Expressive Whole-Body Control for Humanoid Robots

Xuxin Cheng*, Yandong Ji*, Junming Chen, Ruihan Yang, Ge Yang, Xiaolong Wang
UC San Diego
https://expressive-humanoid.github.io

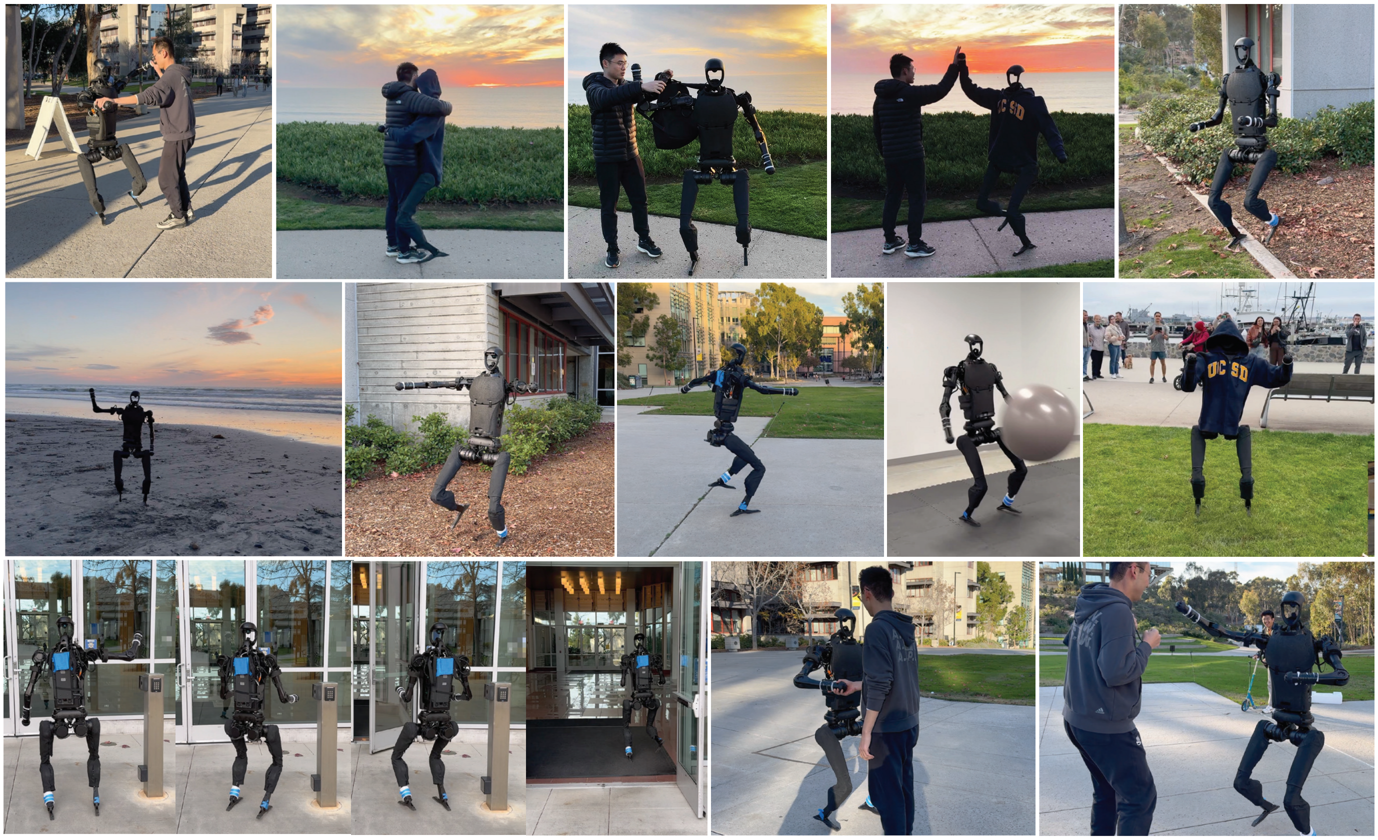


Fig. 1: Our Robot demonstrates diverse and expressive whole-body movements in different scenarios. Top Row: The robot is dancing, hugging and slapping hands with a human. Middle Row: The robot is able to walk on different terrains including gravel and wood chip paths, inclined concrete paths, grass, and curbsides with various expressions like zombie walk, exaggerated stride or waving. Bottom Left: The robot is able to use a waving gesture to open a wave-sensing door. Bottom Right: The robot is shaking hands and provoking.

***Abstract*—Can we enable humanoid robots to generate rich, diverse, and expressive motions in the real world? We propose to learn a whole-body control policy on a human-sized robot to mimic human motions as realistic as possible. To train such a policy, we leverage the large-scale human motion capture data from the graphics community in a Reinforcement Learning framework. However, directly performing imitation learning with the motion capture dataset would not work on the real humanoid robot, given the large gap in degrees of freedom and physical capabilities. Our method Expressive Whole-Body Control (ExBody) tackles this problem by encouraging the upper humanoid body to imitate a reference motion, while relaxing the imitation constraint on its two legs and only requiring them to follow a given velocity robustly. With training in simulation and Sim2Real transfer, our policy can control a humanoid robot to walk in different styles, shake hands with humans, and even dance with a human in the real world. We conduct extensive studies and comparisons on diverse motions in both simulation and the real world to show the effectiveness of our approach.**

I. Introduction

When we think of robots, we often begin by considering what kinds of tasks they can accomplish for us. Roboticists typically work under this framework, and formulate control as optimizing for a specific cost function or task objective. When applied to robots that resemble our house pets such

[*] The first two authors contributed equally.

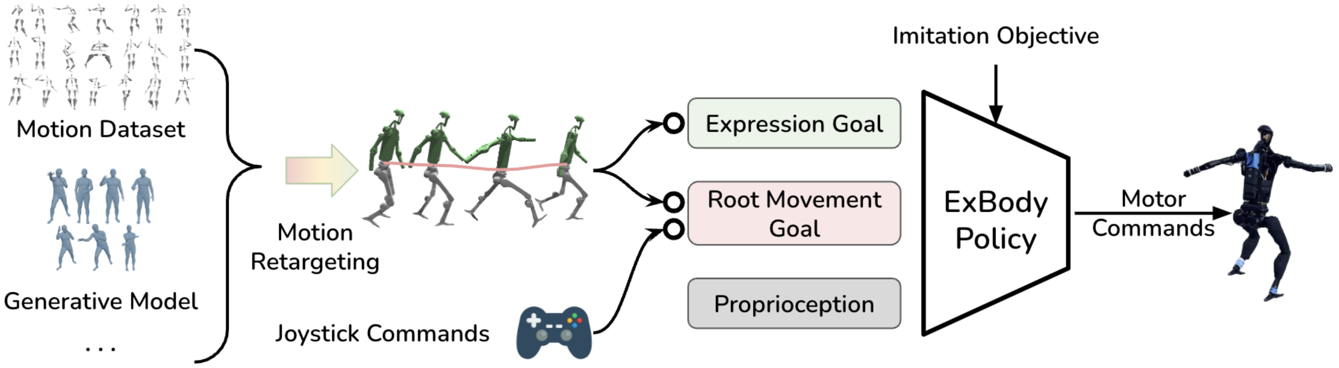


Fig. 2: Overview of our framework. Our framework is able to train on data from various sources such as static human motion datasets, generative models, video to pose models that are widely available. After motion retargeting, we acquire a repertoire of motion clips that are compatible with our robot's kinematic structure. We extract expression goal $\mathbf{g}^e$ and root movement goal $\mathbf{g}^m$ from the rich features from retargeted motion clips as the goal of our goal-conditioned RL objective. The root movement goal $\mathbf{g}^m$ can also be intuitively given by joystick commands, enabling convenient deployment in the real world.

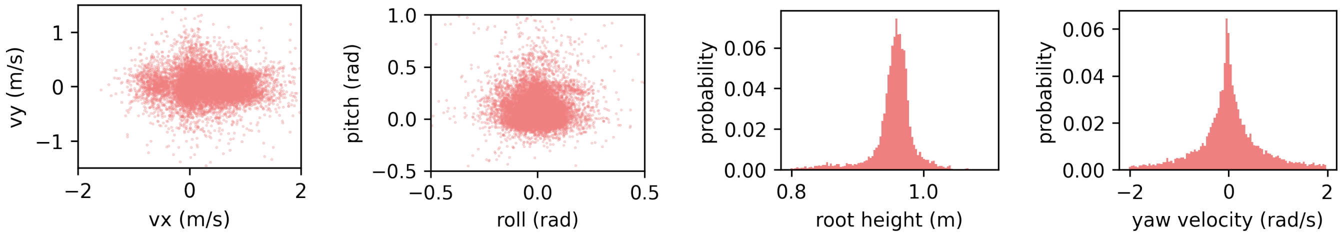
as quadruped robot dogs, or humans, whole-body control methods on both of these two form factors tend to produce singular motion patterns that lack grace and personality — oftentimes a by-product of the additional constraints we have to add to make optimization or learning easier. In contrast, motions from actual humans and animals are rich, diverse, and expressive of their intent or emotional valence. In other words, there exists a large subspace of motion control that is not described by common objectives such as body velocity, heading, and gait patterns. What would it take to build robots that can generate, and perform diverse whole-body motions that are as expressive as humans?

In this paper, we tackle the problem of learning a whole-body motion control policy for a human-sized robot that can match human motions in its expressivity and richness. We do so by combining large-scale human motion capture data from the graphics community with deep Reinforcement Learning (RL) in a simulated environment, to produce a whole-body controller that can be deployed directly on the real robot. We illustrate the expressiveness of our controller in Fig. 1, and show that the robot is sufficiently compliant and robust that it can hold hands and dance with a person.

Our work benefits from prior research from the computer graphics community on physics-based character animation [31], and from the robotics community on using deep reinforcement learning to produce robust locomotion policy on various legged robots [28, 4]. In our study, we found that although physics-based character animation produces natural-looking reactive control policies that look good in a virtual setting, such results often involve large actuator gains in the range of 60kg/m that are one magnitude larger than what is feasible with current hardware. We also found that human reference motion often involves a lot more degrees of freedom (DoF) than the robot hardware. For example, the physics-based animation can use much more DoF (e.g., 69DoF [34]) compared to a real-world robot (e.g., 19DoF on a Unitree H1 robot). These two factors make the direct transfer of graphics techniques onto the real robot infeasible.

Our key idea is to NOT mimic exactly the same as the reference motion. We propose to train a novel controller that takes both a reference motion and a root movement command as inputs for real humanoid robot control. We call our approach **Ex**pressive Whole-**Body** Control (**ExBody**). During training with RL, we encourage the upper body of the humanoid robot to imitate diverse human motions for expressiveness, while relaxing the motion imitation term for its two legs. Concretely, the reward function for the legged locomotion is designed for following the root movement commands robustly provided by the reference motion instead of matching each exact joint angle. We train our policy in highly randomized challenging terrains in simulation. This not only allows robust sim2real transfer but also learns a policy that does not just "repeat" the given motion. The user can command the humanoid robot to move at different speeds, turning in different directions on diverse terrains, and reproduce the reference motion on the upper body at the same time. As shown in Fig. 1, we can command our robot to dance with a human, waving and shaking hands while walking, or walking like a mummy on diverse terrains.

We adopt the Unitree H1 robot in both simulation and real-world experiments. To learn from diverse human motions, we utilize the CMU MoCap dataset (around 780 reference motions). Such richness not only enables more expressive humanoid motion but also more robust walking. Our evaluation shows the upper body motions and diverse moving velocity augment the training data and provide efficient guidance in training. We also compare our method with applying more imitation constraints on legged motion in both simulation and the real world and show our approach that relaxes the constraints indeed leads to better and more robust results. To the best of our knowledge, our work is the first work on learning-based real-world humanoid control with diverse mo-



(a) (b) (c) (d)

Fig. 3: Dataset visualization of our training data from CMU MoCap. We sample all the motion clips at an incremental of 1s. The resulting number of plotting data points are 1338. We can observe the bias of the distribution from human motions. Such distributions are proven to help policy learning in Sec. IV.

tions. While our current results focus on expressive humanoid control, we hope our approach can also shed some light on studying generalizable humanoid whole-body manipulation and navigation.

Metrics	Mimic WBC(Ours)	PHC [34]	ASE [45]
DoFs	19	69	37
Number of Motion Clips	780	11000	187
Total Time of Motions (h)	3.7	40	0.5
Real Robot	✓	×	×
Single Network	✓	×	✓
Linear Velocities Obs	×	✓	✓
Keypoint Positions Obs	×	✓	✓
Robot Height Obs	×	×	✓

TABLE I: Comparisons with physics-based character animation works. In PHC, the policy observes the Linear velocities and keypoint positions of each rigid body, while in ASE linear velocities are for the root only. PHC and ASE both observe privileged states that are not available on the real robot.

II. Problem Formulation

We consider humanoid motion control as learning a goal-conditioned motor policy $\pi : \mathcal{G} \times \mathcal{S} \mapsto \mathcal{A}$, where $\mathcal{G}$ is the goal space that specifies the behavior, $\mathcal{S}$ is the observation space, and $\mathcal{A}$ is the action space that contains the joint positions and torque. We assume in the rest of this paper, without loss of generality, that the observation and action space are given by the H1 humanoid robot design. However, our proposed approach should generalize to similar body forms that differ in the exact number of actuated degrees of freedom.

a) Command-conditioned Locomotion Control: We aim to produce a robust control policy for the Unitree H1 hardware that can be commanded by the linear velocity $\mathbf{v} \in \mathbb{R}^3$, body pose in terms of row/pitch/yaw $rpy \in \mathbb{R}^3$ and the body height h measured at the root link. Formally, the goal space for root movement control $\mathcal{G}^m = \langle \mathbf{v}, rpy, h \rangle$. The observation $\mathcal{S}$ includes the robot's current proprioception information $s_t = [\omega_t, r_t, p_t, \Delta y, q_t, \dot{q}_t, \mathbf{a}_{t-1}]^T$. ω_t is the robot root's angular velocity, r_t, p_t is roll and pitch. Note that the policy does not observe the current velocity $\mathbf{v}$, and the absolute body height h and the current yaw angle y_t because these are privileged information for the real robot (see Tab. I). We let the policy observe the difference between current and desired yaw angle $\Delta y = y_t - y$ to convert the global quantity to a local frame that can be intuitively commanded at deployment time. The actions $\mathbf{a}_t \in \mathbb{R}^{19}$ is the target position of joint-level proportional-derivative (PD) controllers. The PD controllers compute the torque for each motor with the specified PD gains k_p^i and damping coefficient k_d^i.

*b) **Expressive** Whole-**Body** Control:* We extend the command-conditioned locomotion control to include descriptions of the robot's movement that are not captured by root pose and velocity in $\mathcal{G}^m$. We formulate this as the more general goal space $\mathcal{G} = \mathcal{G}^e \times \mathcal{G}^m$, where the expression target $\mathbf{g}^e \sim \mathcal{G}^e$ includes the desired joint angles and various 3D keypoint locations of the body.

Specifically, in this work, we work with a relaxed problem where we exclude the joints and key points from the lower half of the body from $\mathcal{G}^e$. This is because the robot has a different body plan from humans, and including these low-body features from human motion capture data tends to over-constrain the problem and lead to brittle, and poorly performing control policies. Formally, for the rest of this paper, we work with $\mathcal{G}^e = \langle \mathbf{q}, \mathbf{p} \rangle$, where $\mathbf{q} \in \mathbb{R}^9$ are the joint positions of the nine actuators of the upper body, and $\mathbf{p} \in \mathbb{R}^{18}$ are the 3D key points of the two shoulders, two elbows, and the left and right hands. The goal of expressive whole-body control (ExBody) is to simultaneously track both the root movement goal (for the whole body) $\mathbf{g}^m \sim \mathcal{G}^m$, as well as the target expression goal (for upper body) $\mathbf{g}^e \sim \mathcal{G}^e$.

III. Expressive Whole-Body Control

We present Expressive Whole-Body Control (ExBody), our approach for achieving expressive and robust motion control on a humanoid robot as shown in Fig. 2. In the following sections, we cover the key components of this approach, including strategies for curating and retargeting human motion capture data to humanoid robot hardware, and using some of these prior knowledge to improve the RL training procedure.

A. Strategies for Curating Human Behavior Data

In our research, we selectively used a portion of the CMU MoCap dataset, excluding motions involving physical interactions with others, heavy objects, or rough terrain. This

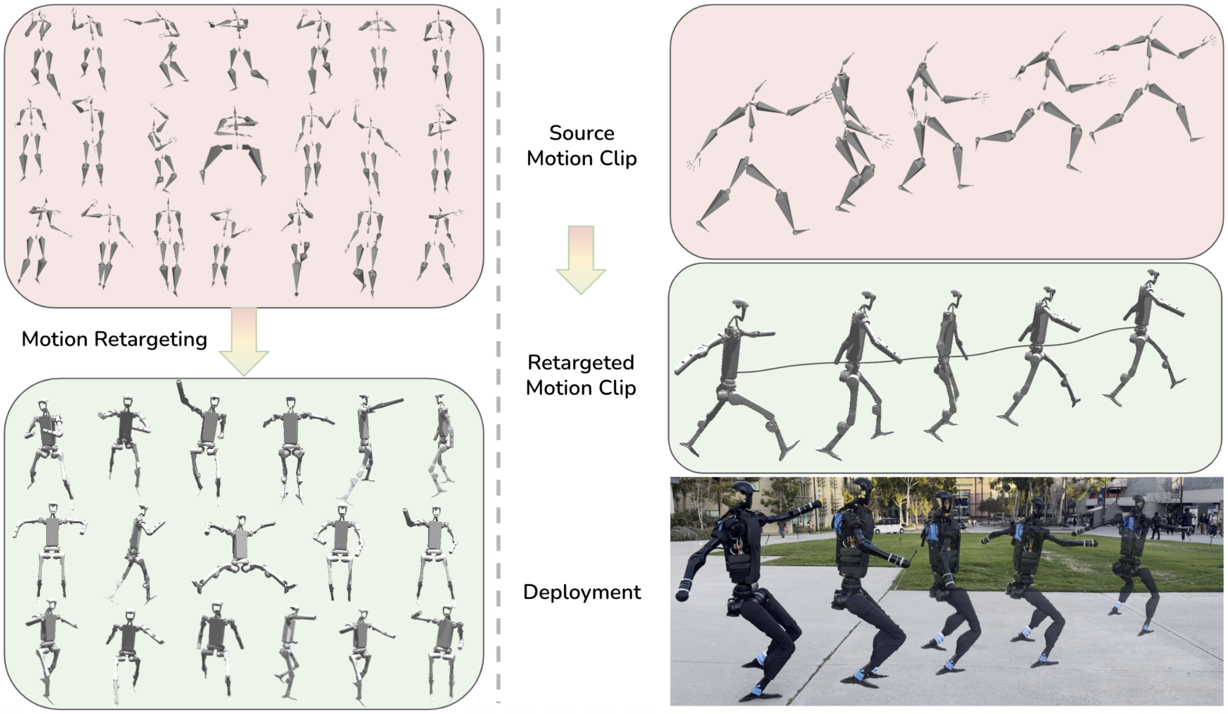


Fig. 4: Left: During training, we extract a large repertoire of retargeted motion clips and train our ExBody policy. Right: During deployment, we can replay motion that can come from a variety of sources such as static motion datasets, diffusion models, or video-to-skeleton models. For Unitree H1, the robot we use, the shoulder and hip joints have three perpendicular DoFs. Other joints are 1 DoF each. There are 19 DoFs in total. We also notice that some of the retargeted motions exhibit exaggerated movement with robot's lower body, which is why we use ExBody to make it transferrable.

was done semi-automatically, as our framework cannot realistically implement motions with significant environmental interactions. The resulting motions are in Tab. II.

	Category	Clips	Length (s)
Training	Walk	546	9076.6
	Dance	78	1552.3
	Basketball	36	766.1
	Punch	20	800.0
	Others	100	1188.0
	Total	780	13383.0
Real-World Test	Punch	1	18.9
	Wave Hello	1	5.0
	Mummy Walk	1	22.5
	Zombie Walk	1	13.0
	Walk, Exaggerated Stride	1	2.5
	High Five	1	3.3
	Basketball Signals	1	32.6
	Adjust Hair	1	9.6
	Drinking from Bottle	1	15.2
	Direct Traffic	1	39.3
	Hand Signal	1	32.2
	Russian Dance	1	8.2
	Total	11	202.3
Additional Realworld Test (Diffusion [54])	Boxing	1	4.0
	Hug	1	4.0
	Shake Hands	1	4.0

TABLE II: The details of our dataset. We select a subset from CMU MoCap dataset for training, and test on various expressive motions in sim and the real world.

We plot the distribution of root movement goal $\mathbf{g}^e$ in Fig. 3. The yaw angle in the world frame does not have too much meaning because during training all our observations are in the robot's local frame. So we visualize the yaw angular velocity instead. From Fig. 3a, we can observe that the motions we choose cover walking forward, backward, and sideways, and are biased towards walking forward and symmetrical on walking sideways. From Fig. 3b, we observe a chestnut-shaped distribution with minimal movement along the roll angle, because humans do not usually incline the body sideways. However, in terms of pitch, it is biased toward bending forward instead of bending backward. And as the pitch angle gets larger the roll angle gets smaller in distribution, which resonates with human bias. From Fig. 3c, we observe that the robot's height is narrowly centered around its nominal height and has small variations. From Fig. 3d we can see that our turning motion is pretty balanced and there are turning left and right motions. Unlike [13], which randomly samples points using a spherical coordinate frame, and then checks if the points are under the ground or have a collision with the robot itself, our approach from large human data naturally has samples that generally do not violate such constraints. Even if there are some collisions with the robot itself after retargeting, the RL will avoid it via collision penalization. We show in Sec. IV that this prior distribution actually helps with policy learning a lot compared with manually designed sample space.

B. Motion Retargeting to Hardware

In consideration of the distinct morphological differences between the H1 robot and humans, we adapt the human motion data to the robot's framework by straightforwardly mapping local joint rotations onto the robot's skeleton. We use the Unitree H1 robot[3] as our platform with a total mass of around $51.5kg$ and a height of around $1.8m$. The Unitree H1 robot has 19 DoFs. The shoulder and hip joints have 3 revolute joint motors connected perpendicularly, thus equivalent to a spherical joint usually used in human motion datasets [35, 18]. During retargeting, we consider the 3 hip or shoulder joints as 1 spherical joint. After retargeting, we remap the spherical joint which is represented by a normalized quaternion $\mathbf{q}^{\mathbf{i}}_{\mathbf{m}} = (q_x, q_y, q_z, q_w)$ to the original joint angle of 3 revolute joints $\mathbf{m} = [m_1, m_2, m_3] \in \mathbb{R}^3$ by exponential mapping. To achieve this we first convert the quaternion $\mathbf{q}$ to the form of axis angle:

$$\theta = 2\arccos(q_w), \ \ \mathbf{a} = \frac{1}{\sqrt{1-q_w^2}} \begin{pmatrix} q_x \\ q_y \\ q_z \end{pmatrix}$$

where $\mathbf{a}$ is the rotation axis and θ is the rotation angle. For small angles, the last axis is used if $\sqrt{1-q_w^2}$ is close to zero. Then the mapped angle is just simply $\mathbf{m} = \theta\mathbf{a}$, where $\mathbf{m} = [q^i, q^j, q^k]$ is 3 corresponding DoFs in $\mathbf{q}$. For 1D joints, namely the elbow, torso, knee, and ankle, we take the rotation angle projected onto the corresponding rotation axis of the 1D joints. In Fig. 4, we show the diverse motions both for the original dataset and the retargeted ones. We can see that although the retargeted data loses some DoFs with hardware constraints, it is still able to keep the important expressions from the original data.

C. Guiding State Initialization from Human Mocap Data

We use massively parallel simulation to train our RL policy with Isaac Gym [36, 47]. We randomly sample an initial state $\mathbf{g} = [\mathbf{g}^e, \mathbf{g}^m]$ from the motion dataset for each environment in simulation and set its state to the sampled state during initialization or resetting. We show by extensive experiments how this random initialization helps with policy learning.

With the diverse goal state $\mathbf{g}$ distribution as shown in Fig. 3 and the corresponding tracking rewards, ExBody is able to produce diverse root movement and diverse arm expressions while still maintaining the balance via the lower body without mimicking rewards. Our policy can make the robot walk forward/backward, sideways, turn yaw, vary root height, adjust roll, pitch, etc.

D. Rewards

In each step, the reward from the environment consists of expression goal, root movement goal tracking, and regularization terms derived from [47]. Imitation rewards are detailed in Tab. III, where $\mathbf{q}_{\text{ref}} \in \mathbb{R}^9$ is reference position of the upper body joints, $\mathbf{p}_{\text{ref}} \in \mathbb{R}^{18}$ is reference position of the upper body keypoints, $\mathbf{v}_{\text{ref}}$ is reference body velocity, $\mathbf{\Omega}^{\phi\theta}_{\text{ref}}$ and $\mathbf{\Omega}^{\phi\theta}$ are reference and actual body roll and pitch. Refer to supplementary for regularization rewards.

Term	Expression	Weight
Expression Goal G^e		
DoF Position	$\exp(-0.7\lvert\mathbf{q}_{\text{ref}} - \mathbf{q}\rvert)$	3.0
Keypoint Position	$\exp(-\lvert\mathbf{p}_{\text{ref}} - \mathbf{p}\rvert$	2.0
Root Movement Goal G^m		
Linear Velocity	$\exp(-4.0\lvert\mathbf{v}_{\text{ref}} - \mathbf{v}\rvert)$	6.0
Roll & Pitch	$\exp(-\lvert\mathbf{\Omega}^{\phi\theta}_{\text{ref}} - \mathbf{\Omega}^{\phi\theta}\rvert)$	1.0
Yaw	$\exp(-\lvert\Delta y\rvert)$	1.0

TABLE III: Expressive Rewards Specification

IV. Results

In this section we aim to answer the following questions through extensive experiments both in sim and the real world:

- How well does ExBody perform on tracking $\mathbf{g}^e$ and $\mathbf{g}^m$?
- How does learning from large datasets help policy exploration and robustness?
- Why do the state of the art approaches in computer graphics for physics based character control not work well in the real robot case and why do we need ExBody?

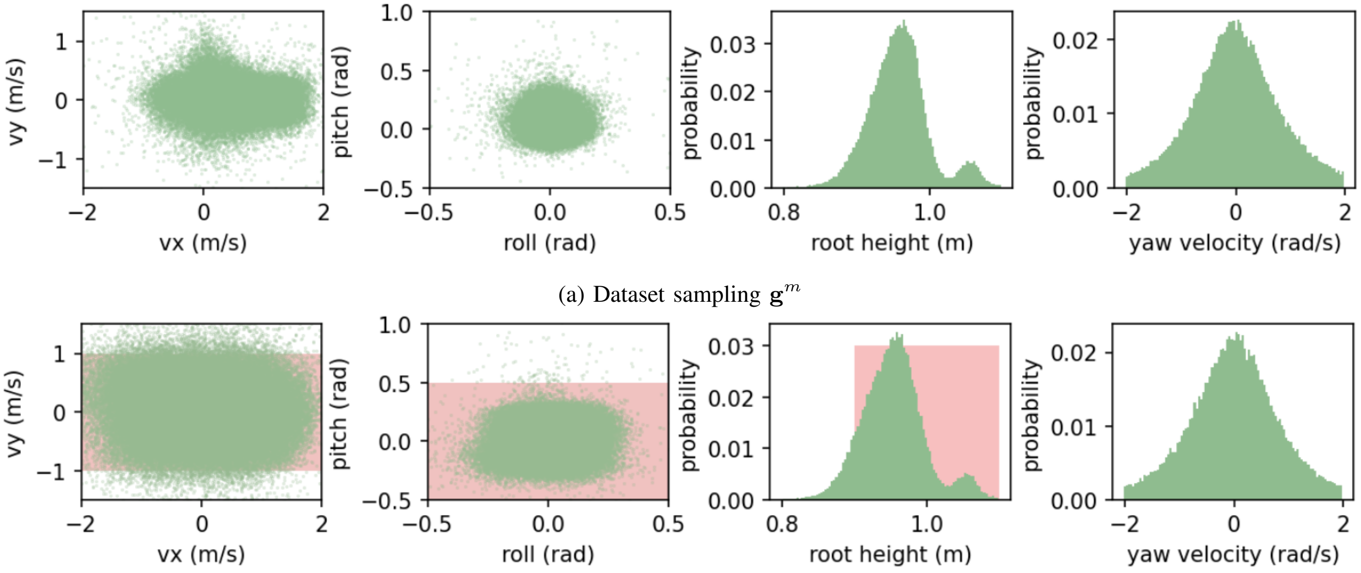
Our baselines are as follows:

- **ExBody + AMP**: This baseline uses an AMP reward to encourage the policy's transitions to be similar to those in the retargeted dataset.
- **ExBody +AMP NoReg**: We remove the regularization terms in our reward formulations and see if AMP reward itself is able to handle the regularization of the imitation learning problem with such a large dataset.
- **Random Sample**: Randomly sample root movement goals $\mathbf{g}^m$ with the range shown in Fig. 5b.
- **No RSI**: Initialize the environment with default DoF positions and root states instead of sampling from the motion dataset.
- **Full body tracking**: Instead of tracking only the upper body with $\mathcal{G}^e$, the objective is to track the joint angles and 3D key points for the entire body including hips, knees, and ankles.

Our metrics are as follows:

- **Mean Episode Linear Velocity Tracking Reward** (MELV)
- **Mean episode roll pitch tracking reward** (MERP)
- **Mean episode lengths** (MEL)
- **Mean episode key body tracking reward** (MEK)

How well does ExBody perform on tracking $\mathbf{g}^m$? In Fig. 5a, we show that our trained ExBody can produce similar root movement state distribution as shown in the training dataset in Fig. 3. The other way to put it, the policy successfully learned to map a few intuitive root movement commands to appropriate motor commands, and executing these motor commands in a physical world will result in tracking the root movement commands. However, our ExBody is trained on motion clips, which have temporal correlations in terms of the goal state $\mathbf{g}$. That is to say, the robot is commanded



(a) Dataset sampling $\mathbf{g}^m$

(b) Random sampling $\mathbf{g}^m$

Fig. 5: Policy's state distribution under different sampling strategies. The green dots are the policy rollout's states. For dataset sampling, we record 20 data points for 4096 environments with randomly sampled arm trajectories from our training set. For random sampling, he red shade represents the randomly sampled $\mathbf{g}^m$ range. For yaw velocity, we do not sample the command, because the policy observes the difference between the desired and actual yaw, and does not explicitly track the angular velocity. The second peak in root height is the initialization bias.

to replay a certain trajectory that has meaning. What about the randomly sampled commands? If the policy can handle these commands, we are able to intuitively control the robot in the real world with a joystick, without worrying about finding the correct motion clips and replay them. We show that our trained ExBody policy can generalize to any input trajectories other than those in the dataset. To do this, we randomly sample the root state commands and record the policy's rollout distribution. We can see from Fig. 5, that the policy covers most areas of the sampled velocity commands, which is larger than the training set in Fig. 3. For roll and pitch commands, the policy's state distribution is smaller than the sampled commands. We speculate this is due to the robot's self-property (upper body too heavy, etc). The reason why we do not give a number for tracking error is that without comparison, it has no meaning, so qualitatively seeing the data distribution is more straightforward and intuitive.

How well does ExBody perform on tracking $\mathbf{g}^e$? Similarly, we render the samples of end-effectors (hands) positions relative to the robot to show a nearly identical distribution of reference motion and learned policy as shown in Fig. 6.

Why should we learn from large data? We compare with baselines to show that our approach ExBody is superior compared with other design choices. Traditionally, RL-based robust locomotion for legged robots is trained either through reward engineering or from a limited set of reference motions. In our work, we show the advantage of learning robust Whole-Body control for humanoid robots from large motion datasets. As shown in Tab. IV, our method achieves the best linear velocity tracking performance (MELV). The benefit largely comes from RSI, where we initialize the robot to different states that encourage exploration. No RSI is not able to discover proper positive reward states before learning to suicide as soon as possible to avoid negative rewards in simulation. The Random Sample baseline's behavior is a kneel-down motion for all the goals as shown in Fig. 8, taking advantage of the environment. It gives up $\mathbf{g}^m$ completely and focuses on $\mathbf{g}^e$. It has a similar MEL score with the motion sample and a higher MEL score with the Random Sample (the training distribution), meaning that kneeling on the ground is more robust. We speculate that the motion dataset offers a more advantageous distribution of $\mathcal{G}^m$, which in turn facilitates the policy learning process. For example, many motions started from standing in place and gradually started to walking, creating a natural curriculum for the policy to learn.

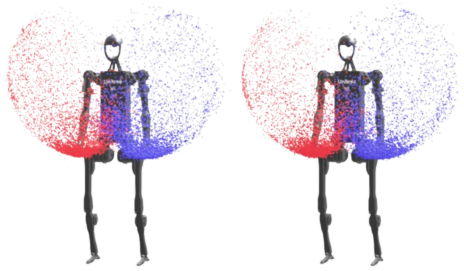


Fig. 6: We sample 10,000 points of hand positions relative to the robot. Left: retargeted motion dataset. Right: learned ExBody policy rollouts. The upper body movement from the dataset forms a natural distribution for learning.

Why does not ExBody do full DoF tracking? Due to the

Baselines	Motion Sample				Random Sample			
	MEL↑	MELV↑	MERP↑	MEK↑	MEL	MELV	MERP	MEK
ExBody (Ours)	16.87	**318.67**	754.92	**659.78**	13.51	**132.14**	523.79	483.67
ExBody + AMP	**17.28**	205.60	**765.85**	635.51	15.59	95.11	583.82	**544.59**
ExBody + AMP NoReg	16.16	87.83	714.74	561.56	15.40	36.76	584.23	515.53
No RSI	0.23	0.63	10.09	7.25	0.22	0.10	7.41	7.15
Random Sample	16.50	181.85	704.73	326.66	16.37	38.51	**586.83**	324.10
Full Body Tracking	13.28	246.11	584.40	397.25	10.76	76.46	407.88	284.69

TABLE IV: Comparisons with baselines. We sample 10,000 trajectories with 4096 environments in simulation and report their mean episode metrics. Motion Sample means we sample $\mathbf{g}^m$ from retargeted motions. Random Sample means we uniformly sample $\mathbf{g}^m$ in Fig. 3.

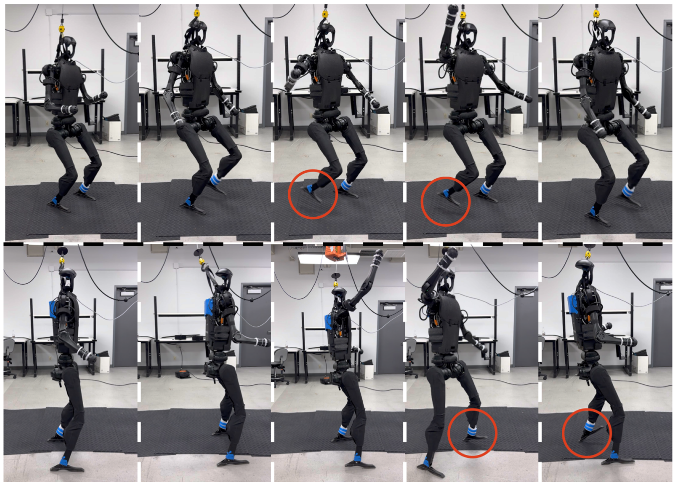


Fig. 7: H1 robot doing a High Five in the real world. Top Row: ExBody only (Ours) walks with more bent knees and has more foot height clearance. Bottom Row: ExBody + AMP tends to walk in a straight-leg way and has less foot height clearance during walking.

limited torque, DoFs of the real robot, we design ExBody to only mimic the arm motions $\mathbf{g}^e \sim \mathcal{G}^e$ while the whole-body's objective is to track root movement goals $\mathbf{g}^m \sim \mathcal{G}^m$. We show in Tab. IV that tracking the Whole-Body expressions will result in reduced performance with all metrics. The robot's lower body movements exhibit numerous artifacts, notably that while the reference motion is designed for a single step, the robot executes multiple steps in an attempt to stabilize.

Comparisons with adversarial methods. We also compare with the adversarial methods that can serve as a regularizer on top of our method [34, 33, 12]. Our method plus an AMP regularizer demonstrates better results in terms of MEL and $MERP$. But just like the Random Sample baseline, it is at a great cost of linear velocity tracking ($MELV$). The policy generated by AMP often results in a gait with less knee flexion and inadequate foot clearance, leading to toes that tilt non-horizontally and a tendency to stumble while walking, as illustrated in Fig. 7. These characteristics could pose challenges for sim-to-real transfer. We also tested a series of motions in the real world as shown in Tab. V and recorded their roll and pitch variations as an indicator of how stable the policy is. We can see that Ours + AMP has more shaking than ours. Our method has a stable stepping gait while the AMP one tries to use straight legs and stand in place, which results in significant stumbling and feet artifacts shown in Fig. 7. ExBody + AMP NoReg tries to replace the regularization terms in Tab. III. However, it has even worse performance, demonstrating a high-frequency jittery movement that is not feasible for sim-to-real transfer, indicating for such a complex system, AMP reward itself is not sufficient.

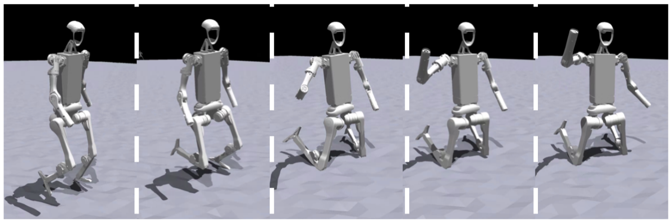


Fig. 8: Random Sampling $\mathbf{g}^m$ results in a behavior that the policy immediately kneels down after initialization, trying to be as stable as possible while ignoring the root movement goal $\mathbf{g}^m$.

Results in Real World We extensively evaluate our framework in the real world, with Fig. 1 and Fig. 7 showcasing the replaying of CMU MoCap trajectories, and Fig. 9 evaluating text prompting diffusion pipeline. Notably, as walking gaits are not predefined during the training phase, our robot autonomously learns to synchronize its stepping frequency with the upper body poses during motion playback. For additional demonstrations, please refer to the supplementary materials, where more result videos are available.

V. Related Work

Whole-Body Control with Legged Robots Legged robots often need to coordinate the entire body to complete some tasks or do some motions such as dancing, reaching for a far object, etc, which were previously primarily achieved by dynamics modeling and control [39, 61, 21, 40, 9, 26, 57]. However, for a humanoid robot that has a high degree of freedom [17, 27, 20, 7, 2, 1], it will require substantial engineering and modeling [51] and are sensitive to real-world dynamics changes. Recent learning-based methods [15, 22, 5, 23, 48, 25, 24] achieved whole-body locomotion and manipulation for a

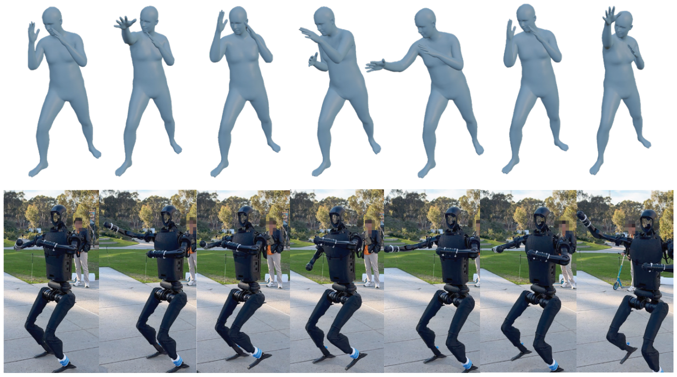


Fig. 9: Text2Motion trajectories replay. A motion sequence is prompted offline with the input ”a man mimics boxing punches” through MDM [54]. Our robot presents robust, responsive, and precise tracking performance.

quadruped robot. These advances also enable better learning-based humanoid control [46, 52, 30, 49]. However, most of the studies focus more on the locomotion side or learning a relatively small dataset. Different from all previous works, our work enables whole-body control for expressive motions on a human-sized robot in the real world.

Legged Locomotion Blind-legged locomotion across challenging terrains has been widely studied, via reward specification [38, 28, 14, 13], via imitation learning [11] and gait heuristics [29, 50]. Vision-based locomotion has achieved great successes traversing stairs [4, 60, 37, 10], conquering parkour obstacles [63, 6], manipulating boxes [8]. However, these works have not fully taken advantage of demonstration data. Even works utilizing re-targeted animal motions or pre-optimized trajectories still leverage a very small dataset [42, 56, 11, 16, 59], while our framework can benefit from learning with a large-scale motion dataset.

Physics-based Character Animation Whole-body humanoid motion control has been widely studied in the realm of computer graphics, where the goal is to generate realistic character behaviors. Adversarial methods such as [44, 45, 53, 19] suffer from mode collapse as the motions get more and more. Peng et al. [45] used a unit sphere latent space to represent the 187 motions. However, it stills suffers from mode collapse and utilized additional skill discovery objective. Imitation-based methods [58, 55, 62, 41] alleviate this problem by decoupling control and motion generation, where a general motion tracking controller is trained to track any motions and a motion generator outputs motions to track. These works demonstrated successful transfer to real robot quadrupeds [43, 11]. [58] separate the entire CMU MoCap data into several clusters and train mixture-of-expert policies to reproduce physically plausible controllers for the entire dataset. Luo et al. [34] used a similar idea by progressively assigning new networks to learn new motions. However, these methods are hard to transfer to the real humanoid robot because of the unrealistic character model (SMPL humanoid [32] has a total of 69 DoFs with 23 actuated spherical joints and each joint has 3 DoFs, there is usually no torque limit), privileged information used in the simulation (world coordinates of robots, velocities, etc) demonstrated in Tab. I. Thus we propose Expressive Whole-Body Control to address this problem. ExBody relaxes the lower body tracking objective and uses a whole-body root movement goal instead. While considering the capability of our robot, we select a subset of motions that includes mainly walking and everyday behaviors and expressions and we use only one single network for all the motions.

Motions	Ours	Ours+AMP
Walk, Exaggerated Stride	0.054	0.087
Zombie Walk	0.072	0.11
Wave Hello	0.062	0.095
Walk Happily	0.037	0.074
Punch	0.052	0.055
Direct Traffic, Wave, Point	0.037	0.094
Highfive	0.04	0.084
Basketball Signals	0.045	0.081
Adjust Hair Walk	0.042	0.09
Russian Dance	0.063	0.1
Mummy Walk	0.064	0.086
Boxing	0.075	0.068
Hug	0.037	0.086
Shake Hand	0.036	0.099
Mean	0.051	0.087

TABLE V: We report the mean absolute roll and pitch angle for a 10-second test in the real world for each motion.

VI. Discussions

We introduce a method designed to enable a humanoid robot to track expressive upper body motions while ensuring the maintenance of robust locomotion capabilities in the wild. This method benefits from extensive training on large motion datasets and the use of RSI, equipping the robot with the ability to mimic a wide range of motions responsively and to robustly execute root movement commands that are randomly sampled. Our comprehensive evaluation encompasses both simulated environments and real-world settings. Additionally, the design choices within our framework are rigorously analyzed: quantitatively through simulations and qualitatively through real-world scenarios. We believe our method paves the way for the development of reliable and versatile humanoid robots, capable of performing multiple functions effectively.

VII. Limitations

In the process of retargeting, the direct mapping of joint angles from the MoCap dataset to the H1 robot, which possesses fewer DoF, leads to a loss of information. Consequently, this can result in the retargeted behavior deviating from the original motion. To mitigate these discrepancies, the application of high-fidelity motion retargeting methods could yield significant improvements. Additionally, given the considerable weight and cost associated with the humanoid robot, the occurrence of falls can sometimes result in broken parts. Therefore, the implementation of a reliable protective and recovery system is imperative to minimize the need for costly repairs.

References

[1] Agility Robotics, Robots, 2024, www.agilityrobotics.com/robots, [Online; accessed Feb. 2024].

[2] Boston Dynamics, Atlas, 2024, www.bostondynamics.com/atlas, [Online; accessed Feb. 2024].

[3] Unitree Robotics, H1, 2024, www.unitree.com/h1, [Online; accessed Feb. 2024].

[4] Ananye Agarwal, Ashish Kumar, Jitendra Malik, and Deepak Pathak. Legged locomotion in challenging terrains using egocentric vision. In *Conference on Robot Learning*, pages 403–415. PMLR, 2023.

[5] Xuxin Cheng, Ashish Kumar, and Deepak Pathak. Legs as manipulator: Pushing quadrupedal agility beyond locomotion. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[6] Xuxin Cheng, Kexin Shi, Ananye Agarwal, and Deepak Pathak. Extreme parkour with legged robots. *arXiv preprint arXiv:2309.14341*, 2023.

[7] 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. IEEE, 2021.

[8] Jeremy Dao, Helei Duan, and Alan Fern. Sim-to-real learning for humanoid box loco-manipulation. *arXiv preprint arXiv:2310.03191*, 2023.

[9] Behzad Dariush, Michael Gienger, Bing Jian, Christian Goerick, and Kikuo Fujimura. Whole body humanoid control from human motion descriptors. In *2008 IEEE International Conference on Robotics and Automation*, pages 2677–2684. IEEE, 2008.

[10] Helei Duan, Bikram Pandit, Mohitvishnu S Gadde, Bart Jaap van Marum, Jeremy Dao, Chanho Kim, and Alan Fern. Learning vision-based bipedal locomotion for challenging terrain. *arXiv preprint arXiv:2309.14594*, 2023.

[11] Alejandro Escontrela, Xue Bin Peng, Wenhao Yu, Tingnan Zhang, Atil Iscen, Ken Goldberg, and Pieter Abbeel. Adversarial motion priors make good substitutes for complex reward functions. 2022 ieee. In *International Conference on Intelligent Robots and Systems (IROS)*, volume 2, 2022.

[12] Alejandro Escontrela, Xue Bin Peng, Wenhao Yu, Tingnan Zhang, Atil Iscen, Ken Goldberg, and Pieter Abbeel. Adversarial motion priors make good substitutes for complex reward functions. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 25–32. IEEE, 2022.

[13] Jiawei Fu, Yunlong Song, Yan Wu, Fisher Yu, and Davide Scaramuzza. Learning deep sensorimotor policies for vision-based autonomous drone racing, 2022.

[14] Zipeng Fu, Ashish Kumar, Jitendra Malik, and Deepak Pathak. Minimizing energy consumption leads to the emergence of gaits in legged robots. *Conference on Robot Learning (CoRL)*, 2021.

[15] Zipeng Fu, Xuxin Cheng, and Deepak Pathak. Deep whole-body control: learning a unified policy for manipulation and locomotion. In *Conference on Robot Learning*, pages 138–149. PMLR, 2023.

[16] Yuni Fuchioka, Zhaoming Xie, and Michiel Van de Panne. Opt-mimic: Imitation of optimized trajectories for dynamic quadruped behaviors. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5092–5098. IEEE, 2023.

[17] Jessy W Grizzle, Jonathan Hurst, Benjamin Morris, Hae-Won Park, and Koushil Sreenath. Mabel, a new robotic bipedal walker and runner. In *2009 American Control Conference*, pages 2030–2036. IEEE, 2009.

[18] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5152–5161, June 2022.

[19] Mohamed Hassan, Yunrong Guo, Tingwu Wang, Michael Black, Sanja Fidler, and Xue Bin Peng. Synthesizing physical character-scene interactions. 2023. doi: 10.1145/3588432.3591525. URL https://doi.org/10.1145/3588432.3591525.

[20] 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.

[21] Marco Hutter, Christian Gehring, Dominic Jud, Andreas Lauber, C Dario Bellicoso, Vassilios Tsounis, Jemin Hwangbo, Karen Bodie, Peter Fankhauser, Michael Bloesch, et al. Anymal-a highly mobile and dynamic quadrupedal robot. In *IROS*, 2016.

[22] Hiroshi Ito, Kenjiro Yamamoto, Hiroki Mori, and Tetsuya Ogata. Efficient multitask learning with an embodied predictive model for door opening and entry with whole-body control. *Science Robotics*, 7(65):eaax8177, 2022.

[23] Seunghun Jeon, Moonkyu Jung, Suyoung Choi, Beomjoon Kim, and Jemin Hwangbo. Learning whole-body manipulation for quadrupedal robot. *arXiv preprint arXiv:2308.16820*, 2023.

[24] Yandong Ji, Zhongyu Li, Yinan Sun, Xue Bin Peng, Sergey Levine, Glen Berseth, and Koushil Sreenath. Hierarchical reinforcement learning for precise soccer shooting skills using a quadrupedal robot. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1479–1486. IEEE, 2022.

[25] Yandong Ji, Gabriel B Margolis, and Pulkit Agrawal. Dribblebot: Dynamic legged manipulation in the wild. *arXiv preprint arXiv:2304.01159*, 2023.

[26] Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kazuhito Yokoi, and Hirohisa Hirukawa. The 3d linear inverted pendulum mode: A simple modeling for a biped walking pattern generation. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next*

Millennium (Cat. No. 01CH37180), volume 1, pages 239–246. IEEE, 2001.

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

[28] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. *arXiv preprint arXiv:2107.04034*, 2021.

[29] Zhongyu Li, Xuxin Cheng, Xue Bin Peng, Pieter Abbeel, Sergey Levine, Glen Berseth, and Koushil Sreenath. Reinforcement learning for robust parameterized locomotion control of bipedal robots. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2811–2817. IEEE, 2021.

[30] Zhongyu Li, Xue Bin Peng, Pieter Abbeel, Sergey Levine, Glen Berseth, and Koushil Sreenath. Robust and versatile bipedal jumping control through multi-task reinforcement learning. *arXiv preprint arXiv:2302.09450*, 2023.

[31] Joan Llobera and Caecilia Charbonnier. Physics-based character animation and human motor control. *Physics of Life Reviews*, 2023.

[32] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, October 2015.

[33] Zhengyi Luo, Jinkun Cao, Josh Merel, Alexander Winkler, Jing Huang, Kris Kitani, and Weipeng Xu. Universal humanoid motion representations for physics-based control. *arXiv preprint arXiv:2310.04582*, 2023.

[34] Zhengyi Luo, Jinkun Cao, Alexander W. Winkler, Kris Kitani, and Weipeng Xu. Perpetual humanoid control for real-time simulated avatars. In *International Conference on Computer Vision (ICCV)*, 2023.

[35] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. Amass: Archive of motion capture as surface shapes. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2019. URL https://amass.is.tue.mpg.de.

[36] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

[37] Gabriel B Margolis, Tao Chen, Kartik Paigwar, Xiang Fu, Donghyun Kim, Sangbae Kim, and Pulkit Agrawal. Learning to jump from pixels. *arXiv preprint arXiv:2110.15344*, 2021.

[38] Gabriel B Margolis, Ge Yang, Kartik Paigwar, Tao Chen, and Pulkit Agrawal. Rapid locomotion via reinforcement learning. *arXiv preprint arXiv:2205.02824*, 2022.

[39] Hirofumi Miura and Isao Shimoyama. Dynamic walk of a biped. *IJRR*, 1984.

[40] Federico L Moro and Luis Sentis. Whole-body control of humanoid robots. *Humanoid Robotics: A reference, Springer, Dordrecht*, 2019.

[41] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Trans. Graph.*, 37(4):143:1–143:14, July 2018. ISSN 0730-0301. doi: 10.1145/3197517.3201311. URL http://doi.acm.org/10.1145/3197517.3201311.

[42] Xue Bin Peng, Erwin Coumans, Tingnan Zhang, Tsang-Wei Lee, Jie Tan, and Sergey Levine. Learning agile robotic locomotion skills by imitating animals. April 2020.

[43] Xue Bin Peng, Erwin Coumans, Tingnan Zhang, Tsang-Wei Edward Lee, Jie Tan, and Sergey Levine. Learning agile robotic locomotion skills by imitating animals. In *Robotics: Science and Systems*, 07 2020. doi: 10.15607/RSS.2020.XVI.064.

[44] 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.

[45] Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Trans. Graph.*, 41(4), July 2022.

[46] Ilija Radosavovic, Tete Xiao, Bike Zhang, Trevor Darrell, Jitendra Malik, and Koushil Sreenath. Real-world humanoid locomotion with reinforcement learning. *arXiv:2303.03381*, 2023.

[47] Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on Robot Learning*, pages 91–100. PMLR, 2022.

[48] Clemens Schwarke, Victor Klemm, Matthijs Van der Boon, Marko Bjelonic, and Marco Hutter. Curiosity-driven learning of joint locomotion and manipulation tasks. In *Proceedings of The 7th Conference on Robot Learning*, volume 229, pages 2594–2610. PMLR, 2023.

[49] Mingyo Seo, Steve Han, Kyutae Sim, Seung Hyeon Bang, Carlos Gonzalez, Luis Sentis, and Yuke Zhu. Deep imitation learning for humanoid loco-manipulation through human teleoperation. In *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, pages 1–8. IEEE, 2023.

[50] Jonah Siekmann, Kevin Green, John Warila, Alan Fern, and Jonathan Hurst. Blind bipedal stair traversal via sim-to-real reinforcement learning. *arXiv preprint arXiv:2105.08328*, 2021.

[51] Koushil Sreenath, Hae-Won Park, Ioannis Poulakakis, and Jessy W Grizzle. A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on mabel. *IJRR*, 2011.

[52] Annan Tang, Takuma Hiraoka, Naoki Hiraoka, Fan Shi, Kento Kawaharazuka, Kunio Kojima, Kei Okada, and Masayuki Inaba. Humanmimic: Learning natural locomotion and transitions for humanoid robot via wasserstein adversarial imitation. *arXiv preprint arXiv:2309.14225*, 2023.

[53] Chen Tessler, Yoni Kasten, Yunrong Guo, Shie Mannor, Gal Chechik, and Xue Bin Peng. Calm: Conditional adversarial latent models for directable virtual characters. In *ACM SIGGRAPH 2023 Conference Proceedings*, SIGGRAPH '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701597. doi: 10.1145/3588432.3591541. URL https://doi.org/10.1145/3588432.3591541.

[54] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022.

[55] Tingwu Wang, Yunrong Guo, Maria Shugrina, and Sanja Fidler. Unicon: Universal neural controller for physics-based character motion, 2020.

[56] Yikai Wang, Zheyuan Jiang, and Jianyu Chen. Amp in the wild: Learning robust, agile, natural legged locomotion skills. *arXiv preprint arXiv:2304.10888*, 2023.

[57] Eric R Westervelt, Jessy W Grizzle, and Daniel E Koditschek. Hybrid zero dynamics of planar biped walkers. *IEEE transactions on automatic control*, 48(1): 42–56, 2003.

[58] Jungdam Won, Deepak Gopinath, and Jessica Hodgins. A scalable approach to control diverse behaviors for physically simulated characters. *ACM Trans. Graph.*, 39(4), 2020. URL https://doi.org/10.1145/3386569.3392381.

[59] Ruihan Yang, Zhuoqun Chen, Jianhan Ma, Chongyi Zheng, Yiyu Chen, Quan Nguyen, and Xiaolong Wang. Generalized animal imitator: Agile locomotion with versatile motion prior. *arXiv preprint arXiv:2310.01408*, 2023.

[60] Ruihan Yang, Ge Yang, and Xiaolong Wang. Neural volumetric memory for visual locomotion control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1430–1440, 2023.

[61] KangKang Yin, Kevin Loken, and Michiel Van de Panne. Simbicon: Simple biped locomotion control. *ACM Transactions on Graphics*, 2007.

[62] Haotian Zhang, Ye Yuan, Viktor Makoviychuk, Yunrong Guo, Sanja Fidler, Xue Bin Peng, and Kayvon Fatahalian. Learning physically simulated tennis skills from broadcast videos. *ACM Trans. Graph.*, 42(4), jul 2023. ISSN 0730-0301. doi: 10.1145/3592408. URL https://doi.org/10.1145/3592408.

[63] Ziwen Zhuang, Zipeng Fu, Jianren Wang, Christopher Atkeson, Sören Schwertfeger, Chelsea Finn, and Hang Zhao. Robot parkour learning. In *Conference on Robot Learning (CoRL)*, 2023.

Expressive Whole-Body Control for Humanoid Robots

Appendix

A. *Motion Dataset Selection*

We curated the training and inference dataset shown in Tab. II to single person motions on flat terrain to ensure these expressive motions are reasonable to track. We filter the motions in CMU MoCap by checking if the following keywords are in the description of the motion: [”walk”, ”navigate”, ”basketball”, ”dance”, ”punch”, ”fight”, ”push”, ”pull”, ”throw”, ”catch”, ”crawl”, ”wave”, ”high five”, ”hug”, ”drink”, ”wash”, ”signal”, ”balance”, ”strech”, ”leg”, ”bend”, ”squat”, ”traffic”, ”high-five”, ”low-five”]. And excluding motions with the following keywords: [”ladder”, ”suitcase”, ”uneven”, ”terrain”, ”stair”, ”stairway”, ”stairwell”, ”clean”, ”box”, ”climb”, ”backflip”, ”handstand”, ”sit”, ”hang”].

B. *Additional Training Details*

Rewards Expression and root movement goal rewards are specified in Tab. III. Regularization reward items are listed in Tab. VI, where h_{feet} is feet height, t_i^{air} indicates the duration each foot remains airborne, $\mathbb{1}_{\text{new contact}}$ represents new foot contact with ground, $\mathbf{F}_i^{xy}$ and F_i^z are for foot contact force in horizontal plane and along the z-axis respectively, with F_{th} is the contact force threshold. $\ddot{\mathbf{q}}$ is joint acceleration, $\mathbf{a}_t$ is action at timestep t, $\mathbb{1}_{\text{collision}}$ indicates self-collision, $q_{\max}$ and $q_{\min}$ are limits for joint positions, $\mathbf{g}_{\text{xy}}$ is gravity vector projected on horizontal plane. We specifically add feet related reward items to make sure the feet are comfortably lifted high enough and having a reasonable contact force with the ground when putting down.

Term	Expression	Weight
	Feet Related	
Height	$\max(\lvert\mathbf{h}_{\text{feet}}\rvert - 0.2, 0)$	2.0
Time in Air	$\sum t_i^{\text{air}} * \mathbb{1}_{\text{new contact}}$	10.0
Drag	$\sum \lvert\mathbf{v}_i^{\text{foot}}\rvert * \sim \mathbb{1}_{\text{new contact}}$	-0.1
Contact Force	$\mathbb{1}\left\{\lvert F_i^z\rvert \geq F_{\text{th}}\right\} * (\lvert F_i^z\rvert - F_{\text{th}})$	-3e-3
Stumble	$\mathbb{1}\left\{\exists i,\ \lvert\mathbf{F}_i^{xy}\rvert > 4\lvert F_i^z\rvert\right\}$	-2.0
	Other Items	
DoF Acceleration	$\lvert\ddot{\mathbf{q}}\rvert^2$	-3e-7
Action Rate	$\lvert\mathbf{a}_{t-1} - \mathbf{a}_t\rvert$	-0.1
Energy	$\lvert\ddot{\mathbf{q}}\rvert^2$	-1e-3
Collision	$\mathbb{1}_{\text{collision}}$	-0.1
DoF Limit Violation	$\mathbb{1}_{q_i > q_{\max} \lvert\rvert q_i < q_{\min}}$	-10.0
DoF Deviation	$\lvert\mathbf{q}_{\text{default}}^{\text{low}} - \mathbf{q}^{\text{low}}\rvert^2$	-10.0
Vertical Linear Velocity	v_z^2	-1.0
Horizontal Angular Velocity	$\lvert\boldsymbol{\omega}_{xy}\rvert^2$	-0.4
Projected Gravity	$\lvert\mathbf{g}_{xy}\rvert^2$	-2.0

TABLE VI: Regularization Rewards Specification

Training Parameters We use PPO with hyperparameters listed in Tab. VII to train the policy. AMP baseline parameters used in Section IV are provided in Tab. VIII.

Hyperparameter	Value
Discount Factor	0.99
GAE Parameter	0.95
Timesteps per Rollout	21
Epochs per Rollout	5
Minibatches per Epoch	4
Entropy Bonus (α_2)	0.01
Value Loss Coefficient (α_1)	1.0
Clip Range	0.2
Reward Normalization	yes
Learning Rate	1e$-$3
# Environments	4096
Optimizer	Adam

TABLE VII: PPO hyperparameters.

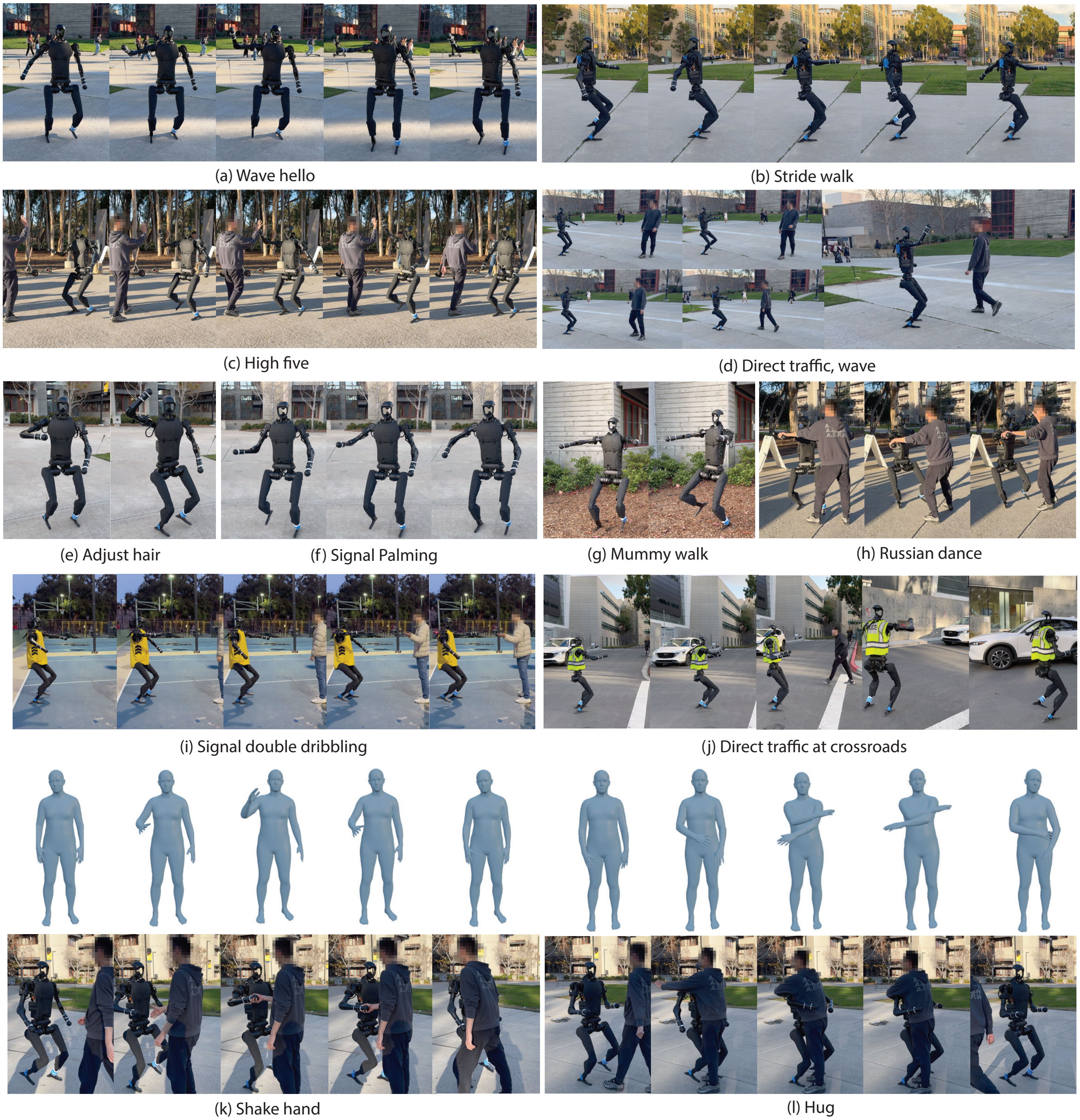
Hyperparameter	Value
Discriminator Hidden Layer Dim	$[1024, 512]$
Replay Buffer Size	1000000
Demo Buffer Size	200000
Demo Fetch Batch Size	512
Learning Batch Size	4096
Learning Rate	$1e-4$
Reward Coefficient	4.0
Gradient Penalty Coefficient	1.0

TABLE VIII: AMP hyperparameters.

Text2motion Diffusion Model We utilize the pre-trained transformer based MDM model to generate human motions from text prompts. In each generating process, 10 repetitions are requested and the most reasonable motion is manually selected for retargetting.

C. *Additional Real World Results Visualization*

We provide detailed visualization for some motions evaluated in the real world. Fig. 10 presents 8 motions from CMU MoCap and 2 motions from text2motion diffusion model. The diffusion model target motions are first generated through MDM [54] on SMPL skeleton, then we retarget this motion to H1 morphology offline. The top images in (k) and (l) are visualizations of target motions rendered with SMPL mesh in Blender.



(a) Wave hello

(b) Stride walk

(c) High five

(d) Direct traffic, wave

(e) Adjust hair

(f) Signal Palming

(g) Mummy walk

(h) Russian dance

(i) Signal double dribbling

(j) Direct traffic at crossroads

(k) Shake hand

(l) Hug

Fig. 10: Expressive motion evaluation in the real world. Target motions of (a)-(j) are from CMU MoCap. Target motions of (k) and (l) are prompted using MDM [54]. The prompts respectively are ”moving arm out to shake hands” and ”a person crosses their arms and then puts them back to their side”.