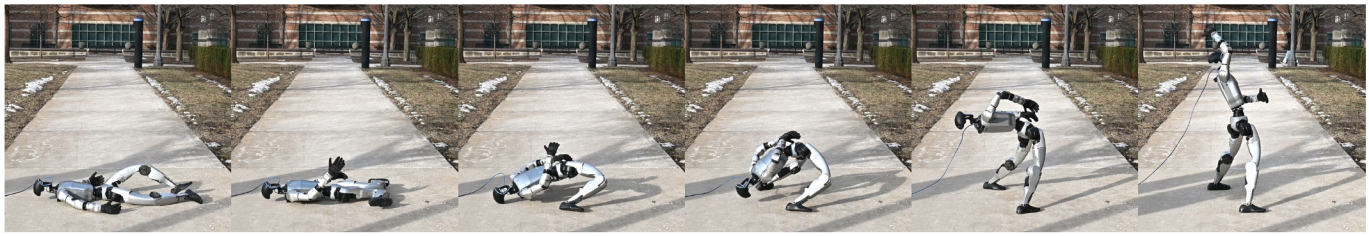
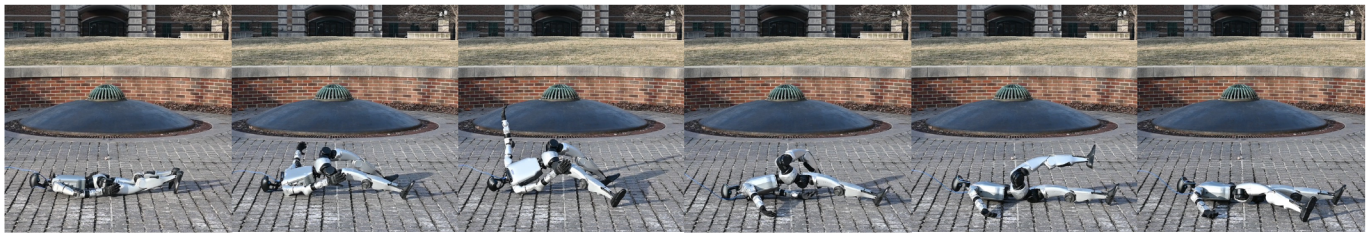
Robotics: Science and Systems 2025
Los Angeles, CA, USA, June 21–25, 2025

Learning Getting-Up Policies for Real-World Humanoid Robots

Xialin He*[1] Runpei Dong*[1] Zixuan Chen[2] Saurabh Gupta[1]
[1]University of Illinois Urbana-Champaign [2]Simon Fraser University



(a) Getting Up from Supine (i.e., Lying Face Up) Poses



(b) Rolling Over from Prone (i.e., Lying Face Down) Poses

(c) Getting up on Different Terrains

Fig. 1: HUMANUP provides a simple and general two-stage training method for humanoid getting-up tasks, which can be directly deployed on Unitree G1 humanoid robots [75]. Our policies showcase robust and smooth behavior that can get up from diverse lying postures (both supine and prone) on varied terrains such as grass slopes and stone tiles.

***Abstract*—Automatic fall recovery is a crucial prerequisite before humanoid robots can be reliably deployed. Hand-designing controllers for getting up is difficult because of the varied configurations a humanoid can end up in after a fall and the challenging terrains humanoid robots are expected to operate on. This paper develops a learning framework to produce controllers that enable humanoid robots to get up from varying configurations on varying terrains. Unlike previous successful applications of learning to humanoid locomotion, the getting-up task involves complex contact patterns (which necessitates accurately modeling of the collision geometry) and sparser rewards. We address these challenges through a two-phase approach that induces a curriculum. The first stage focuses on discovering a good getting-up trajectory under minimal constraints on smoothness or speed / torque limits. The second stage then refines the discovered motions into deployable (*i.e.* smooth and slow) motions that are robust to variations in initial configuration and terrains. We find these innovations enable a real-world G1 humanoid robot to get up from two main situations that we considered: a) lying face up and b) lying face down, both tested on flat, deformable, slippery surfaces and slopes (*e.g.*, sloppy grass and snowfield). This is one of the first successful demonstrations of learned getting-up policies for human-sized humanoid robots in the real world. Project page: https://humanoid-getup.github.io/**

* Equal contributions.

I. INTRODUCTION

This paper develops learned controllers that enable a humanoid robot to get up from varied fall configurations on varied terrains. Humanoid robots are susceptible to falls, and their reliance on humans for fall recovery hinders their deployment. Furthermore, as humanoid robots are expected to work in environments involving complex terrains and tight workspaces

(*i.e.* challenging scenarios that are too difficult for wheeled robots), a humanoid robot may end up in an unpredictable configuration upon a fall, or may be on an unknown terrain. 26 of the 46 trials at the DARPA Robotics Challenge (DRC) had a fall, and 25 of these falls required human intervention for recovery [44]. The DRC identified fall prevention and recovery as a major topic needing more research. This paper pursues it and proposes a learning-based framework for learning fall recovery policies for humanoid robots under varying conditions.

The need for recovering from varied initial conditions makes it hard to design a fall recovery controller by hand and motivates the need for learning via trial and error in simulation. Such learning has produced exciting results in recent years for locomotion problems involving quadrupeds and humanoids, *e.g.* [47, 64]. Motivated by these exciting results, we started with simply applying the Sim-to-Real (Sim2Real) paradigm for the getting-up problem. However, we quickly realized that the getting-up problem is different from typical locomotion problems in the following three significant ways that made a naive adaptation of previous work inadequate:

a) Non-periodic behavior. In locomotion, contacts with the environment happen in structured ways: cyclic left-right stepping pattern. The getting-up problem doesn't have such a periodic behavior. The contact sequence necessary for getting up itself needs to be figured out. This makes optimization harder and may render phase coupling of left and right feet commonly used in locomotion ineffective.

b) Richness in contact. Different from locomotion, contacts necessary for getting up are not limited to just the feet. Many other parts of the robot are likely already in touch with the terrain. But more importantly, the robot may find it useful to employ its body, outside of the feet, to exert forces upon the environment, in order to get up. Freezing / decoupling the upper body, only coarsely modeling the upper body for collisions, and using a larger simulation step size: the typical design choices made in locomotion, are no longer applicable for the getting up task.

c) Reward sparsity. Designing rewards for getting up is harder than other locomotion tasks. Velocity tracking offers a dense reward and feedback on whether the robot is meaningfully walking forward is available within a few tens of simulation steps. In contrast, many parts of the body make negative progress, *e.g.*, the torso first needs to tilt down for seconds before tilting up to finally get up.

We present HUMANUP, a two-stage reinforcement learning (RL) training framework that circumvents these issues. Stage I targets *solving the task* in easier settings (sparse task rewards with weak regularization), while Stage II makes the learned motion *deployable* (*i.e.*, control should be smooth; velocities and executed torques should be small; *etc*). Discovering the getting-up motion is hard because of sparse and underspecified rewards. Stage I tackles this hard problem without being limited by smoothness in motion or speed / torque limits. Tracking a trajectory is easier as it offers dense rewards. Stage II tackles this easier problem but does it under strict Sim2Real control regularization and randomization of terrains and initial poses. Thus, going from Stage I to Stage II corresponds to a learning curriculum that progresses from simplified $\rightarrow$ full collision mesh, canonical $\rightarrow$ random initial lying posture, and weak $\rightarrow$ strong control regularization, and domain randomization. This amounts to a *hard-to-easy* curriculum on task difficulty (Stage I: getting-up task; Stage II: motion tracking), and an *easy-to-hard* curriculum on regularization and variability (Stage I: weaker, Stage II: stronger).

We conduct experiments in simulation and the real world with the G1 platform from Unitree. In the real world, we find our framework enables the G1 robot to get up from two different poses (supine, *i.e.* lying face up, and prone, *i.e.* lying face down) across six different terrains. This expands the capability of the G1 robot: the manufacturer-provided hand-crafted getting-up controller only successfully gets up from supine poses on a flat surface without bumps. In simulated experiments, our framework can successfully learn getting-up policies that work on varied terrains and varied starting poses.

II. Related Work

We review related works on humanoid control, learning for humanoid control, and work specifically targeted toward fall recovery for legged robots.

A. Humanoid Control

Controlling a high degree of freedom humanoid robots has fascinated researchers for the last several decades. Model-based techniques, such as those based on the Zero Moment Point (ZMP) principle [35, 65, 74, 76], optimization [4, 14, 45], and Model Predictive Control (MPC) [12, 15, 22, 79], have demonstrated remarkable success in fundamental locomotion tasks like walking, running and jumping. However, these approaches often struggle to generalize or adapt to novel environments. In contrast, learning-based approaches have recently made significant strides, continuously expanding the generalization capabilities of humanoid locomotion controllers.

1) Learning for humanoid control: Learning in simulation via reinforcement followed by a sim-to-real transfer has led to many successful locomotion results for quadrupeds [46, 47] and humanoids [2, 8, 29, 62–64]. This has enabled locomotion on challenging in-the-wild terrain [28, 62], agile motions like jumping [48, 81], and even locomotion driven by visual inputs [50, 83]. Researchers have also expanded the repertoire of humanoid motions to skillful movements like dancing and naturalistic walking gaits through use of human mocap or video data [9, 34, 38, 57]. Some works address locomotion and manipulation problems for humanoids simultaneously to enable loco-manipulation controllers in an end-to-end fashion facilitated by teleportation [20, 32, 52]. Notably, these tasks mostly involve contact between the feet and the environment, thus requiring only limited contact reasoning. How to effectively develop controllers for more *contact-rich* tasks like crawling, tumbling, and getting up that require numerous, dynamic, and unpredictable contacts between the whole body and the environment remains under-explored.

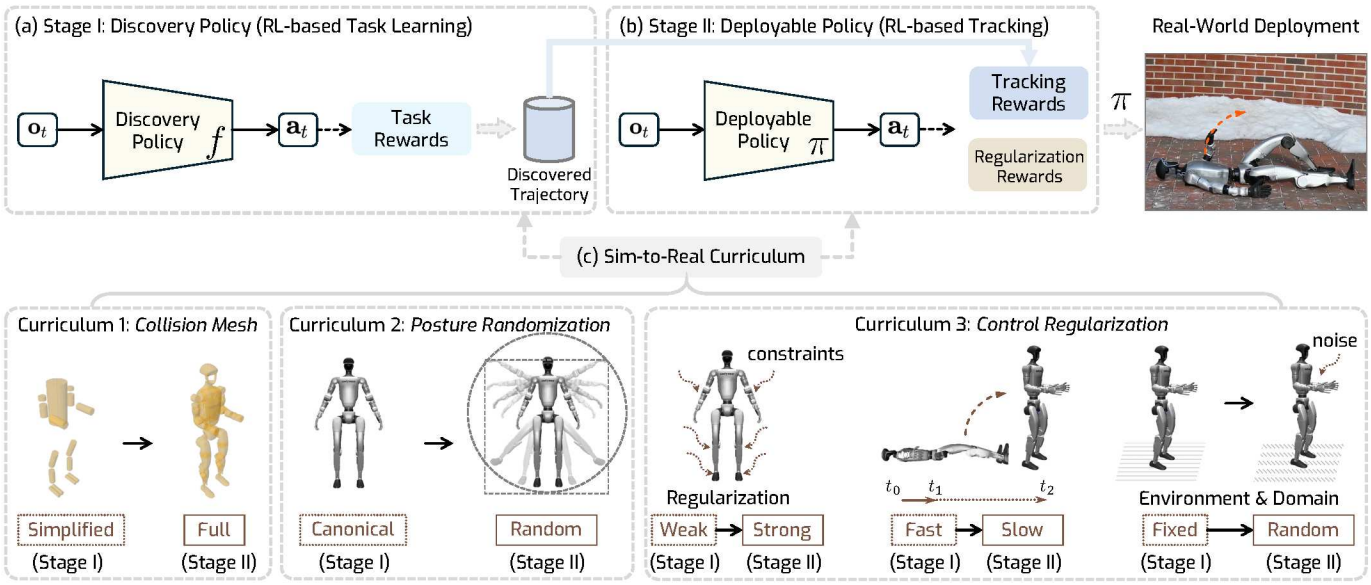


Fig. 2: **HUMANUP system overview**. Our getting-up policy (Sec. III-A) is trained in simulation using two-stage RL training, after which it is directly deployed in the real world. (a) Stage I (Sec. III-B1) learns a discovery policy f that figures out a getting-up trajectory with minimal deployment constraints. (b) Stage II (Sec. III-B2) converts the trajectory discovered by Stage I into a policy π that is deployable, robust, and generalizable. This policy π is trained by learning to track a slowed down version of the discovered trajectory under strong control regularization on varied terrains and from varied initial poses. (c) The two-stage training induces a curriculum (Sec. III-C). Stage I targets motion discovery in easier settings (simpler collision geometry, same starting poses, weak regularization, no variations in terrain), while Stage II solves the task of making the learned motion deployable and generalizable.

B. Legged robots fall recovery

Humanoid robots are vulnerable to falls due to under-actuated control dynamics, high-dimensional states, and unstructured environments [27, 30, 35, 36, 42, 44], making the ability to recover from falling of great significance. Over the years, this problem has been tackled in the following ways.

1) Getting up via motion planning: Early work from Morimoto and Doya [58] solved the getting-up problem for a two-joint, three-link walking robot in 2D, and several discrete states are used as subgoals to transit via hierarchical RL. This line of work can be viewed as an application of motion planning by *configuration graph transition* learning [43], where stored robot states between lying and standing are used as graph nodes to transit [21, 40, 41, 69]. More recently, some progress has been made to enable toy-sized humanoid robots to get up [24, 26, 37, 67]. For example, González-Fierro et al. [26] explores getting up from a canonical sitting posture with motion planning by imitating human demonstration with ZMP criterion. To address the high-dimensionality of humanoid configurations, Jeong and Lee [37] leverage bilateral symmetry to reduce the control DoFs by half and a clustering technique is used for further reducing the complexity of configuration space, thereby improving getting-up learning efficiency. However, such state machine learning using predefined configuration graphs may not be sufficient for generalizing to unpredictable initial and intermediate states, which happens when the robot operates on challenging terrains.

2) Hand-designed getting-up trajectories: Another solution, often adopted by commercial products, is to replay a manually designed motion trajectory. For example, Unitree [75] has a getting-up controller built into G1's default controllers. Booster Robotics [1] designed a specific recovery controller for their robots that can help the robot recover from fallen states. Concurrent work from Zhuang and Zhao [82] enables a G1 robot to get up by tracking the getting-up motion of a real human. The main drawback of such pre-defined trajectory getting-up controllers is that they may only handle a limited number of fallen states and lack generalization.

3) Learned getting-up policies for real robots: RL followed by sim-to-real has also been successfully applied for quadruped [39, 47, 55, 77] fall recovery. For example, Lee et al. [47] explore sim2real RL to achieve real-world quadruped fall recovery from complex configurations. Ji et al. [39] train a recovery policy that enables the quadruped to dribble in snowy and rough terrains continuously. Wang et al. [77] develop a quadruped recovery policy in highly dynamic scenarios.

4) Learned getting-up policies for character animation: A parallel research effort in character animation, also explores the design of RL-based motion imitation algorithms: DeepMimic [59], AMP [60], PHC [53], among others [5, 11, 23, 54, 72, 78]. These have also demonstrated successful getting-up controllers in simulation. By tracking user-specified getting-up curves, Frezzato et al. [17] enable humanoid characters to get up by synthesizing physically plausible motion. Without

recourse to mocap data, such naturalistic getting-up controllers for simulated humanoid characters can also be developed with careful curriculum designs [70]. Some works explore sampling-based methods for addressing contact-rich character locomotion, including getting up [31, 49, 61], while some works have demonstrated success in humanoid getting up with online model-predictive control [71]. It is worth noticing, however, that these works use humanoid characters with larger DoFs compared to humanoid robots (*e.g.*, 69 DoFs in SMPL [51]) and use simplified dynamics. As a result, learned policies operate body parts at high velocities and in infeasible ways, leading to behavior that cannot be transferred into the real world directly. Hence, developing generalizable recovery controllers for humanoid robots remains an open problem.

III. HumanUP: Sim-to-Real Humanoid Getting Up

Our goal is to learn a getting-up policy π that enables a humanoid to get up from arbitrary initial postures. We consider getting up from two families of lying postures: a) supine poses (*i.e.* lying face up) and b) prone poses (*i.e.* lying face down). Getting up from these two groups of postures may require different behaviors, which makes it challenging to learn a single policy that handles both scenarios. To tackle this issue, we decompose the getting-up task from a prone pose to first rolling over and then standing up from the resulting supine posture. Therefore, we aim to learn policies for rolling over from a prone pose and getting up from a supine pose separately.

To solve these two tasks, we propose HumanUP, a general learning framework for training getting-up and rolling over policies, which is illustrated in Fig. 2. In Stage I, a discovery policy f is trained to figure out standing-up or rolling-over motions. f is trained without deployment constraints, and only the task and symmetry rewards are used. In stage II, a deployable policy π imitates the rolling-over / getting-up behaviors obtained from stage I under strong control regularization. This deployable policy π is transferred from simulation to the real world as the final policy. We detail the policy model and two-stage training in Sec. III-B, and then discuss the induced curriculum in Sec. III-C.

A. Policy Architecture

HumanUP trains two policy models f and π with RL. Both policy models take observation $\mathbf{o}_t = [\mathbf{z}_t, \mathbf{s}_t, \mathbf{s}_{t-10:t-1}] \in \mathbb{R}^{868}$ as input and output action $\mathbf{a}_t \in \mathbb{R}^{23}$, where $\mathbf{s}_t \in \mathbb{R}^{74}$ is the proprioceptive information, $\mathbf{s}_{t-10:t-1}$ is the 10 steps history states. $\mathbf{z}_t \in \mathbb{R}^{54}$ are the encoded environment extrinsic latents that are predicted from observation history and learned using regularized online adaptation [19]. The proprioceptive information $\mathbf{s}_t$ consists of the robot's `roll` and `pitch`, angular velocity, DoF velocities, and DoF positions. Such proprioceptive information can be accurately obtained in the real world, and we find that this is sufficient for the robot to infer the overall posture. We do not use any linear velocity and `yaw` information as it is difficult to reliably estimate them in the real world [32, 33].

The policy models are implemented as MLPs and trained via PPO [66]. The optimization maximizes the expected γ-discounted policy return within T episode length: $\mathbb{E}\left[\sum_{t=1}^{T} \gamma^{t-1} r_t\right]$, where r_t is the reward at timestamp t.

B. Two-Stage Policy Learning

1) Stage I: Discovery Policy: This stage discovers getting-up / rolling-over behavior efficiently without deployment constraints. We use the following task rewards with very weak regularization to train this discovery policy f. Timestep t and reward weight terms are omitted for simplicity. The precise expressions for each reward term and their weights are provided in Sec. A.1.

Rewards for Getting Up: $r_{\text{up}} = r_{\text{height}} + r_{\Delta\text{height}} + r_{\text{uprightness}} + r_{\text{stand\_on\_feet}} + r_{\Delta\text{feet\_contact\_forces}} + r_{\text{symmetry}}$, where

- r_{height} encourages the robot's height to be close to a target height when standing;
- $r_{\Delta\text{height}}$ encourages the robot to continuously increasing its height;
- $r_{\text{uprightness}}$ encourages the robot to increase the z-component of the projected gravity,[1] so that the robot stands upright;
- $r_{\text{stand\_on\_feet}}$ encourages the robot to stand on both feet;
- $r_{\Delta\text{feet\_contact\_forces}}$ encourages the robot to increase contact forces applied to the feet continuously;
- r_{symmetry} reduces the search space by *encouraging (but not requiring)* the robot to output bilaterally symmetric actions. Past work [37, 68] employed hard symmetry which improves RL sample efficiency at the cost of limiting robots' DoFs and generalization. Our *soft symmetry reward* partially leverages the benefit but mitigates the limitation.

Rewards for Rolling Over: $r_{\text{roll}} = r_{\text{gravity}}$, which encourages the robot to change its body orientation so that its projected gravity is close to the projected gravity when lying face up.

2) Stage II: Deployable Policy: This stage trains policy π that will be directly deployed in the real world. Policy π is trained to imitate an $8\times$ slowed-down version of the state trajectories discovered in Stage I, while also respecting strong regularization to ensure Sim2Real transferability. We use the typical regularization rewards and describe them in Sec. A.2. Below, we describe the tracking reward.

Tracking Rewards: r_{tracking} encourages the robot to act close to the given motion trajectory derived from the discovered motion. $r_{\text{tracking}} = r_{\text{tracking\_DoF}} + r_{\text{tracking\_body}}$, where

- $r_{\text{tracking\_DoF}}$ encourages the robot to move to the same DoF position as the reference motion, and
- $r_{\text{tracking\_body}}$ encourages the robot to move the body to the same position as the reference. Specifically, $r_{\text{tracking\_body}}$ becomes two different rewards to encourage tracking upright posture ($r_{\text{head\_height}}$) and correct head orientation ($r_{\text{head\_gravity}}$) for getting-up and rolling-over tasks, respectively.

[1] Projected gravity on a robot part is the gravity vector transformed from the world frame to the part's local frame.

C. Stage I to Stage II Curriculum

The design of two-stage policy learning induces a *hard-to-easy* curriculum [7]. Stage I targets motion discovery in easier settings (weak regularization, no variations in terrain, same starting poses, simpler collision geometry). Once motions have been discovered, Stage II solves the task of making the learned motion deployable and generalizable. As our experiments will show, splitting the work into two phases is crucial for successful learning. Specifically, complexity increases from Stage I to Stage II in the following ways:

1) Collision mesh: As shown in Fig. 2, Stage I uses a simplified collision mesh for faster motion discovery, while Stage II uses the full mesh for improved Sim2Real performance.

2) Posture randomization: Stage I learns to get up (and roll over) from a canonical pose, accelerating learning, while Stage II starts from arbitrary initial poses, enhancing generalization. To further speed up Stage I, we mix in standing poses. For Stage II, we generate a dataset $\mathcal{P}$ of 20K supine poses $\mathcal{P}_{\text{supine}}$ and 20K prone poses $\mathcal{P}_{\text{prone}}$ by randomizing initial DoFs from canonical lying poses, dropping the humanoid from 0.5m, and simulating for 10s to resolve self-collisions. We use 10K poses from each set for training and the rest for evaluation.

3) Control Regularization and Terrain Randomization: For Sim2Real transfer, we use the following control regularization terms and environment randomization in Stage II:

• **Weak $\rightarrow$ strong control regularization.** Weak control regularization in Stage I enables discovery of getting-up / rolling-over motion, while strong control regularization (via smoothness rewards, DoF velocity penalties, *etc*, see the full list in Sec. A.2) in Stage II encourages more deployable action.

• **Fast $\rightarrow$ slow motion speed.** Without strong control regularization, Stage I discovers a fast but unsafe getting-up motion (<1s), infeasible for real-world deployment. To address this, we slow it to 8s via interpolation, providing stable tracking targets for Stage II, which better aligns with its control regularization.

• **Fixed $\rightarrow$ random dynamics and domain parameters.** Stage II also employs domain and dynamics randomization via terrain randomization and noise injection. Such randomization has been shown to play a vital role in successful Sim2Real [73].

IV. Implementation Details

A. Platform Configurations

We use the Unitree G1 platform [75] in all real-world and simulation experiments. G1 is a medium-sized humanoid robot with 29 actuatable degrees of freedom (DoF) in total. Specifically, the upper body has 14 DoFs, the lower body has 12 DoFs, and the waist has 3 DoFs. As getting up does not involve object manipulation, we disable the 3 DOFs in the wrists, resulting in 23 DoFs in total. Unlike previous robots, G1 has waist `yaw` and `roll` DoFs, and we find them useful for our getting-up task. The robot has an IMU sensor for `roll` and `pitch` states, and the joint states can be obtained from the motor encoders. We use position control where the torque is derived by a PD controller operating at 50 Hz.

B. Simulation Configurations

We use Isaac Gym [56] for simulated training and evaluation. We use a URDF with simplified collision for stage-I training and the official whole-body URDF from Unitree [75] for stage-II. To accurately model the numerous contacts between the humanoid and the ground, we use a high simulation frequency of 1000 Hz, while the low-level PD controller frequency operates at 50 Hz. More details can be found in Sec. C.

V. Simulation Results

A. Tasks

We evaluate three tasks involved in the humanoid getting-up process: ❶ *getting up from supine poses*, ❷ *rolling over from prone to supine poses*, and ❸ *getting up from prone poses* which can be addressed by solving task ❷ and task ❶ consecutively. Simulation tests are conducted with the full URDF.

B. Baselines

We compare to the following baselines,

a) **RL with Simple Task Rewards (Tao et al. [70]):** This policy is trained with RL using rewards from Tao et al. [70] originally designed for physically animated characters instead of humanoid robots. Similar to our method, this baseline applies a three-stage strong-to-weak torque limit and motion speed curriculum for getting-up policy learning. Because [70] does not consider sim2real deployment regularization and requirements (*e.g.*, smoothness and collision mesh usage), policies learned through their scheme aren't appropriate for real-world humanoid deployment.

b) **HumanUP w/o Stage II:** Our policy trained with only stage I, where no deployment constraints are applied.

c) **HumanUP w/o Full URDF:** Our policy trained with two stages, but stage II uses the simplified collision mesh.

d) **HumanUP w/o Posture Randomization:** Our policy trained on a single canonical lying posture without any randomization of initialization postures.

e) **HumanUP w/ Hard Symmetry:** Our policy trained using a humanoid with a symmetric controller. This symmetric controller follows the symmetry control principle of the manufacturer-provided controller baseline described in real-world experiments, which leads to bilaterally symmetric control. We set all `pitch` DoFs actions to be the same between the left and the right DoFs, while flipping the directions of all the `roll` and `yaw` actions.

f) **HumanUP w/o Two-Stage Learning:** Our policy trained in a single stage with the full collision mesh, posture randomization, and all rewards and regularization terms applied at the same time.

C. Metrics

• **Task Success.** i) Task success rate *Success (%)*: For the getting-up task, the robot's head height must be $\geq$ 1.1m at termination, thus, the robot needs to continue to stand for success. For the rolling-over task, the cosine between the robot's base, knee, and torso orientation and the target orientation when

TABLE I: **Simulation results**. We compare HUMANUP with several baselines on the `held-out` split of our curated posture set $\mathcal{P}_{\text{supine}}$ and $\mathcal{P}_{\text{prone}}$ using full URDF. All methods are trained on the `training` split of our posture set $\mathcal{P}$, except for methods HUMANUP w/o Stage II and w/o posture randomization. HUMANUP solves task ❸ by solving task ❷ and task ❶ consecutively. We do not include the results of baseline 6 (HUMANUP w/o Two-Stage Learning) as it cannot solve the task. † Tao et al. [70] is trained to directly solving task ❸ as it does not have a rolling over policy. SIM2REAL column indicates whether the method can transfer to the real world successfully. We tested all methods in the real world for which the SMOOTHNESS and SAFETY metrics are reasonable, and SIM2REAL is false if deployment wasn't successful. Metrics are introduced in Sec. V-C.

	SIM2REAL	TASK		SMOOTHNESS			SAFETY	
		Success↑	Task Metric↑	Action Jitter↓	DoF Pos Jitter↓	Energy↓	$\mathcal{S}^{\text{Torque}}_{0.8,0.5}$ ↑	$\mathcal{S}^{\text{DoF}}_{0.8,0.5}$ ↑
❶ Getting Up from Supine Poses								
Tao et al. [70]	✗	92.62 ± 0.54	**1.27 ± 0.00**	5.39 ± 0.01	0.48 ± 0.00	650.19 ±1.26	0.72 ± 3.10e-4	0.73 ± 1.39e-4
HUMANUP w/o Stage II	✗	24.82 ± 0.25	0.83 ± 0.00	13.70 ± 0.18	0.71 ± 0.00	1311.22 ± 8.57	0.57 ± 1.45e-3	0.67 ± 5.56e-4
HUMANUP w/o Full URDF	✗	93.95 ± 0.24	1.22 ± 0.00	0.71 ± 0.00	0.11 ± 0.00	104.14 ± 0.57	0.92 ± 8.36e-5	0.77 ± 9.40e-5
HUMANUP w/o Posture Rand.	✓	65.39 ± 0.50	1.09 ± 0.04	0.75 ± 0.05	0.15 ± 0.03	141.52 ± 0.61	0.91 ± 2.32e-4	0.74 ± 7.24e-5
HUMANUP w/ Hard Symmetry	✓	84.56 ± 0.11	1.23 ± 0.00	0.97 ± 0.01	0.22 ± 0.00	182.39 ± 0.22	0.89 ± 1.70e-5	0.78 ± 8.81e-5
HUMANUP	✓	**95.34 ± 0.12**	1.24 ± 0.00	**0.56 ± 0.01**	**0.10 ± 0.00**	**91.74 ± 0.33**	**0.93 ± 1.55e-5**	**0.78 ± 4.15e-5**
❷ Rolling Over from Prone to Supine Poses								
HUMANUP w/o Stage II	✗	43.48 ± 0.41	0.91 ± 0.00	3.32 ± 0.31	0.40 ± 0.05	1684.66 ± 0.43	0.65 ± 6.28e-4	0.72 ± 7.18e-5
HUMANUP w/o Full URDF	✗	87.73 ± 0.33	0.97 ± 0.00	0.33 ± 0.00	0.07 ± 0.00	59.01 ± 0.05	0.93 ± 7.91e-5	0.75 ± 9.98e-5
HUMANUP w/o Posture Rand.	✓	37.27 ± 1.14	0.77 ± 0.01	0.77 ± 0.01	0.15 ± 0.00	234.46 ± 1.00	0.90± 4.98e-4	0.72 ± 2.04e-4
HUMANUP w/ Hard Symmetry	✓	75.53 ± 0.25	0.60 ± 0.00	**0.31 ± 0.00**	0.09 ± 0.00	84.95 ± 0.33	0.95 ± 3.12e-5	0.76 ± 2.49e-5
HUMANUP	✓	**94.40 ± 0.21**	**0.99 ±0.00**	**0.31 ± 0.00**	**0.06 ± 0.00**	**57.08 ± 0.20**	**0.95 ± 1.51e-4**	**0.76 ± 2.48e-5**
❸ Getting Up from Prone Poses								
Tao et al. [70]†	✗	**98.99 ± 0.20**	**1.26 ± 0.00**	11.73 ± 0.01	0.76 ± 0.00	1015.27 ± 0.65	0.67 ± 2.24e-4	0.68 ± 6.41e-5
HUMANUP w/o Stage II	✗	27.59 ± 0.28	0.82 ± 0.00	5.56 ± 0.36	0.45 ± 0.04	1213.07 ± 5.56	0.67 ± 4.71e-3	0.71 ± 2.17e-3
HUMANUP w/o Full URDF	✗	89.59 ± 0.29	1.23 ± 0.00	0.44 ± 0.01	0.08 ± 0.00	77.61 ± 0.86	0.92 ± 2.88e-5	0.75 ± 3.19e-5
HUMANUP w/o Posture Rand.	✓	30.25 ± 0.24	0.87 ± 0.02	1.05 ± 0.01	0.15 ± 0.00	208.23 ± 1.27	0.90 ± 3.06e-4	0.73 ± 1.01e-4
HUMANUP w/ Hard Symmetry	✓	67.12 ± 0.34	1.09 ± 0.01	0.94 ± 0.01	0.23 ± 0.01	196.17 ± 3.68	0.91 ± 3.54e-5	0.76 ± 4.45e-5
HUMANUP	✓	92.10 ± 0.46	1.24 ± 0.00	**0.39 ± 0.01**	**0.07 ± 0.00**	**69.98 ± 0.45**	**0.94 ± 1.82e-4**	**0.77 ± 3.70e-4**

lying face up should be ≥ 0.9, thus, the robot needs to move until lying face up. ii) *Task Metrics*: the head height (m) for the getting-up task, and the cosine of the angle between the robot's torso X-axis (sticking out to the front from the torso) and the gravity, for the rolling-over task.

• **Smoothness.** We measure the *Action Jitter* (rad/s^3), *DoF Pos Jitter* (rad/s^3), and mean *Energy* ($N \cdot m \cdot rad/s$) for action smoothness evaluation [18]. The jitter metrics are computed as the third derivative values [16], which indicate the coordination stability of body movements.

• **Safety.** We introduce safety scores $\mathcal{S}^{\text{Torque}}_{\delta,\alpha} \in [0,1]$ and $\mathcal{S}^{\text{DoF}}_{\delta,\beta} \in [0,1]$ that measure the relative magnitude of commanded torque / DoF compared to the torque and DoF limits, where δ is a safety threshold. This is essential for robotic safety during execution, as large torques or DoFs will lead to overheating issues and cause mechanical and motor damage. Formally, these scores are defined as:

$$\mathcal{S}^{\text{Torque}}_{\delta,\alpha} = 1 - \left(\frac{\alpha}{TJ} \sum_{t,j} \frac{|\tau_{t,j}|}{\tau^{\max}_j} + \frac{1-\alpha}{TJ} \sum_{t,j} \mathbb{1}\left(\frac{|\tau_{t,j}|}{\tau^{\max}_j} > \delta \right) \right),$$

$$\mathcal{S}^{\text{DoF}}_{\delta,\beta} = 1 - \left(\frac{\beta}{TJ} \sum_{t,j} \frac{|q_{t,j}|}{q^{\max}_j} + \frac{1-\beta}{TJ} \sum_{t,j} \mathbb{1}\left(\frac{|q_{t,j}|}{q^{\max}_j} > \delta \right) \right),$$

where $\tau_{t,j}$ and $q_{t,j}$ denote the applied torque and joint displacement at time step t for joint j, respectively. $\tau^{\max}_j$ and $q^{\max}_j$ represent their respective limits, T is the total number of time steps, and J is the total number of joints. The threshold δ determines when a torque or displacement is considered excessive. The indicator function $\mathbb{1}(\cdot)$ returns 1 if the condition is met and 0 otherwise. The parameters $\alpha, \beta \in [0,1]$ control the trade-off between peak and prolonged violations, ensuring a balanced assessment of safety risks. In this paper, we use $\delta = 0.8$, $\alpha = 0.5$, $\beta = 0.5$ as default during evaluation.

D. Results and Analysis

Tab. I presents results based on policies tested on `held-out` subsets of our curated initial posture set $\mathcal{P}$, *i.e.* 10K val samples each from $\mathcal{P}_{\text{supine}}$ and $\mathcal{P}_{\text{prone}}$. Fig. 4 shows the learning curve for the getting-up task, where the termination base height reflects the robot's ability to lift its body, and body uprightness indicates whether it achieves a stable standing posture.

1) Ignoring Torque / Control Limits Leads to Undeployable Policies: While [70] and HUMANUP achieve similar success rates, the smoothness and safety metrics for [70] are significantly worse than HUMANUP. For example, the average action jitter metric is nearly 18× higher than HUMANUP. Actions from [70] are highly unstable and unsafe and thus cannot be safely deployed to the real robot. Furthermore, [70] learns a very fast getting-up motion that keeps jumping after getting up. See visualization [70]'s getting up motion in Sec. B.1. A similar trend can be seen when comparing HUMANUP to HUMANUP w/o Stage II. While HUMANUP w/o Stage II also solves the task to some extent, it achieves unsatisfying smoothness and safety metrics similar making it inappropriate for real-world deployment. Thus, the regularization imposed in Stage II is essential to making policies more amenable to Sim2Real transfer.

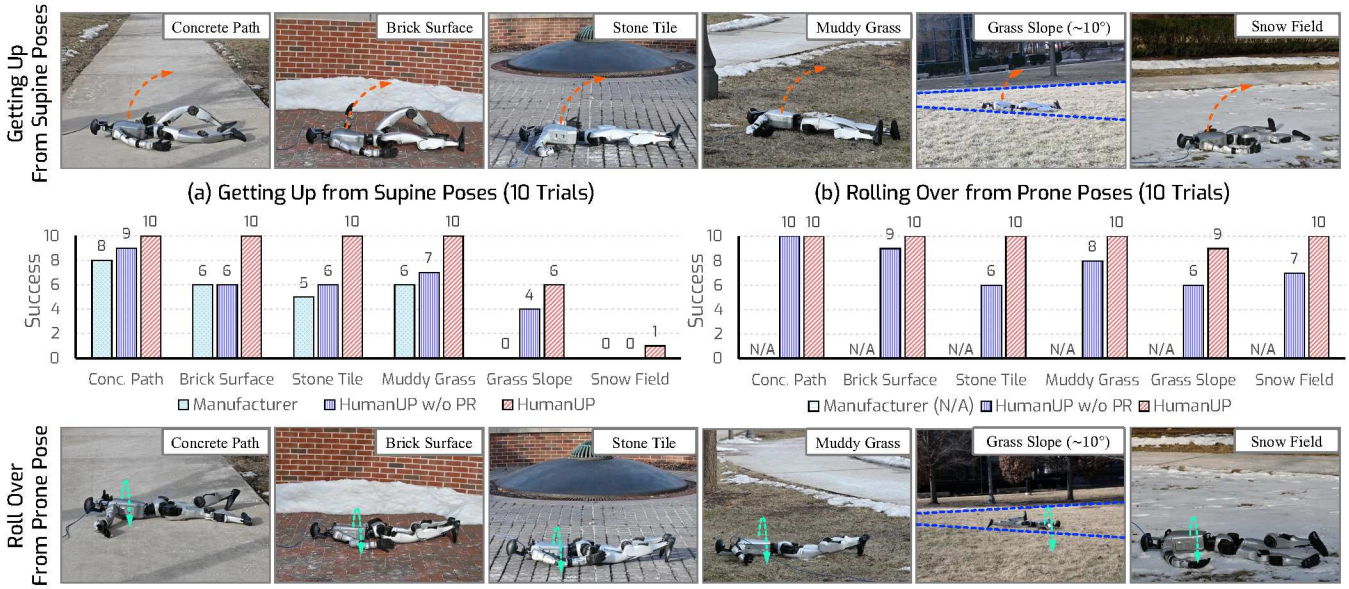


Fig. 3: **Real-world results**. We evaluate HUMANUP (ours) in several real setups that span diverse surface properties, including both man-made and natural surfaces, and cover a wide range of roughness (rough concrete to slippery snow), bumpiness (flat concrete to tiles), ground compliance (completely firm concrete to being swampy muddy grass), and slope (flat to about 10°). We compare HUMANUP with G1's manufacturer-provided controller and HUMANUP w/o posture randomization (PR). HUMANUP succeeds more consistently (78.3% vs 41.7%) and can solve terrains that the manufacturer-provided controller can't.

2) Importance of Learning via a Curriculum: So, while it is clear that we need to incorporate strong control regularization for good safety metrics and Sim2Real transfer, our 2 stage process is better than doing it in a single stage. In fact, as plotted in Fig. 4, HUMANUP w/o Two Stage Learning where the policy is trained in a single stage using all sim2real regularization fails to solve the task. This is because the strict Sim2Real regularization makes task learning extremely challenging. Our two-stage curriculum successfully incorporates both aspects: it learns to solve the task, but the policy also operates safely.

3) Full URDF vs. Simplified URDF: Somewhat surprisingly, even though HUMANUP w/o Full URDF was trained without the full URDF mesh, it generalizes fine when tested with the full URDF in simulation, as reported in Tab. I. However, we found poor transfer of this policy to the real world. It failed on all 5 trials on a simple flat terrain. We believe the poor real-world performance was because of the mismatch between the contact it was expecting and the contact that actually happened.

4) Posture randomization helps: HUMANUP w/o posture randomization (PR) works much worse than HUMANUP, suggesting that PR is necessary for generalizable control.

5) Soft symmetry vs. hard symmetry: Compared to HUMANUP w/ Hard Symmetry, HUMANUP achieves better task success in Tab. I, particularly for the rolling-over task, which is very difficult with symmetric commands.

VI. REAL WORLD RESULTS

We also tested HUMANUP policies in the real world on G1 robot. Our real-world test bed consists of 6 different terrains shown in Fig. 3: concrete, brick, stone tiles, muddy grass, grassy slope, and a snow field. These terrains span diverse surface properties, including both man-made and natural surfaces, and cover a wide range of roughness (rough concrete to slippery snow), bumpiness (flat concrete to tiles), ground compliance (completely firm concrete to being swampy muddy grasp), and slope (flat to $\sim 10°$). We tested two tasks: a) getting up from supine poses, and b) rolling over from prone to supine poses.

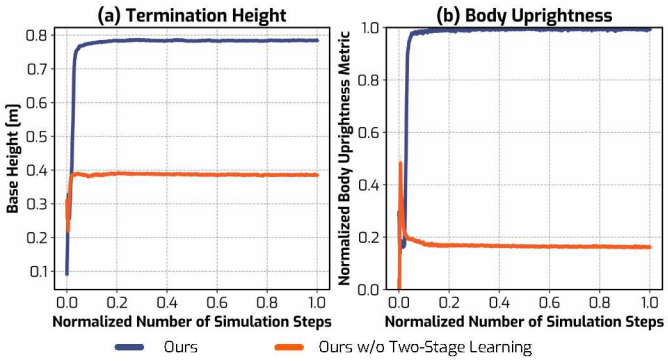


Fig. 4: **Learning curve**. (a) Termination height of the torso, indicating whether the robot can lift the body. (b) Body uprightness, computed as the projected gravity on the z-axis, normalized to $[0, 1]$ for better comparison. The overall number of simulation sampling steps is about 5B, normalized to $[0, 1]$.

We compare our policy with 1) **Manufacturer-provided Controller** and 2) a high-performing ablation of HUMANUP (**HUMANUP w/o posture randomization**). The manufacturer-provided controller, which comes with the robot G1, tracks a hand-crafted trajectory in three phases shown in Fig. 5: Phase

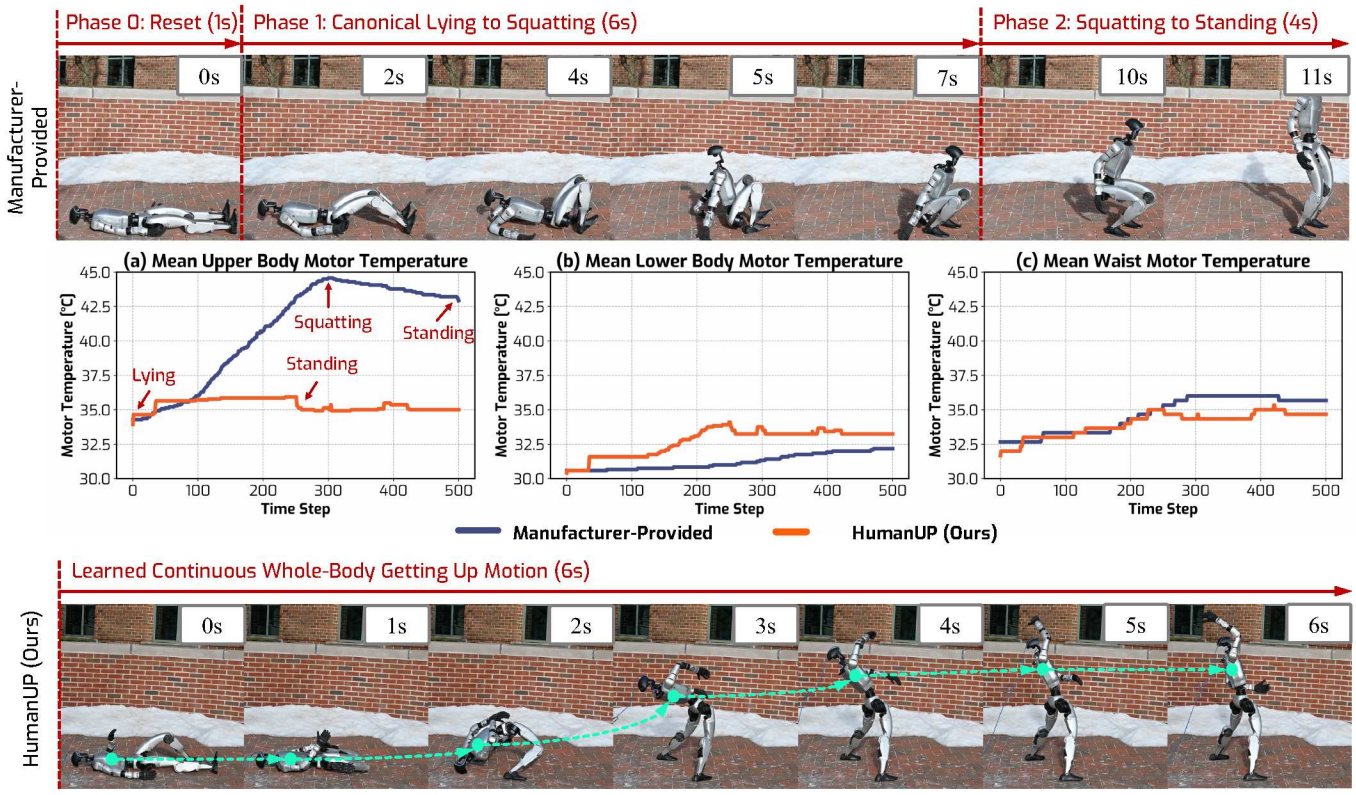


Fig. 5: **Getting up execution comparison with G1's manufacturer-provided controller**. The manufacturer-provided controller uses a handcrafted motion trajectory, which can be divided into three phases, while our HUMANUP learns a continuous and more efficient whole-body getting-up motion. Our HUMANUP enables the humanoid to get up within 6 seconds, half of the manufacturer-provided controller's 11 seconds of control. (a), (b), and (c) record the corresponding mean motor temperature of the upper body, lower body, and waist, respectively. G1's manufacturer-provided controller's execution causes the arm motors to heat up significantly, whereas our policy makes more use of the leg motors that are stronger (higher torque limit of 83N as opposed to 25N for the arm motors) and thus able to take more load.

0 brings the robot to a canonical lying pose; Phase 1 first props up and then slides the torso forward using hands, followed by bending legs to squat; Phase 2 uses its waist to tilt up the torso to stand up from squatting. Motions in phase 1 and phase 2 are *symmetric*, and this controller only works for supine poses.

A. Results

Fig. 3 presents experimental results. Overall, we find that HUMANUP policies perform better than the manufacturer-provided controller and HUMANUP without posture randomization (PR). We discuss the results and observed behavior further.

1) Getting up from supine poses: The manufacturer-provided controller works under nominal conditions, *i.e.*, firm, flat concrete ground with a reasonable friction value. However, it starts to fail on more challenging terrains. For the bumpier and rougher terrains (brick surface and stone tiles), the arms may get stuck between bumps, causing failures. On slopes, the robot fails to squat or hoist itself up due to both the resistance of the grassland and the unstable squatting pose prone to falling caused by slopes. On the compliant ground, the robot gets destabilized. On slippery snow, the robot slips.

Both versions of HUMANUP outperform the manufacturer-provided controller. Trained with terrain and domain randomization, they are robust to real-world variations such as slipperiness, bumps, and slopes. Dynamics randomization further enhances resilience to minor perturbations like slippage or ground compliance. The full method, incorporating posture randomization, performs better than the variant without it, as it is specifically trained to handle diverse initial configurations. Overall, HUMANUP achieves a 78.3% getting-up success rate.

2) Rolling over from prone to supine poses: Findings are similar for the rolling over task. As noted, the manufacturer-provided controller can't handle this situation. The full model exhibits more robust performance than the model trained without posture randomization. Rolling over seems to be easier than getting up, HUMANUP achieves a 98.3% success rate.

B. Motion analysis

Fig. 5 shows the motion and motor temperatures for the manufacturer-provided controller and HUMANUP policy.

1) Motor temperature: The manufacturer-provided controller uses the arms during Phase 1 of getting up. Fig. 5 (a) shows that the default controller's execution causes the arm motors

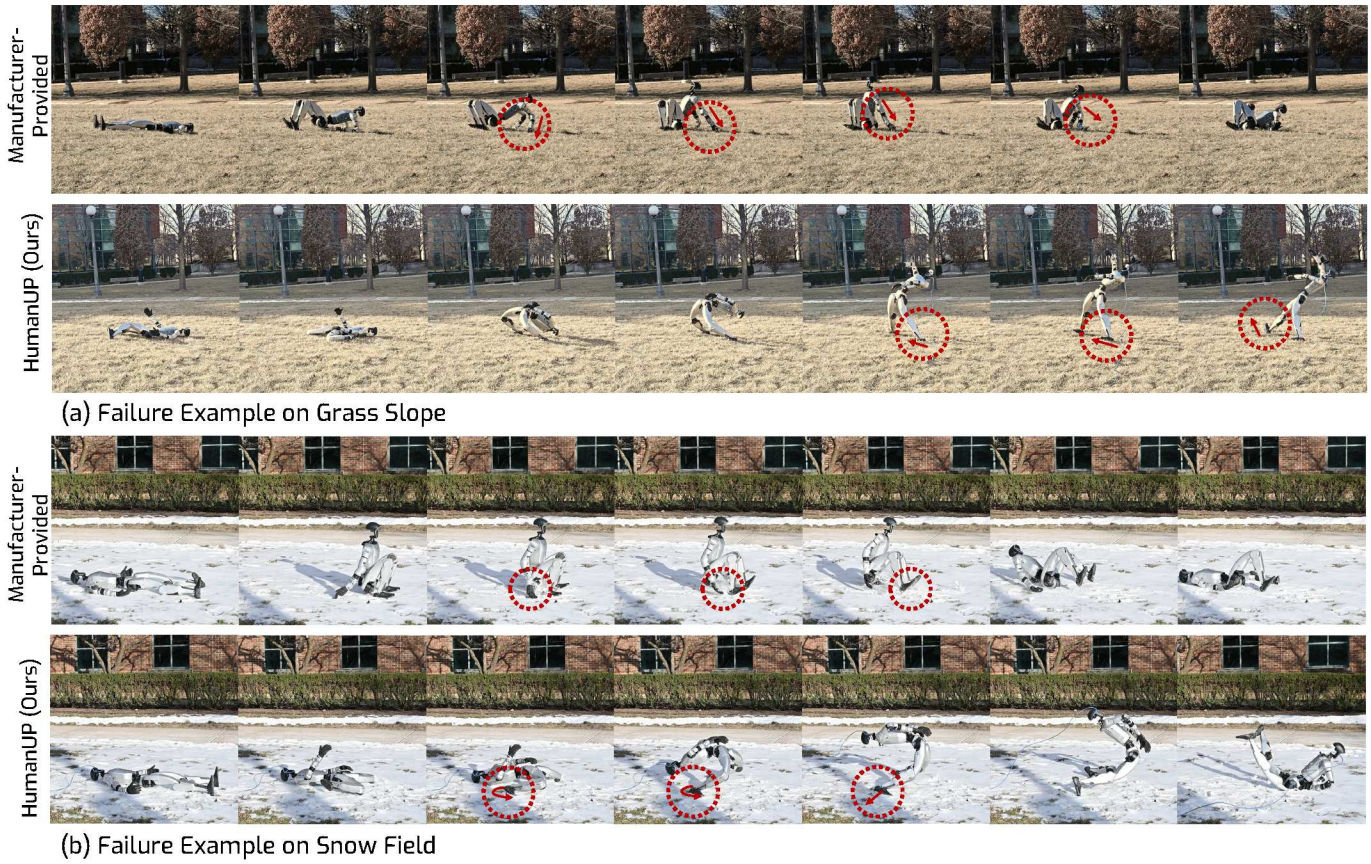


Fig. 6: **Qualitative examples of failure modes on grass slope and snow field.** G1's manufacturer-provided controller isn't able to squat on the sloping grass and slips on the slope. HUMANUP policy can partially get up on both the slope and the snow, but falls due to unstable foot placement on the slope and slippage on the snow.

to heat up significantly more when compared to HUMANUP execution. Our policy makes more use of the leg motors that are stronger (higher torque limit of $83N$ as opposed to $25N$ for the arm motors) and thus able to take more load.

2) Efficiency: HUMANUP gets the robot to stand successfully within about 6 seconds through a smooth and continuous motion, which is over $2\times$ more efficient than the manufacturer-provided controller, which takes nearly 11 seconds.

C. Failure Mode Analysis

Fig. 6 shows example failure modes for the manufacturer-provided controller and HUMANUP on challenging terrains. Fig. 6 (a) shows that the manufacturer-provided controller tries to utilize the robot's hands to squat, while the sloping ground prevents it from getting to the full squatting pose due to high friction and weak waist torques to move against the dumping tendency. In contrast, HUMANUP manages to lift the body, while the sloping ground sometimes causes an unstable foot orientation. Fig. 6 (b) shows that on even more challenging terrains like snow fields, both manufacturer-provided and HUMANUP controllers may fail due to the slippery and deformable ground.

VII. LIMITATIONS

HUMANUP has several limitations: 1) Motions discovered in Stage I could be incompatible with stronger control regularization used in Stage II. We didn't encounter this issue in our experiments, possibly because of the weak control regularization used in Stage I and the use of $8\times$ slower motion in Stage II. 2) HUMANUP depends on high-performance physics simulators (IsaacGym [56]) running at high frequency (*e.g.*, 1 kHz). Simulation speed and fidelity for more complex tasks involving perception and contacts remain a challenge. Recent advances such as Genesis [6], Mujoco Playground [80], and Roboverse [25] could help address these limitations. 3) The RL formulation in HUMANUP is under-specified [3] and may lead to reward hacking [13], complicating precise alignment with natural human behaviors. For instance, our learned motions sometimes include unnatural hand raising for balance. 4) Extending HUMANUP to handle more complex terrains like stairs or uneven surfaces remains under-explored, while humanoid robots may fall more easily on such terrains. Encouraging adaptive behaviors involving strong-arm usage on more powerful platforms may be useful to properly handle such situations.

VIII. Discussion

In this paper, we tackle the problem of *learning* getting-up controllers for real-world humanoid robots. Different from locomotion tasks, getting up involves complex contact patterns that are not known apriori. We develop a two-stage solution for this problem based on reinforcement learning and sim-to-real. Stage I finds a solution under minimal constraints, while Stage II learns to track the trajectory discovered in Stage I under regularization on control and from varied starting poses and on varied terrains. We found this two stage strategy to be effective both in simulation and the real world. Specifically, it enabled us to get a real-world G1 humanoid to stand up from a supine pose and roll over from a supine pose to a prone pose on different terrains and from different starting poses. HumanUP achieves a higher success rate than G1's manufacturer-provided controller and expands the capabilities of the G1 robot.

We hope our learned policies for automatic fall recovery will be useful to researchers and practitioners, while our two-stage learning framework may be helpful for other problems that require figuring out complex contact patterns.

Acknowledgments

This material is based upon work supported by an NSF CAREER Award (IIS2143873) and a DURIP grant (N00014-23-1-2166). We also thank the Coordinated Science Laboratory for providing experimental space.

References

[1] Booster robotics. URL https://www.boosterobotics.com/. 3

[2] Alphonsus Adu-Bredu, Grant Gibson, and Jessy Grizzle. Exploring kinodynamic fabrics for reactive whole-body control of underactuated humanoid robots. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10397–10404. IEEE, 2023. 2

[3] Pulkit Agrawal. The task specification problem. In *Conference on Robot Learning*, pages 1745–1751. PMLR, 2022. 9

[4] Min Sung Ahn. *Development and Real-Time Optimization-based Control of a Full-sized Humanoid for Dynamic Walking and Running*. University of California, Los Angeles, 2023. 2

[5] Anonymous. Hierarchical world models as visual whole-body humanoid controllers. In *The Thirteenth International Conference on Learning Representations*, 2025. 3

[6] Genesis Authors. Genesis: A universal and generative physics engine for robotics and beyond, December 2024. URL https://github.com/Genesis-Embodied-AI/Genesis. 9

[7] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA, 2009. Association for Computing Machinery. 5

[8] Zixuan Chen, Xialin He, Yen-Jen Wang, Qiayuan Liao, Yanjie Ze, Zhongyu Li, S Shankar Sastry, Jiajun Wu, Koushil Sreenath, Saurabh Gupta, et al. Learning smooth humanoid locomotion through lipschitz-constrained policies. *arXiv preprint arXiv:2410.11825*, 2024. 2

[9] Xuxin Cheng, Yandong Ji, Junming Chen, Ruihan Yang, Ge Yang, and Xiaolong Wang. Expressive Whole-Body Control for Humanoid Robots. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. 2

[10] Xuxin Cheng, Kexin Shi, Ananye Agarwal, and Deepak Pathak. Extreme parkour with legged robots. In *IEEE International Conference on Robotics and Automation, ICRA 2024, Yokohama, Japan, May 13-17, 2024*, pages 11443–11450. IEEE, 2024. 15

[11] Nuttapong Chentanez, Matthias Müller, Miles Macklin, Viktor Makoviychuk, and Stefan Jeschke. Physics-based motion capture imitation with deep reinforcement learning. In Panayiotis Charalambous, Yiorgos Chrysanthou, Ben Jones, and Jehee Lee, editors, *Proceedings of the 11th Annual International Conference on Motion, Interaction, and Games, MIG 2018, Limassol, Cyprus, November 08-10, 2018*, pages 1:1–1:10. ACM, 2018. 3

[12] Matthew Chignoli, Donghyun Kim, Elijah Stanger-Jones, and Sangbae Kim. The mit humanoid robot: Design, motion planning, and control for acrobatic behaviors. In *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, pages 1–8, 2021. 2

[13] Jack Clark and Dario Amodei. Faulty reward functions in the wild. 2016. URL https://openai.com/index/faulty-reward-functions/. 9

[14] Devin Crowley, Jeremy Dao, Helei Duan, Kevin Green, Jonathan Hurst, and Alan Fern. Optimizing bipedal locomotion for the 100m dash with comparison to human running. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12205–12211, 2023. 2

[15] Jared Di Carlo, Patrick M. Wensing, Benjamin Katz, Gerardo Bledt, and Sangbae Kim. Dynamic locomotion in the mit cheetah 3 through convex model-predictive control. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9, 2018. 2

[16] Tamar Flash and Neville Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of neuroscience*, 5(7):1688–1703, 1985. 6

[17] Anthony Frezzato, Arsh Tangri, and Sheldon Andrews. Synthesizing get-up motions for physics-based characters. In *Computer Graphics Forum*, volume 41, pages 207–218. Wiley Online Library, 2022. 3

[18] Zipeng Fu, Ashish Kumar, Jitendra Malik, and Deepak Pathak. Minimizing energy consumption leads to the emergence of gaits in legged robots. In Aleksandra Faust, David Hsu, and Gerhard Neumann, editors, *Conference on Robot Learning, 8-11 November 2021, London, UK*, volume 164 of *Proceedings of Machine Learning Research*, pages 928–937. PMLR, 2021. 6

[19] Zipeng Fu, Xuxin Cheng, and Deepak Pathak. Deep

whole-body control: Learning a unified policy for manipulation and locomotion. In Karen Liu and Dana Kulic andD Jeffrey Ichnowski, editors, *Conference on Robot Learning, CoRL 2022, 14-18 December 2022, Auckland, New Zealand*, volume 205 of *Proceedings of Machine Learning Research*, pages 138–149. PMLR, 2022. 4

[20] Zipeng Fu, Qingqing Zhao, Qi Wu, Gordon Wetzstein, and Chelsea Finn. Humanplus: Humanoid shadowing and imitation from humans. In *8th Annual Conference on Robot Learning*, 2024. 2

[21] Kiyoshi Fujiwara, Fumio Kanehiro, Shuuji Kajita, Kenji Kaneko, Kazuhito Yokoi, and Hirohisa Hirukawa. Ukemi: Falling motion control to minimize damage to biped humanoid robot. In *IEEE/RSJ international conference on Intelligent robots and systems*, volume 3, pages 2521–2526. IEEE, 2002. 3

[22] Manuel Y Galliker, Noel Csomay-Shanklin, Ruben Grandia, Andrew J Taylor, Farbod Farshidian, Marco Hutter, and Aaron D Ames. Planar bipedal locomotion with nonlinear model predictive control: Online gait generation using whole-body dynamics. In *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*, pages 622–629. IEEE, 2022. 2

[23] Jiawei Gao, Ziqin Wang, Zeqi Xiao, Jingbo Wang, Tai Wang, Jinkun Cao, Xiaolin Hu, Si Liu, Jifeng Dai, and Jiangmiao Pang. Coohoi: Learning cooperative human-object interaction with manipulated object dynamics. *arXiv preprint arXiv:2406.14558*, 2024. 3

[24] Clément Gaspard, Marc Duclusaud, Grégoire Passault, Mélodie Daniel, and Olivier Ly. Frasa: An end-to-end reinforcement learning agent for fall recovery and stand up of humanoid robots. *arXiv preprint arXiv:2410.08655*, 2024. 3

[25] Haoran Geng, Feishi Wang, Songlin Wei, Yuyang Li, Bangjun Wang, Boshi An, Charlie Tianyue Cheng, Haozhe Lou, Peihao Li, Yen-Jen Wang, Yutong Liang, Dylan Goetting, Chaoyi Xu, Haozhe Chen, Yuxi Qian, Yiran Geng, Jiageng Mao, Weikang Wan, Mingtong Zhang, Jiangran Lyu, Siheng Zhao, Jiazhao Zhang, Jialiang Zhang, Chengyang Zhao, Haoran Lu, Yufei Ding, Ran Gong, Yuran Wang, Yuxuan Kuang, Ruihai Wu, Baoxiong Jia, Carlo Sferrazza, Hao Dong, Siyuan Huang, Koushil Sreenath, Yue Wang, Jitendra Malik, and Pieter Abbeel. Roboverse: Towards a unified platform, dataset and benchmark for scalable and generalizable robot learning, April 2025. URL https://github.com/RoboVerseOrg/RoboVerse. 9

[26] Miguel González-Fierro, Carlos Balaguer, Nicola Swann, and Thrishantha Nanayakkara. A humanoid robot standing up through learning from demonstration using a multimodal reward function. In *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 74–79. IEEE, 2013. 3

[27] J. W. Grizzle, Jonathan W. Hurst, Benjamin Morris, Hae-Won Park, and Koushil Sreenath. Mabel, a new robotic bipedal walker and runner. In *American Control Conference, ACC 2009. St. Louis, Missouri, USA, June 10-12, 2009*, pages 2030–2036. IEEE, 2009. 3

[28] Xinyang Gu, Yen-Jen Wang, Xiang Zhu, Chengming Shi, Yanjiang Guo, Yichen Liu, and Jianyu Chen. Advancing Humanoid Locomotion: Mastering Challenging Terrains with Denoising World Model Learning. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. 2

[29] Xinyang Gu, Yen-Jen Wang, Xiang Zhu, Chengming Shi, Yanjiang Guo, Yichen Liu, and Jianyu Chen. Advancing humanoid locomotion: Mastering challenging terrains with denoising world model learning. *arXiv preprint arXiv:2408.14472*, 2024. 2

[30] Zhaoyuan Gu, Junheng Li, Wenlan Shen, Wenhao Yu, Zhaoming Xie, Stephen McCrory, Xianyi Cheng, Abdulaziz Shamsah, Robert Griffin, C Karen Liu, et al. Humanoid locomotion and manipulation: Current progress and challenges in control, planning, and learning. *arXiv preprint arXiv:2501.02116*, 2025. 3

[31] Perttu Hämäläinen, Sebastian Eriksson, Esa Tanskanen, Ville Kyrki, and Jaakko Lehtinen. Online motion synthesis using sequential monte carlo. *ACM Transactions on Graphics (TOG)*, 33(4):1–12, 2014. 4

[32] Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris M. Kitani, Changliu Liu, and Guanya Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. In *8th Annual Conference on Robot Learning*, 2024. 2, 4, 15

[33] Tairan He, Zhengyi Luo, Wenli Xiao, Chong Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Learning human-to-humanoid real-time whole-body teleoperation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2024, Abu Dhabi, United Arab Emirates, October 14-18, 2024*, pages 8944–8951. IEEE, 2024. 4

[34] Tairan He, Wenli Xiao, Toru Lin, Zhengyi Luo, Zhenjia Xu, Zhenyu Jiang, Jan Kautz, Changliu Liu, Guanya Shi, Xiaolong Wang, et al. Hover: Versatile neural whole-body controller for humanoid robots. *arXiv preprint arXiv:2410.21229*, 2024. 2

[35] Kazuo Hirai, Masato Hirose, Yuji Haikawa, and Toru Takenaka. The development of honda humanoid robot. In *Proceedings. 1998 IEEE international conference on robotics and automation (Cat. No. 98CH36146)*, volume 2, pages 1321–1326. IEEE, 1998. 2, 3

[36] Masayuki Inaba, Takashi Igarashi, Satoshi Kagami, and Hirochika Inoue. A 35 dof humanoid that can coordinate arms and legs in standing up, reaching and grasping an object. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS'96*, volume 1, pages 29–36. IEEE, 1996. 3

[37] Heejin Jeong and Daniel D. Lee. Efficient learning of stand-up motion for humanoid robots with bilateral symmetry. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2016, Daejeon, South Korea, October 9-14, 2016*, pages 1544–1549. IEEE,

2016. 3, 4

[38] Mazeyu Ji, Xuanbin Peng, Fangchen Liu, Jialong Li, Ge Yang, Xuxin Cheng, and Xiaolong Wang. Exbody2: Advanced expressive humanoid whole-body control. *arXiv preprint arXiv:2412.13196*, 2024. 2

[39] Yandong Ji, Gabriel B Margolis, and Pulkit Agrawal. Dribblebot: Dynamic legged manipulation in the wild. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5155–5162. IEEE, 2023. 3

[40] Fumio Kanehiro, Kenji Kaneko, Kiyoshi Fujiwara, Kensuke Harada, Shuuji Kajita, Kazuhito Yokoi, Hirohisa Hirukawa, Kazuhiko Akachi, and Takakatsu Isozumi. The first humanoid robot that has the same size as a human and that can lie down and get up. In *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, volume 2, pages 1633–1639. IEEE, 2003. 3

[41] Fumio Kanehiro, Kiyoshi Fujiwara, Hirohisa Hirukawa, Shin'ichiro Nakaoka, and Mitsuharu Morisawa. Getting up motion planning using mahalanobis distance. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 2540–2545. IEEE, 2007. 3

[42] Ichiro Kato. Development of wabot 1. *Biomechanism*, 2: 173–214, 1973. 3

[43] Lydia E. Kavraki, Petr Svestka, Jean-Claude Latombe, and Mark H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robotics Autom.*, 12(4):566–580, 1996. 3

[44] Eric Krotkov, Douglas Hackett, Larry Jackel, Michael Perschbacher, James Pippine, Jesse Strauss, Gill Pratt, and Christopher Orlowski. The darpa robotics challenge finals: Results and perspectives. *The DARPA robotics challenge finals: Humanoid robots to the rescue*, pages 1–26, 2018. 2, 3

[45] Scott Kuindersma, Robin Deits, Maurice Fallon, Andrés Valenzuela, Hongkai Dai, Frank Permenter, Twan Koolen, Pat Marion, and Russ Tedrake. Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. *Autonomous robots*, 40:429–455, 2016. 2

[46] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. RMA: rapid motor adaptation for legged robots. In *Robotics: Science and Systems XVII, Virtual Event, July 12-16, 2021*, 2021. 2

[47] Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Sci. Robotics*, 4(26), 2019. 2, 3

[48] Zhongyu Li, Xue Bin Peng, Pieter Abbeel, Sergey Levine, Glen Berseth, and Koushil Sreenath. Robust and versatile bipedal jumping control through reinforcement learning. In Kostas E. Bekris, Kris Hauser, Sylvia L. Herbert, and Jingjin Yu, editors, *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, 2023. 2

[49] Libin Liu, KangKang Yin, Michiel van de Panne, Tianjia Shao, and Weiwei Xu. Sampling-based contact-rich motion control. In *ACM SIGGRAPH 2010 Papers*, SIGGRAPH '10, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781450302104. 4

[50] Junfeng Long, Junli Ren, Moji Shi, Zirui Wang, Tao Huang, Ping Luo, and Jiangmiao Pang. Learning humanoid locomotion with perceptive internal model. *arXiv preprint arXiv:2411.14386*, 2024. 2

[51] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: a skinned multi-person linear model. *ACM Trans. Graph.*, 34(6):248:1–248:16, 2015. 4

[52] Chenhao Lu, Xuxin Cheng, Jialong Li, Shiqi Yang, Mazeyu Ji, Chengjing Yuan, Ge Yang, Sha Yi, and Xiaolong Wang. Mobile-television: Predictive motion priors for humanoid whole-body control. *arXiv preprint arXiv:2412.07773*, 2024. 2

[53] Zhengyi Luo, Jinkun Cao, Alexander Winkler, Kris Kitani, and Weipeng Xu. Perpetual humanoid control for real-time simulated avatars. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 10861–10870. IEEE, 2023. 3

[54] Zhengyi Luo, Jinkun Cao, Josh Merel, Alexander Winkler, Jing Huang, Kris M. Kitani, and Weipeng Xu. Universal humanoid motion representations for physics-based control. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. 3

[55] Yuntao Ma, Farbod Farshidian, and Marco Hutter. Learning arm-assisted fall damage reduction and recovery for legged mobile manipulators. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12149–12155. IEEE, 2023. 3

[56] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance GPU based physics simulation for robot learning. In Joaquin Vanschoren and Sai-Kit Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. 5, 9, 15

[57] Jiageng Mao, Siheng Zhao, Siqi Song, Tianheng Shi, Junjie Ye, Mingtong Zhang, Haoran Geng, Jitendra Malik, Vitor Guizilini, and Yue Wang. Learning from massive human videos for universal humanoid pose control. *arXiv preprint arXiv:2412.14172*, 2024. 2

[58] Jun Morimoto and Kenji Doya. Reinforcement learning of dynamic motor sequence: Learning to stand up. In *Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No. 98CH36190)*, volume 3, pages 1721–1726. IEEE, 1998. 3

[59] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. Deepmimic: example-guided deep rein-

forcement learning of physics-based character skills. *ACM Trans. Graph.*, 37(4):143, 2018. 3

[60] Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (ToG)*, 40(4):1–20, 2021. 3

[61] Cristina Pinneri, Shambhuraj Sawant, Sebastian Blaes, Jan Achterhold, Joerg Stueckler, Michal Rolinek, and Georg Martius. Sample-efficient cross-entropy method for real-time planning. In *Conference on Robot Learning*, pages 1049–1065. PMLR, 2021. 4

[62] Ilija Radosavovic, Sarthak Kamat, Trevor Darrell, and Jitendra Malik. Learning humanoid locomotion over challenging terrain. *arXiv preprint arXiv:2410.03654*, 2024. 2

[63] Ilija Radosavovic, Jathushan Rajasegaran, Baifeng Shi, Bike Zhang, Sarthak Kamat, Koushil Sreenath, Trevor Darrell, and Jitendra Malik. Humanoid locomotion as next token prediction. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[64] Ilija Radosavovic, Tete Xiao, Bike Zhang, Trevor Darrell, Jitendra Malik, and Koushil Sreenath. Real-world humanoid locomotion with reinforcement learning. *Sci. Robotics*, 9(89), 2024. 2

[65] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura. The intelligent asimo: system overview and integration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 2478–2483 vol.3, 2002. 2

[66] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 4

[67] Sebastian Stelter, Marc Bestmann, Norman Hendrich, and Jianwei Zhang. Fast and reliable stand-up motions for humanoid robots using spline interpolation and parameter optimization. In *20th International Conference on Advanced Robotics, ICAR 2021, Ljubljana, Slovenia, December 6-10, 2021*, pages 253–260. IEEE, 2021. 3

[68] Zhi Su, Xiaoyu Huang, Daniel Felipe Ordoñez Apraez, Yunfei Li, Zhongyu Li, Qiayuan Liao, Giulio Turrisi, Massimiliano Pontil, Claudio Semini, Yi Wu, and Koushil Sreenath. Leveraging symmetry in rl-based legged locomotion control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2024, Abu Dhabi, United Arab Emirates, October 14-18, 2024*, pages 6899–6906. IEEE, 2024. 4

[69] Jie Tan, Zhaoming Xie, Byron Boots, and C Karen Liu. Simulation-based design of dynamic controllers for humanoid balancing. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2729–2736. IEEE, 2016. 3

[70] Tianxin Tao, Matthew Wilson, Ruiyu Gou, and Michiel van de Panne. Learning to get up. In Munkhtsetseg Nandigjav, Niloy J. Mitra, and Aaron Hertzmann, editors, *SIGGRAPH '22: Special Interest Group on Computer Graphics and Interactive Techniques Conference, Vancouver, BC, Canada, August 7 - 11, 2022*, pages 47:1–47:10. ACM, 2022. 4, 5, 6, 14, 15

[71] Yuval Tassa, Tom Erez, and Emanuel Todorov. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4906–4913, 2012. 4

[72] Chen Tessler, Yunrong Guo, Ofir Nabati, Gal Chechik, and Xue Bin Peng. Maskedmimic: Unified physics-based character control through masked motion inpainting. *ACM Trans. Graph.*, 43(6):209:1–209:21, 2024. 3

[73] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30, 2017. 5

[74] Nikolaos G Tsagarakis, Darwin G Caldwell, Francesca Negrello, Wooseok Choi, Lorenzo Baccelliere, Vo-Gia Loc, J Noorden, Luca Muratore, Alessio Margan, Alberto Cardellino, et al. Walk-man: A high-performance humanoid platform for realistic environments. *Journal of Field Robotics*, 34(7):1225–1259, 2017. 2

[75] Unitree. Unitree G1: Humanoid Agent AI Avatar. 2024. URL https://www.unitree.com/g1. 1, 3, 5

[76] Miomir Vukobratović and Branislav Borovac. Zero-moment point—thirty five years of its life. *International journal of humanoid robotics*, 1(01):157–173, 2004. 2

[77] Yikai Wang, Mengdi Xu, Guanya Shi, and Ding Zhao. Guardians as you fall: Active mode transition for safe falling. In *2024 IEEE International Automated Vehicle Validation Conference (IAVVC)*, pages 1–8. IEEE, 2024. 3

[78] Zhen Wu, Jiaman Li, and C Karen Liu. Human-object interaction from human-level instructions. *arXiv preprint arXiv:2406.17840*, 2024. 3

[79] Haoru Xue, Chaoyi Pan, Zeji Yi, Guannan Qu, and Guanya Shi. Full-order sampling-based mpc for torque-level locomotion control via diffusion-style annealing. *arXiv preprint arXiv:2409.15610*, 2024. 2

[80] Kevin Zakka, Baruch Tabanpour, Qiayuan Liao, Mustafa Haiderbhai, Samuel Holt, Jing Yuan Luo, Arthur Allshire, Erik Frey, Koushil Sreenath, Lueder A Kahrs, et al. Mujoco playground. *arXiv preprint arXiv:2502.08844*, 2025. 9

[81] Chong Zhang, Wenli Xiao, Tairan He, and Guanya Shi. Wococo: Learning whole-body humanoid control with sequential contacts. In *8th Annual Conference on Robot Learning*, 2024. 2

[82] Ziwen Zhuang and Hang Zhao. Embrace collisions: Humanoid shadowing for deployable contact-agnostics motions. *arXiv preprint arXiv:2502.01465*, 2025. 3

[83] Ziwen Zhuang, Shenzhe Yao, and Hang Zhao. Humanoid parkour learning. *arXiv preprint arXiv:2406.10759*, 2024. 2, 15

APPENDIX

TABLE II: **Reward components and weights in Stage I.** Penalty rewards prevent undesired behaviors for sim-to-real transfer, regularization refines motion, and task rewards ensure successful getting up or rolling over.

TERM	EXPRESSION	WEIGHT
Penalty:		
Torque limits	$\mathbb{1}(\boldsymbol{\tau}_t \notin [\boldsymbol{\tau}_{\min}, \boldsymbol{\tau}_{\max}])$	-0.1
DoF position limits	$\mathbb{1}(\boldsymbol{d}_t \notin [\boldsymbol{q}_{\min}, \boldsymbol{q}_{\max}])$	-5
Energy	$\| \boldsymbol{\tau} \odot \dot{\mathbf{q}} \|$	-1e-4
Termination	$\mathbb{1}_{\text{termination}}$	-500
Regularization:		
DoF acceleration	$\|\ddot{\boldsymbol{d}}_t\|_2$	-1e-7
DoF velocity	$\|\dot{\boldsymbol{d}}_t\|_2^2$	-1e-4
Action rate	$\|\boldsymbol{a}_t\|_2^2$	-0.1
Torque	$\|\boldsymbol{\tau}_t\|$	-6e-7
DoF position error	$\mathbb{1}(h_{\text{base}} \geq 0.8) \cdot \exp\left(-0.05 \|\boldsymbol{d}_t - \boldsymbol{d}_t^{\text{default}}\|\right)$	-0.75
Angular velocity	$\|\omega^2\|$	-0.1
Base velocity	$\|\boldsymbol{v}^2\|$	-0.1
Foot slip	$\mathbb{1}(\boldsymbol{F}_z^{\text{feet}} > 5.0) \cdot \sqrt{\|\boldsymbol{v}_z^{\text{feet}}\|}$	-1
Feet distance reward	$\frac{1}{2}\Big(\exp(-100\,\lvert\max(\boldsymbol{d}_{\text{feet}} - \boldsymbol{d}_{\min}, -0.5)\rvert) + \exp(-100\,\lvert\max(\boldsymbol{d}_{\text{feet}} - \boldsymbol{d}_{\max}, 0)\rvert)\Big)$	2
Feet orientation	$\sqrt{\|\mathbf{g}_{xy}^{\text{feet}}\|}$	-0.5
Feet height reward	$\exp(-10 \cdot \boldsymbol{h}^{\text{feet}})$	2.5
Getting-Up Task Rewards:		
Base height exp	$\exp(\boldsymbol{h}^{base}) - 1$	5
Head height exp	$\exp(\boldsymbol{h}^{head}) - 1$	5
Δ base height	$\mathbb{1}(\boldsymbol{h}_t^{base} > \boldsymbol{h}_{t-1}^{base})$	1
Feet contact forces reward	$\mathbb{1}(\|\boldsymbol{F}_t^{feet}\| > \|\boldsymbol{F}_{t-1}^{\text{feet}}\|)$	1
Standing on feet reward	$\mathbb{1}\big((\|\boldsymbol{F}^{\text{feet}}\| > 0) \& (\boldsymbol{h}^{\text{feet}} < 0.2)\big)$	2.5
Body upright reward	$\exp(-\mathbf{g}_z^{\text{base}})$	0.25
Soft body symmetry penalty	$\|\mathbf{a}_{\text{left}} - \mathbf{a}_{\text{right}}\|$	-1.0
Soft waist symmetry penalty	$\|\mathbf{a}^{\text{waist}}\|$	-1.0
Rolling-Over Task Rewards:		
Base Gravity Error	$1 - \cos\theta_{\text{base}}$, $\cos\theta_{\text{base}} = \frac{\mathbf{g}^{\text{base}} \cdot \mathbf{g}^{\text{target}}}{\|\mathbf{g}^{\text{base}}\| \|\mathbf{g}^{\text{target}}\|}$	-2
Torso Gravity Error	$1 - \cos\theta_{\text{torso}}$, $\cos\theta_{\text{torso}} = \frac{\mathbf{g}^{\text{torso}} \cdot \mathbf{g}^{\text{target}}}{\|\mathbf{g}^{\text{torso}}\| \|\mathbf{g}^{\text{target}}\|}$	-2
Knee Gravity Error	$\frac{1}{2}\Big((1 - \cos\theta_{\text{left}}^{\text{knee}}) + (1 - \cos\theta_{\text{right}}^{\text{knee}})\Big)$, $\cos\theta^{\text{knee}} = \frac{\mathbf{g}^{\text{knee}} \cdot \mathbf{g}^{\text{target}}}{\|\mathbf{g}^{\text{knee}}\| \|\mathbf{g}^{\text{target}}\|}$	-2

A REWARDS

A.1 Rewards Components in Stage I

Detailed reward components used in Stage I are summarized in Tab. II.

A.2 Rewards Components in Stage II

Detailed reward components used in Stage II are summarized in Tab. III.

TABLE III: **Reward components and weights in Stage II.** Penalty rewards prevent undesired behaviors for sim-to-real transfer, regularization refines motion, and task rewards ensure successful whole-body tracking in real time.

TERM	EXPRESSION	WEIGHT
Penalty:		
Torque limits	$\mathbb{1}(\boldsymbol{\tau}_t \notin [\boldsymbol{\tau}_{\min}, \boldsymbol{\tau}_{\max}])$	-5
Ankle torque limits	$\mathbb{1}(\boldsymbol{\tau}_t{}^{\text{ankle}} \notin [\boldsymbol{\tau}_{\min}^{\text{ankle}}, \boldsymbol{\tau}_{\max}^{\text{ankle}}])$	-0.01
Upper torque limits	$\mathbb{1}(\boldsymbol{\tau}_t{}^{\text{upper}} \notin [\boldsymbol{\tau}_{\min}^{\text{upper}}, \boldsymbol{\tau}_{\max}^{\text{upper}}])$	-0.01
DoF position limits	$\mathbb{1}(\boldsymbol{d}_t \notin [\boldsymbol{q}_{\min}, \boldsymbol{q}_{\max}])$	-5
Ankle DoF position limits	$\mathbb{1}(\boldsymbol{d}_t{}^{\text{ankle}} \notin [\boldsymbol{q}_{\min}^{\text{ankle}}, \boldsymbol{q}_{\max}^{\text{ankle}}])$	-5
Upper DoF position limits	$\mathbb{1}(\boldsymbol{d}_t{}^{\text{upper}} \notin [\boldsymbol{q}_{\min}^{\text{upper}}, \boldsymbol{q}_{\max}^{\text{upper}}])$	-5
Energy	$\| \boldsymbol{\tau} \odot \dot{\mathbf{q}} \|$	-1e-4
Termination	$\mathbb{1}_{\text{termination}}$	-50
Regularization:		
DoF acceleration	$\|\ddot{\boldsymbol{d}}_t\|_2$	-1e-7
DoF velocity	$\|\dot{\boldsymbol{d}}_t\|_2^2$	-1e-3
Action rate	$\|\boldsymbol{a}_t\|_2^2$	-0.1
Torque	$\|\boldsymbol{\tau}_t\|$	-0.003
Ankle torque	$\|\boldsymbol{\tau}_t{}^{\text{ankle}}\|$	-6e-7
Upper torque	$\|\boldsymbol{\tau}_t{}^{\text{upper}}\|$	-6e-7
Angular velocity	$\|\omega^2\|$	-0.1
Base velocity	$\|\boldsymbol{v}^2\|$	-0.1
Feet distance reward	$\frac{1}{2}\Big(\exp(-100\,\lvert\max(\boldsymbol{d}_{\text{feet}} - \boldsymbol{d}_{\min}, -0.5)\rvert) + \exp(-100\,\lvert\max(\boldsymbol{d}_{\text{feet}} - \boldsymbol{d}_{\max}, 0)\rvert)\Big)$	2
Foot orientation	$\sqrt{\|\mathbf{g}_{xy}^{\text{feet}}\|}$	-0.5
Tracking Rewards:		
Tracking DoF position	$\exp\left(-\frac{(\boldsymbol{d}_t - \boldsymbol{d}_t^{\text{target}})^2}{4}\right)$	8

B ADDITIONAL RESULTS

B.1 Additional Baseline Result Visualization

Fig. 7 showcases a visualization of getting up from a prone pose generated by the baseline method [70]. This method generates motion that is highly unstable and unsafe to deploy in the real world. For example, its joints continuously jitter, the feet are stumbling and the body keeps jumping up. This indicates that this baseline [70] cannot be Sim2Real.

B.2 Additional Results on Getting Up From Sitting On Stairs & Leaning Against Walls

Fig. 8(a) and (b) show the simulation results of getting up from additional initial postures: sitting on the stairs and leaning against the wall. The results show that HUMANUP can be generalized to more diverse initial postures in addition to lying on the ground. Besides, we find that the convergence is $\sim 4\times$ faster than getting up from lying. We argue that getting

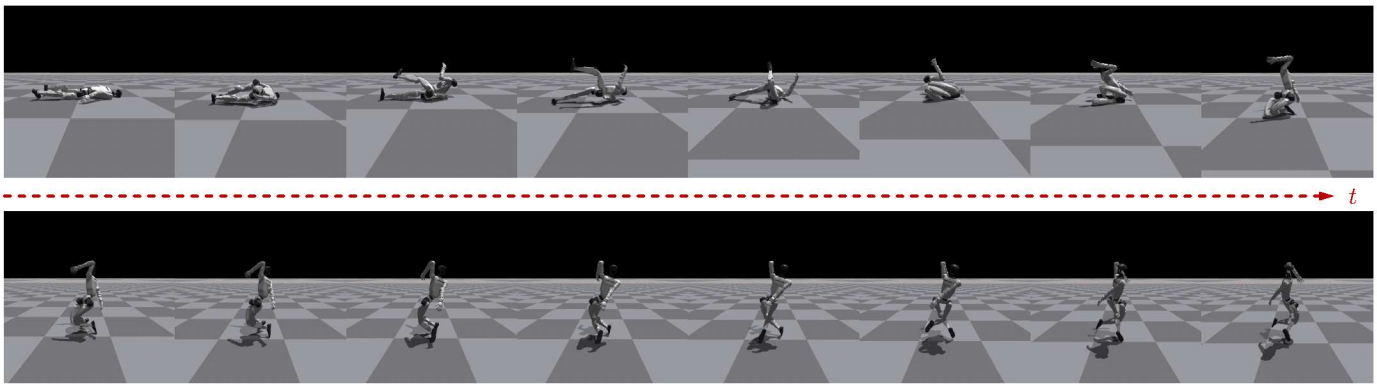


Fig. 7: **Getting-up from prone pose result visualization of Tao et al. [70].** The motion generated by method [70] is highly unstable and unsafe, and it keeps jittering and jumping during the getting-up phase.

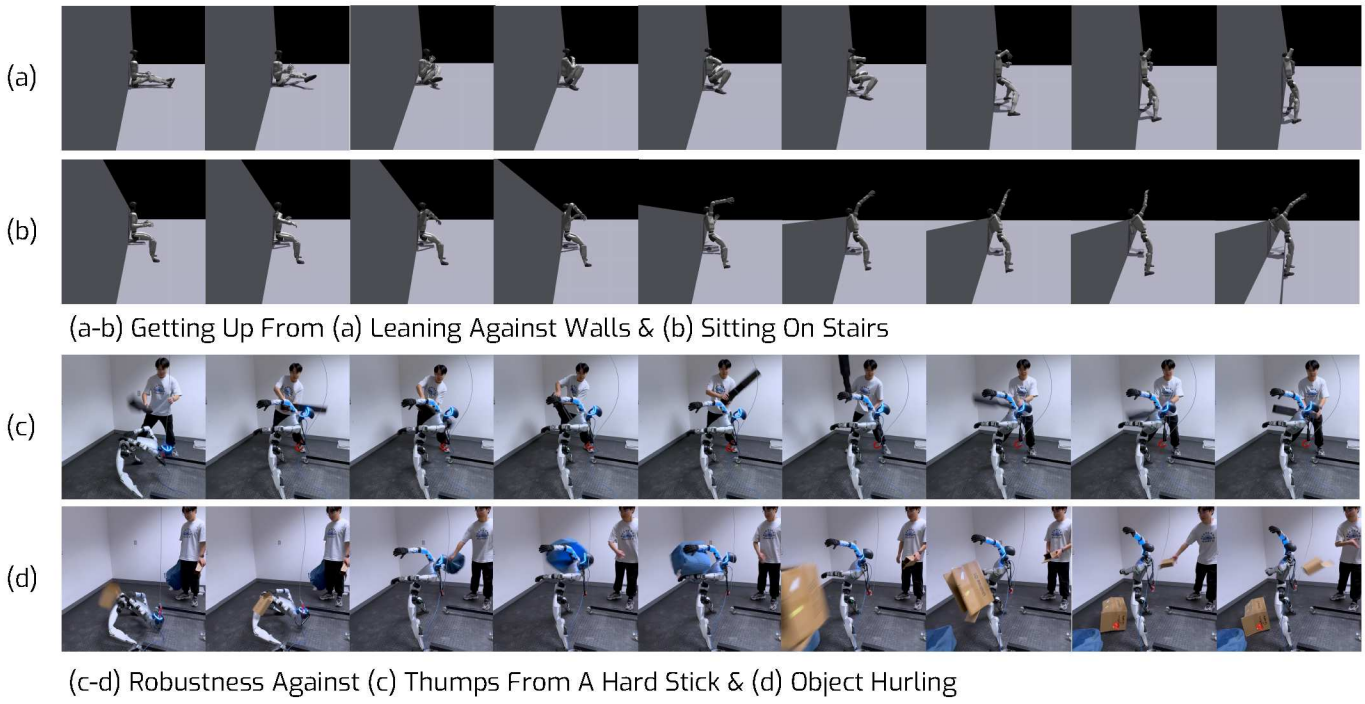


Fig. 8: **Getting up from additional initial poses and against real-world external turbulence.** Getting up from (a) leaning against walls, (b) sitting on the stairs. Robustness to external turbulence: (c) thumps from a hard stick, and (d) object hurling. HUMANUP enables the robot to get up from initial poses other than lying, and is robust against external turbulence.

up from sitting or leaning is easier because of the additional support from the ground, chairs, or walls.

B.3 Additional Robustness Test Against External Turbulence

Fig. 8(c) and (d) show the robustness test against external turbulence: thumps from a hard stick and object hurling. The results show that HUMANUP getting-up policy is practically robust against certain external turbulence in the real world.

C TRAINING DETAILS

In Stage I, we train the discovery policy f for overall 5B simulation steps, and 20K simulation steps for the Stage II deployable policy π. Each stage uses a regularization curriculum within, an implementation detail common to policy learning in legged locomotion literature. All training is conducted on Isaac Gym [56], and we train our policies using 4,096 paralleled environments on a single NVIDIA RTX 4090 or L40S GPU. For the getting-up task, we slow down the discovered trajectory to 8 seconds ($8\times$). We also tried $4\times$ and $10\times$. $4\times$ leads to large torques and DoF velocities, and $10\times$ does not converge. For the rolling-over task, the trajectory is slowed down to 4 seconds (selected through trials similar to the getting-up task). We use flat terrains in Stage I and varied terrains during Stage II, involving flat, rough, and slope terrains. We follow previous works [10, 32, 83] to apply varied dynamics randomization, such as base center of mass (CoM) offset and control delay.