtain assistive support, enhancing adaptability in real-world environments.

REFERENCES

[1] Y. Tong, H. Liu, and Z. Zhang, "Advancements in humanoid robots: A comprehensive review and future prospects," *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 2, pp. 301–328, 2024.

[2] Z. Gu, J. Li, W. Shen, W. Yu, Z. Xie, S. McCrory, X. Cheng, A. Shamsah, R. Griffin, C. K. Liu *et al.*, "Humanoid locomotion and manipulation: Current progress and challenges in control, planning, and learning," *arXiv preprint arXiv:2501.02116*, 2025.

[3] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath, "Real-world humanoid locomotion with reinforcement learning," *Science Robotics*, vol. 9, no. 89, p. eadi9579, 2024.

[4] Y. Xue, W. Dong, M. Liu, W. Zhang, and J. Pang, "A unified and general humanoid whole-body controller for fine-grained locomotion," *arXiv preprint arXiv:2502.03206*, 2025.

[5] H. Wang, Z. Wang, J. Ren, Q. Ben, T. Huang, W. Zhang, and J. Pang, "Beamdojo: Learning agile humanoid locomotion on sparse footholds," *arXiv preprint arXiv:2502.10363*, 2025.

[6] T. He, J. Gao, W. Xiao, Y. Zhang, Z. Wang, J. Wang, Z. Luo, G. He, N. Sobanbab, C. Pan *et al.*, "Asap: Aligning simulation and real-world physics for learning agile humanoid whole-body skills," *arXiv preprint arXiv:2502.01143*, 2025.

[7] T. He, Z. Luo, X. He, W. Xiao, C. Zhang, W. Zhang, K. Kitani, C. Liu, and G. Shi, "Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning," *arXiv preprint arXiv:2406.08858*, 2024.

[8] M. Ji, X. Peng, F. Liu, J. Li, G. Yang, X. Cheng, and X. Wang, "Exbody2: Advanced expressive humanoid whole-body control," *arXiv preprint arXiv:2412.13196*, 2024.

[9] F. Liu, Z. Gu, Y. Cai, Z. Zhou, S. Zhao, H. Jung, S. Ha, Y. Chen, D. Xu, and Y. Zhao, "Opt2skill: Imitating dynamically-feasible whole-body trajectories for versatile humanoid loco-manipulation," *arXiv preprint arXiv:2409.20514*, 2024.

[10] H. L. Kwa, J. Leong Kit, and R. Bouffanais, "Balancing collective exploration and exploitation in multi-agent and multi-robot systems: A review," *Frontiers in Robotics and AI*, vol. 8, p. 771520, 2022.

[11] P. Ladosz, L. Weng, M. Kim, and H. Oh, "Exploration in deep reinforcement learning: A survey," *Information Fusion*, vol. 85, pp. 1–22, 2022.

[12] G. Wulf, *Attention and motor skill learning*. Human Kinetics, 2007.

[13] L. J. Claxton, D. K. Melzer, J. H. Ryu, and J. M. Haddad, "The control of posture in newly standing infants is task dependent," *Journal of experimental child psychology*, vol. 113, no. 1, pp. 159–165, 2012.

[14] C. Von Hofsten, "Eye–hand coordination in the newborn." *Developmental psychology*, vol. 18, no. 3, p. 450, 1982.

[15] P. H. Werner, L. H. Williams, and T. J. Hall, *Teaching children gymnastics*. Human Kinetics, 2012.

[16] T. Dowdell, "Characteristics of effective gymnastics coaching," *Science of gymnastics Journal*, vol. 2, no. 1, pp. 15–24, 2010.

[17] Q. Zhang, C. Weng, G. Li, F. He, and Y. Cai, "Hilo: Learning whole-body human-like locomotion with motion tracking controller," *arXiv preprint arXiv:2502.03122*, 2025.

[18] X. Gu, Y.-J. Wang, X. Zhu, C. Shi, Y. Guo, Y. Liu, and J. Chen, "Advancing humanoid locomotion: Mastering challenging terrains with denoising world model learning," *arXiv preprint arXiv:2408.14472*, 2024.

[19] T. Huang, J. Ren, H. Wang, Z. Wang, Q. Ben, M. Wen, X. Chen, J. Li, and J. Pang, "Learning humanoid standing-up control across diverse postures," *arXiv preprint arXiv:2502.08378*, 2025.

[20] X. He, R. Dong, Z. Chen, and S. Gupta, "Learning getting-up policies for real-world humanoid robots," *arXiv preprint arXiv:2502.12152*, 2025.

[21] Q. Zhang, P. Cui, D. Yan, J. Sun, Y. Duan, G. Han, W. Zhao, W. Zhang, Y. Guo, A. Zhang *et al.*, "Whole-body humanoid robot locomotion with human reference," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 11 225–11 231.

[22] 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.

[23] M. Seo, S. Han, K. Sim, S. H. Bang *et al.*, "Deep imitation learning for humanoid loco-manipulation through human teleoperation," *arXiv preprint arXiv:2309.01952*, 2023.

[24] Y. Matsuura, K. Kawaharazuka, N. Hiraoka, K. Kojima, K. Okada, and M. Inaba, "Development of a whole-body work imitation learning system by a biped and bi-armed humanoid," *arXiv preprint arXiv:2309.15756*, 2023.

[25] J. Li, X. Cheng, T. Huang, S. Yang, R.-Z. Qiu, and X. Wang, "Amo: Adaptive motion optimization for hyper-dexterous humanoid whole-body control," *arXiv preprint arXiv:2505.03738*, 2025.

[26] L. Canese, G. C. Cardarilli, L. Di Nunzio, R. Fazzolari, D. Giardino, M. Re, and S. Spanò, "Multi-agent reinforcement learning: A review of challenges and applications," *Applied Sciences*, vol. 11, no. 11, p. 4948, 2021.

[27] Y. Du, J. Z. Leibo, U. Islam, R. Willis, and P. Sunehag, "A review of cooperation in multi-agent learning," *arXiv preprint arXiv:2312.05162*, 2023.

[28] 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.

[29] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel, "Asymmetric actor critic for image-based robot learning," *arXiv preprint arXiv:1710.06542*, 2017.

[30] I. M. A. Nahrendra, B. Yu, and H. Myung, "Dreamwaq: Learning robust quadrupedal locomotion with implicit terrain imagination via deep reinforcement learning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5078–5084.

[31] Y. Zhang, B. Nie, and Y. Gao, "Robust locomotion policy with adaptive lipschitz constraint for legged robots," *IEEE Robotics and Automation Letters*, 2024.

[32] I. Higgins, L. Matthey, A. Pal, C. P. Burgess, X. Glorot, M. M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework." *ICLR (Poster)*, vol. 3, 2017.

[33] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, "Understanding disentangling in β-vae," *arXiv preprint arXiv:1804.03599*, 2018.

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

[35] Z. Fu, A. Kumar, J. Malik, and D. Pathak, "Minimizing energy consumption leads to the emergence of gaits in legged robots," *arXiv preprint arXiv:2111.01674*, 2021.

[36] D. Kim, H. Kwon, J. Kim, G. Lee, and S. Oh, "Stage-wise reward shaping for acrobatic robots: A constrained multi-objective reinforcement learning approach," *arXiv preprint arXiv:2409.15755*, 2024.

[37] F. G. Harvey, M. Yurick, D. Nowrouzezahrai, and C. Pal, "Robust motion in-betweening," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 60–1, 2020.

[38] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar *et al.*, "Orbit: A unified simulation framework for interactive robot learning environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3740–3747, 2023.

[39] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Conference on Robot Learning*. PMLR, 2022, pp. 91–100.

[40] G. Authors, "Genesis: A universal and generative physics engine for robotics and beyond," December 2024. [Online]. Available: https://github.com/Genesis-Embodied-AI/Genesis