Robotics: Science and Systems 2024
Delft, Netherlands, July 15-July 19, 2024

Expressive Whole-Body Control for Humanoid Robots

Xuxin Cheng*† Yandong Ji*† Junming Chen† Ruihan Yang† Ge Yang‡ Xiaolong Wang†

†UC San Diego ‡MIT

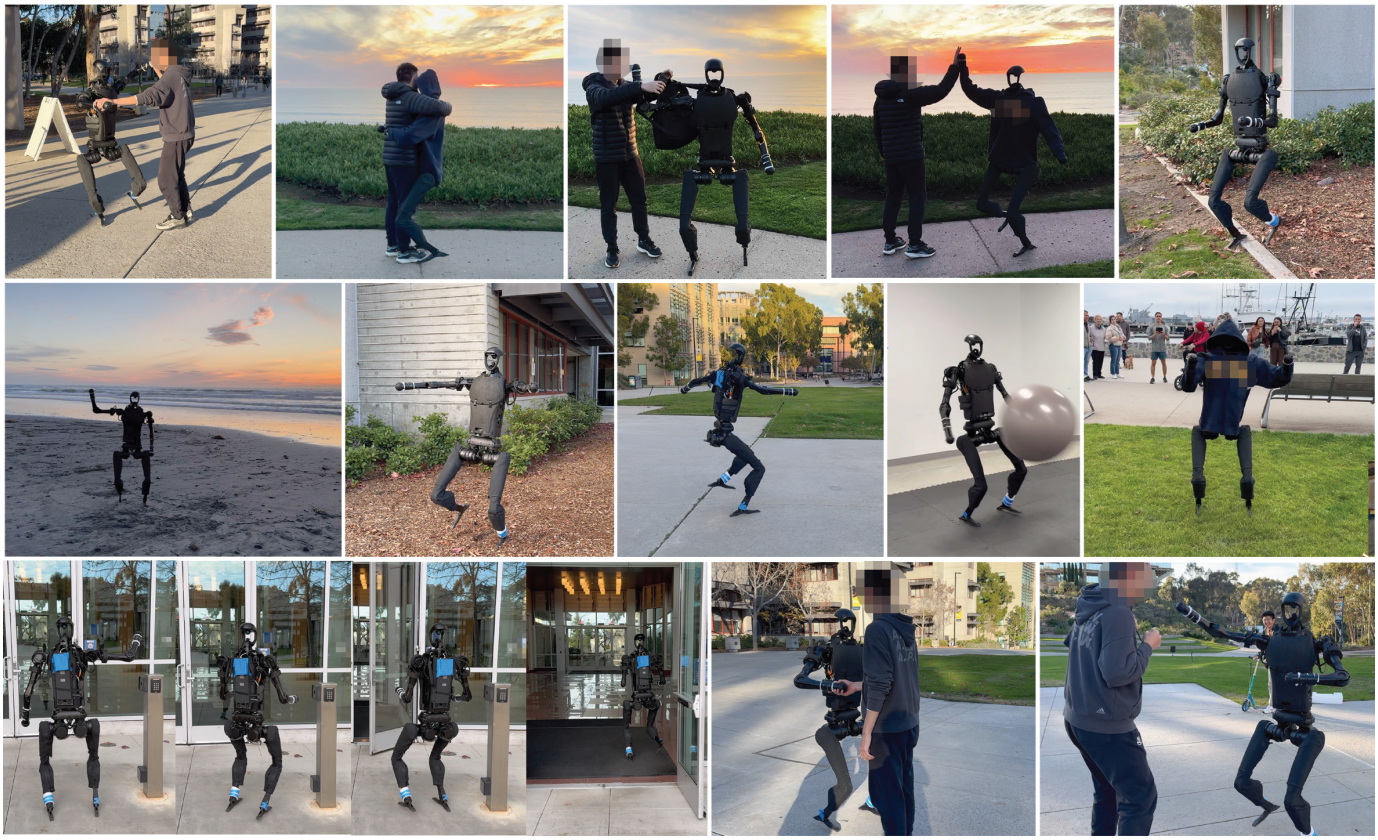


Fig. 1: Our Robot demonstrates diverse and expressive whole-body movements in different scenarios. Top Row: The robot is dancing, hugging and doing high-five with a human. Middle Row: The robot is able to walk on different terrains including gravel and wood chip paths, inclined concrete paths, grass, 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. Website: https://expressive-humanoid.github.io/.

***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 as quadruped robot dogs, or humans, whole-body control

* Equal contribution. Junming Chen is also affiliated with HKUST, work done at UC San Diego.

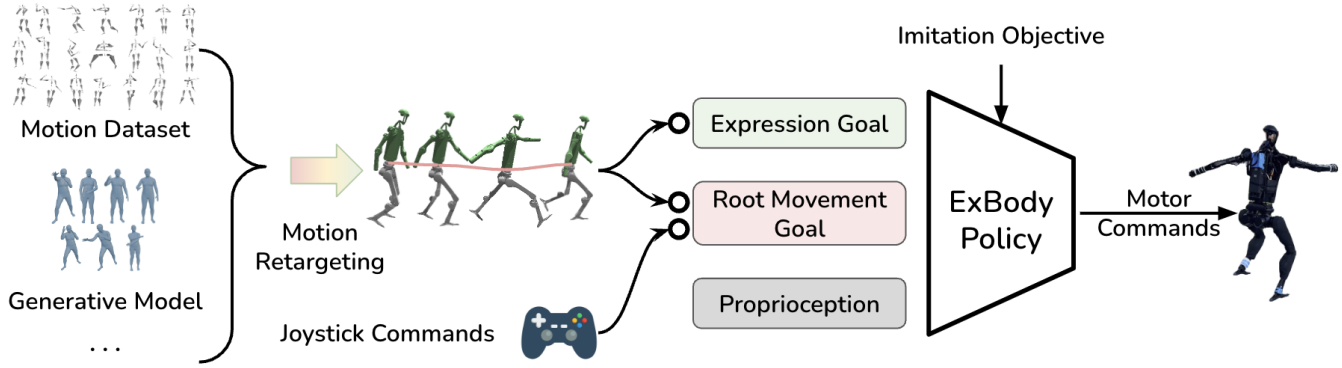


Fig. 2: Overview of our framework. Our framework is able to train on data from various sources such as static human motion datasets, generative models, video to pose models that are widely available. After motion retargeting, we acquire a repertoire of motion clips that are compatible with our robot's 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.

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 [35], and from the robotics community on using deep reinforcement learning to produce robust locomotion policy on various legged robots [31, 5]. 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 [38]) 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 motions. 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 [38]	ASE [52]
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) **Ex**pressive 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) $\mathbf{g}^e \sim \mathcal{G}^e$.

	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
O.O.D. Text to motion [64]	Boxing	1	4.0
	Hug	1	4.0
	Shake Hands	1	4.0
O.O.D. Video to motion [3]	Exaggerated greeting	1	11.0
	Put on backpack	1	11.0
	Dance: uptown funk [60]	1	15.9
	Dance: hiphop [65]	1	31.0
	O.O.D. Total	7	80.9

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. The source videos of the ones with references are taken from YouTube. Other source videos are self-recorded.

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

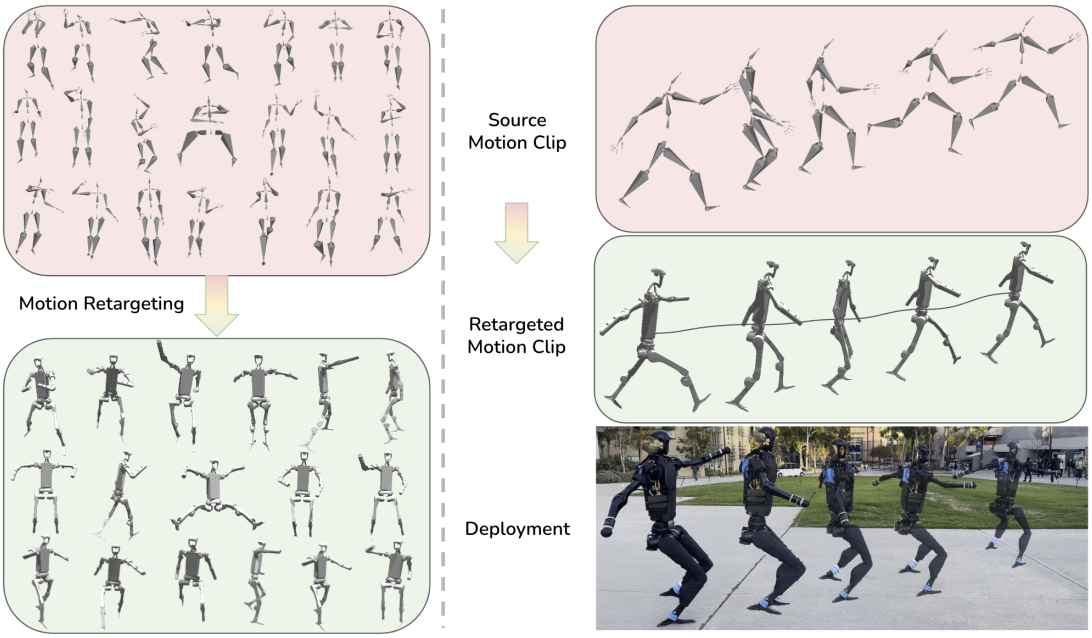


Fig. 3: Left: During training, we extract a large repertoire of retargeted motion clips and train our ExBody policy. Right: During deployment, we can replay motion that can come from a variety of sources such as static motion datasets, diffusion models, or video-to-skeleton models. For Unitree H1, the 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.

the in-distribution motions, we also test a variety of motions taken from text-to-motion [64] and video-to-motion models [3]. The corresponding source videos and real-world robot videos are provided in supplementary videos. The O.O.D. performance is further evaluated in Sec. IV.

Unlike [16], 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[4] 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 [39, 21]. 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:

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

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. 3, 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 [40, 56]. 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. 5 and the corresponding tracking rewards, ExBody can 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

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

TABLE III: Expressive Rewards Specification

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 [56]. 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.

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 can handle the regularization of the imitation learning problem with such a large dataset.
- **Random Sample**: Randomly uniformly sample root movement goals $\mathbf{g}^m$ with the range shown in Tab. IV.
- **Random Sample Small**: Smaller random sample ranges in Tab. IV

Baseline	vx	vy	roll	pitch	base height
Random Sample	± 2.0	± 1.0	± 0.5	± 0.5	[0.9, 1.1]
Random Sample Small	± 1.5	± 1.0	± 0.2	± 0.2	[0.9, 1.1]

TABLE IV: Random sample ranges.

- **Separate**: The upper body and lower body are two separately trained policies with observations and actions only for upper and lower body as well. The details of the separate policies are in supplementary materials.
- **ExBody O.O.D.**: Ours evaluated on a small out-of-distribution dataset in Tab. II.
- **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)

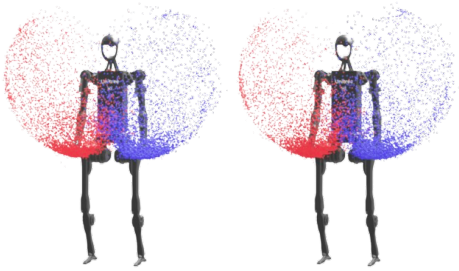


Fig. 4: We sample 10,000 points of hand positions relative to the robot. Left: retargeted motion dataset. Right: learned ExBody policy rollouts. The upper body movement from the dataset forms a natural distribution for learning.

How well does ExBody perform on tracking $\mathbf{g}^m$? From Fig. 5 we can see that for the motion sample, the tracking error is very low where the sample density is dense. In areas where the sample density is sparse, the tracking error is slightly higher. This can be explained by the lack of samples leads to inaccurate results and the difficulty of learning long-tail distributions. For the last three columns of Fig. 5, we study whether velocity goal v_x will affect the performance of other goals. We can see that our policy can track *roll, pitch* and *root height* well without being affected by walking velocity. However, the policy is not very good at some root goals with large *pitch* and *roll* angles. We hypothesize due to human-to-robot gaps, these goals can greatly influence stability. Another factor can be the robot only has one DoF at the waist, making it harder to adapt to complicated poses.

For *Random Sample*, our method performs well on a manually selected uniform sampling range. The range is among the largest root movement goals in recent literature [32, 54, 34, 70], which is discussed in detail in Tab. IV. Note that although *Random Sample* looks better than *Motion Sample*, the heatmap does not consider the sample density. The average performance is not directly implied from the heatmap and is further discussed in Tab. IV.

How well does ExBody perform on tracking $\mathbf{g}^e$? We

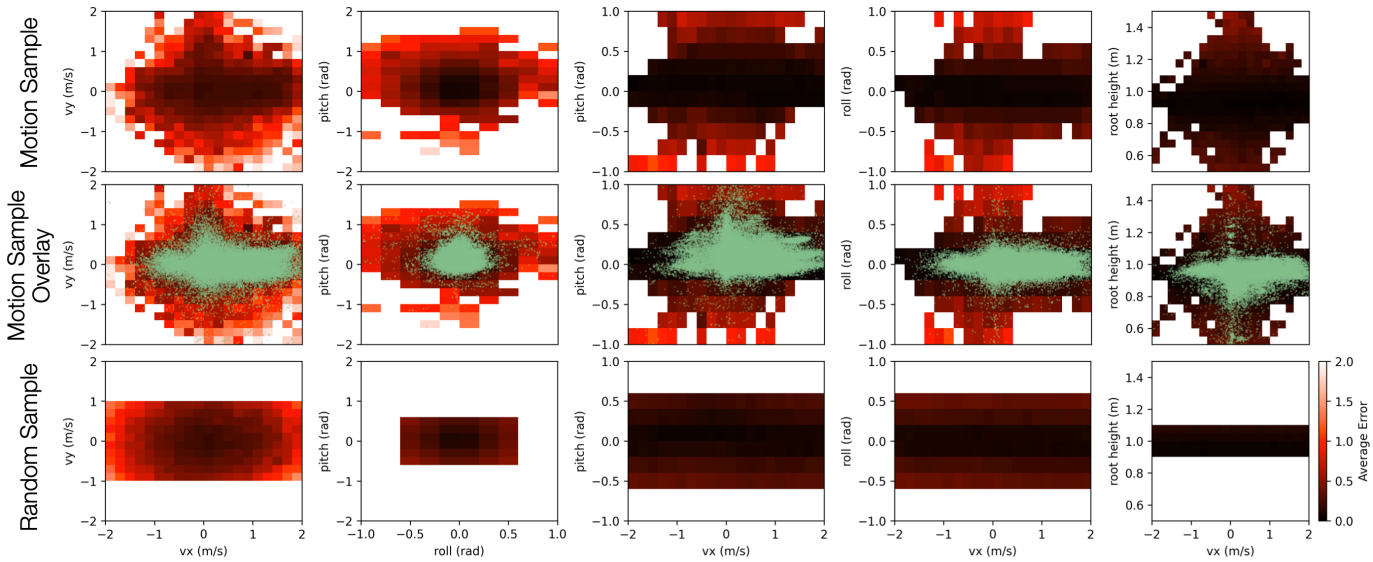


Fig. 5: Tracking error heatmaps for root movement goal G^m. Top row: goals sampled from MoCap motion dataset. Middle row: op row with the sampled goals overlay. Bottom row: uniformly random sample goals. We first bin all sampled points into a grid of size 0.2x0.2 (regardless of the unit, except the grid size along y axis of the last column is 0.05), and compute the average mean squared tracking error of all the samples in the grid. For the first two columns, the tracking error considers both x and y axes. For the last three columns, we want to see if the v_x goal will affect the tracking error of *roll, pitch* and *root height*., thus only y axis error is considered.

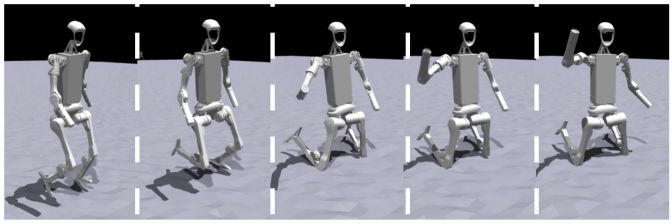


Fig. 6: Random Sampling $\mathbf{g}^m$ results in a behavior that the policy immediately kneels after initialization, trying to be as stable as possible while ignoring the root movement goal $\mathbf{g}^m$.

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. 4.

Why do we need coupled goals? 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. V, 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 due to poor exploration of the environment, resulting in a policy in which the policy actively pursues episode termination by lowering its root lower than the termination threshold as soon as possible to avoid further accumulation of negative rewards. The Random Sample baseline's behavior is a kneel-down motion for all the goals as shown in Fig. 6, 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. The Random Sample Small baseline does not generate a kneeling behavior due to a reduced sampling range that leads to an easier initial task. If we want the policy to work well on a relatively large range of commands, traditionally it is done with curriculum learning [41, 18] that can have many parameters to tune (grid v.s. box, etc). Again our method does not require such manual tuning of curriculum to work. From Tab. V, we can see the linear velocity tracking (MELV) increases dramatically, but at the cost of MERP, indicating the conflict of objective problem. However, even with a reduced sampling range, the performance is significantly worse than ours, indicating ExBody's advantage in overcoming conflicts of objectives problems. 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.

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

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
Full Body Tracking	13.28	246.11	584.40	397.25	10.76	76.46	407.88	284.69
Random Sample	16.50	181.85	704.73	326.66	16.37	38.51	**586.83**	324.10
Random Sample Small	15.99	251.74	688.82	591.77	12.09	106.94	428.10	438.40
Separate	15.38	264.67	671.24	582.38	11.42	110.55	409.60	417.23
ExBody (Ours) O.O.D.	19.26	330.33	828.99	683.97	15.39	179.26	583.04	498.05

TABLE V: Comparisons with baselines. We sample 20 seconds simulation rollouts 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 Tab. IV.

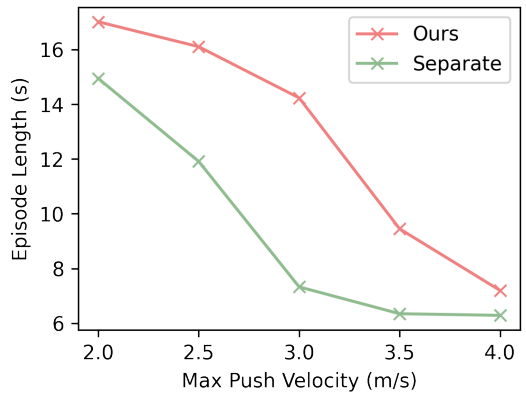


Fig. 7: We sample 20-second simulation rollouts with 4096 environments and take the mean episode length as our metric. The termination condition for an episode is when the root height is lower than 0.5m. The robot is randomly pushed every 3 seconds by setting a uniformly sampled root velocity $\mathbf{v^p} = [v_x^p, v_y^p]^T$ along the ground plane. $v_x^p, v_y^p \in [-v^{max}, v^{max}]$. v^{max} is the x axis of the figure.

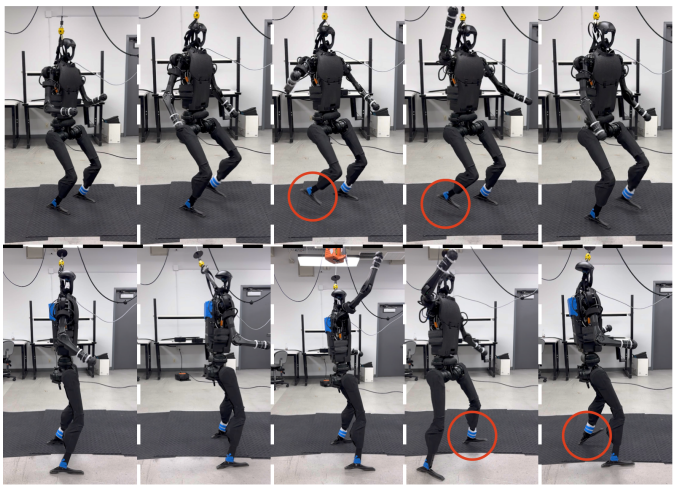


Fig. 8: H1 robot doing a High Five in the real world. Top Row: ExBody only (Ours) walks with more bent knees and has more foot height clearance. Bottom Row: ExBody + AMP tends to walk in a straight-leg way and has less foot height clearance during walking.

Unified policy is more robust than separate ones. From Fig. 7 we can see that our method is significantly more robust than the separate baseline. This is due to the drawbacks of separate modules introduced in [18]. If we want to deploy *Separate* policy in the real world, additional delay and frequency mismatch may also cause unpredictable behaviors and instability.

Comparisons with adversarial methods. We also compare with the adversarial methods that can serve as a regularizer on top of our method [38, 37, 15]. 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. 8. These characteristics could pose challenges for sim-to-real transfer. 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. 8. 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. **Evaluation on O.O.D. motions.** We also added the O.O.D. motions to the evaluation on the tracking performance in Tab. V. The numbers in Tab. V are not directly comparable because they are evaluated on different sets. However we can still see that it works better on a small O.O.D dataset than a large training set. Our method successfully generalizes to a small set of motions taken from different motion generation approaches, showing the great adaptability and potential to learn from or deploy zero-shot to even larger scale data. An example of motions is shown in Fig. 11. Additional snapshots of O.O.D. motions are in supplementary.

How expressive goal G^e affects stepping frequency We do

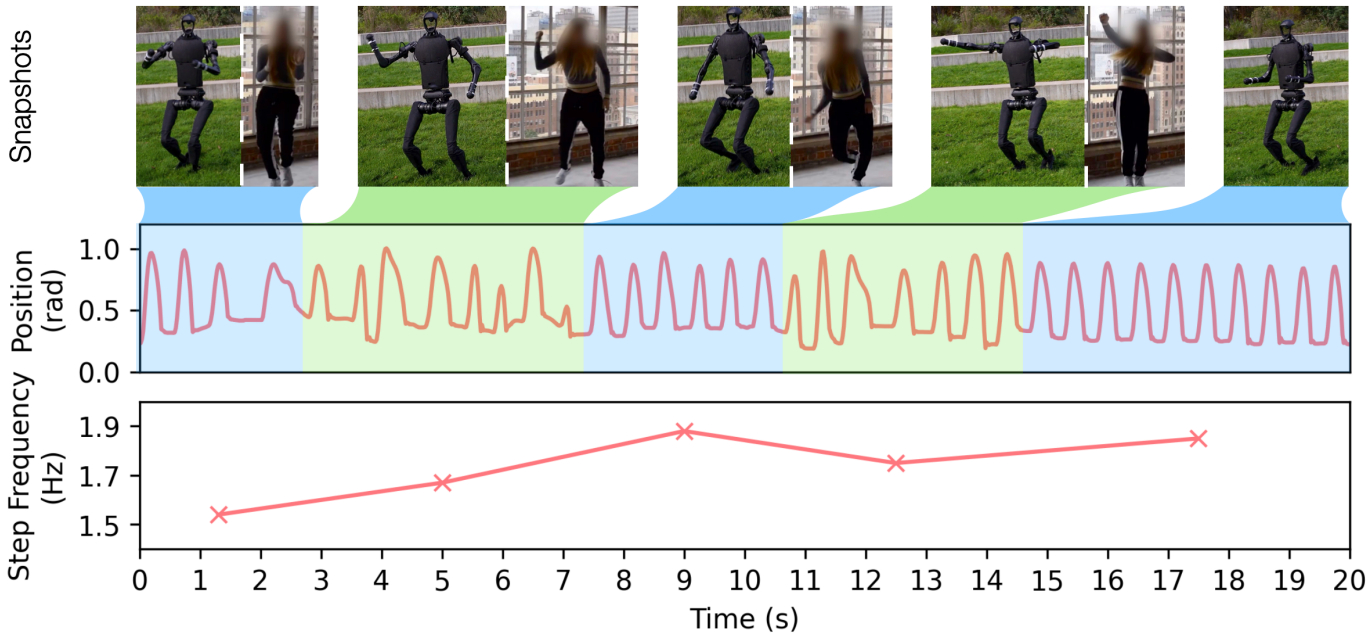


Fig. 9: Robot following an out-of-distribution dynamic dance move extracted from a YouTube video [60]. The top row shows the robot motions and original dance video snapshots. The middle row shows the motor position of the left knee to indicate the step frequency because the robot is not equipped with foot contact sensors. The blue and green segments are to differentiate different motion segments. The color itself has no meaning. The bottom row is the computed step frequency $f = n/\Delta t$ for each segment, where n and Δt are the total steps and time of a segment.

a case study in the real world where the root movement goal G^m is set to zero. We choose the uptown funk motion from O.O.D. dataset form II

From Fig. 9 we can see that the robot dynamically adjusts its stepping frequency to maintain balance. The first, second, and fourth segments contain motions that involve dynamic fast swings of the upper body. The robot adapts by taking slightly larger steps in different directions to maintain dynamic balance, which might explain why the step frequency is lower. The third and fifth segments involve less dynamic dance moves. The robot presents a rhythmic step-in-place behavior and has faster-stepping frequencies. **How root movement goal G^m affects stepping frequency** We set the expression goal G^e to the default upper body position to observe how velocity command v_x in G^m affects stepping frequency.

From Fig. 10, the robot emerges a higher step frequency than humans with low forward velocity v_x. This observation further indicates the necessity of ExBody. The morphology difference between robots and humans makes the preferred stepping frequency different, and the policy needs to optimize the gait itself instead of blindly tracking the given motions. This phenomenon can be further evidenced by *Full Body Tracking* baseline having a poor performance compared to our method and is not able to transfer to the real world, despite being more natural-looking in simulation. But the natural look also comes with additional artifacts. *Full Body Tracking* tends to add sub-steps in between the steps of the original motion, introducing additional instability and jerkiness.

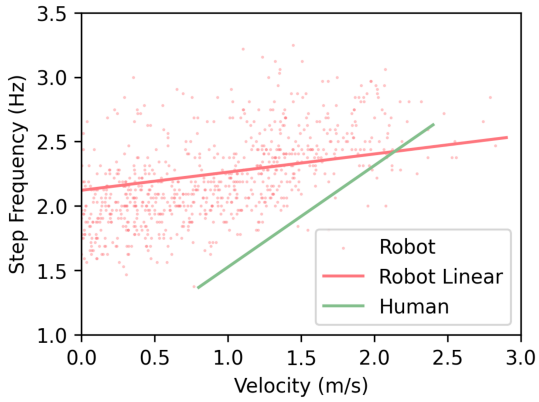


Fig. 10: We uniformly sample 4096 different $v_x \in [0, 2]$ in root movement goal G^m with 15s for each v_x. We compute the step frequency using the same method in Fig. 9 with a moving window of length $4s$. We plot a randomly sampled 1500 data points to avoid visual complexity (*Robot*) and do a linear regression with the original sampled points (*Robot Linear*). *Human* is the linearly fitted line with human locomotion data in [46]. The linearly fitted stepping frequency when $v_x = 0$ is slightly above 2.0Hz, while in Fig. 9 it is around 1.9Hz. This can be explained by the relationship not being strictly linear and sim-to-real gaps.

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

Fig. 11: Text2Motion trajectories replay. A motion sequence is prompted offline with the input "a man mimics boxing punches" through MDM [64]. Our robot presents robust, responsive, and precise tracking performance.

far object, etc, which were previously primarily achieved by dynamics modeling and control [44, 73, 24, 45, 11, 29, 68]. However, for a humanoid robot that has a high degree of freedom [20, 30, 23, 8, 2, 1], it will require substantial engineering and modeling [61] and are sensitive to real-world dynamics changes. Recent research in control [9, 12, 47, 55] has enabled the teleoperation of humanoid robots using model-based methods. Dallard et al. [9] achieve full-body synchronization between the humanoid robot and the human teleoperator by anticipating human motion. Darvish et al. [12] improve the scalability of retargeting in teleoperation of humanoid robots through inverse kinematics over the robot model. Penco et al. [47] separate humanoid teleoperation into low-level whole-body control and high-level velocity tracking to ease the burden of controlling humanoid robots. Joao et al. [55] enable robust whole-body teleoperation of humanoid robots by applying feedback forces to the operator based on the robot's movements. Despite these encouraging results, these works inherit the limitations of the underlying model-based controllers. Recent learning-based methods [18, 25, 6, 26, 57, 28, 27] achieved whole-body locomotion and manipulation for a quadruped robot. These advances also enable better learning-based humanoid control [53, 62, 33, 58]. 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 [43, 31, 17, 16], via imitation learning [14] and gait heuristics [32, 59]. Vision-based locomotion has achieved great successes traversing stairs [5, 72, 42, 13], conquering parkour obstacles [75, 7], manipulating boxes [10]. 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 [49, 67, 14, 19, 71], 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 [51, 52, 63, 22] suffer from mode collapse as the motions get more and more. Peng et al. [52] used a unit sphere latent space to represent the 187 motions. However, it still suffers from mode collapse and utilizes additional skill discovery objectives. Imitation-based methods [69, 66, 74, 48] 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 [50, 14]. [69] 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. [38] 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 [36] 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. ExBody does not rely on such information and instead relaxes the lower body tracking objective and uses a whole-body root movement goal. While considering the capability of our robot, we select a subset of motions that include mainly walking and everyday behaviors and expressions and use only one single network for all the motions.

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 further show its capability to imitate motions from generative models and the internet.

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. Unlike quadrupeds, humanoid robots now still need to start from a stand-still pose. Auto recovery and initialization could be explored to reduce the cost of doing experiments.

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] Move AI, Move One IOS, 2024, https://www.move.ai/single-camera, [Online; accessed Apr. 2024].

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

[5] 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.

[6] 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.

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

[8] 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.

[9] Antonin Dallard, Mehdi Benallegue, Fumio Kanehiro, and Abderrahmane Kheddar. Synchronized human-humanoid motion imitation. *IEEE Robotics and Automation Letters*, 8(7):4155–4162, 2023. doi: 10.1109/LRA.2023.3280807.

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

[11] 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.

[12] Kourosh Darvish, Yeshasvi Tirupachuri, Giulio Romualdi, Lorenzo Rapetti, Diego Ferigo, Francisco Javier Andrade Chavez, and Daniele Pucci. Whole-body geometric retargeting for humanoid robots. In *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, pages 679–686, 2019. doi: 10.1109/Humanoids43949.2019.9035059.

[13] 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.

[14] 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.

[15] 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.

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

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

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

[27] 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.

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

[29] 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.

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

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

[32] 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.

[33] 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.

[34] Zhongyu Li, Xue Bin Peng, Pieter Abbeel, Sergey Levine, Glen Berseth, and Koushil Sreenath. Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control. *arXiv preprint arXiv:2401.16889*, 2024.

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

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] Gabriel Margolis, Ge Yang, Kartik Paigwar, Tao Chen, and Pulkit Agrawal. Rapid locomotion via reinforcement learning. In *Robotics: Science and Systems*, 2022.

[42] 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.

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

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

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

[46] T Nguyen, Emad Gad, J Wilson, Noel Lythgo, Nicholas Haritos, et al. Evaluation of footfall induced vibration in building floor. In *Australian earthquake engineering society annual conference*, pages 1–8, 2011.

[47] Luigi Penco, Nicola Scianca, Valerio Modugno, Leonardo Lanari, Giuseppe Oriolo, and Serena Ivaldi. A multimode teleoperation framework for humanoid loco-manipulation: An application for the icub robot. *IEEE Robotics and Automation Magazine*, 26(4):73–82, 2019. doi: 10.1109/MRA.2019.2941245.

[48] 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.

[49] 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.

[50] 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.

[51] 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.

[52] 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.

[53] Ilija Radosavovic, Tete Xiao, Bike Zhang, Trevor Dar-

rell, Jitendra Malik, and Koushil Sreenath. Real-world humanoid locomotion with reinforcement learning. *arXiv:2303.03381*, 2023.

[54] Ilija Radosavovic, Tete Xiao, Bike Zhang, Trevor Darrell, Jitendra Malik, and Koushil Sreenath. Learning humanoid locomotion with transformers. *arXiv preprint arXiv:2303.03381*, 2023.

[55] Joao Ramos and Sangbae Kim. Dynamic locomotion synchronization of bipedal robot and human operator via bilateral feedback teleoperation. *Science Robotics*, 4(35):eaav4282, 2019. doi: 10.1126/scirobotics.aav4282. URL https://www.science.org/doi/abs/10.1126/scirobotics.aav4282.

[56] 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.

[57] 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.

[58] 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.

[59] 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.

[60] Mary Spinney. Learn uptown funk — dance tutorial — mary spinney, 2022. URL https://www.youtube.com/watch?v=-u_MdAfUToQ&t=175s. YouTube video.

[61] 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.

[62] 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.

[63] 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.

[64] 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.

[65] Tito Bhoy TV. Combination of basic hip hop moves, 2021. URL https://www.youtube.com/watch?v=V5IBmCP9SOI. YouTube video.

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

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

[68] 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.

[69] 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.

[70] Zhaoming Xie, Patrick Clary, Jeremy Dao, Pedro Morais, Jonathan Hurst, and Michiel van de Panne. Iterative reinforcement learning based design of dynamic locomotion skills for cassie. *arXiv preprint arXiv:1903.09537*, 2019.

[71] 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.

[72] 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.

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

[74] 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.

[75] 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	$\mathbb{1}_{q_i > q_{\text{max}} \lvert\rvert q_i < q_{\text{min}}}$	-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.

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.

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.

Details of separately trained baseline We train an upper body policy π^u. The observation $\mathcal{S}^u$ includes $s_t^u = [q_t^u, \dot{q}_t^u, \mathbf{a}_{t-1}^u]^T$. $q_t^u, \dot{q}_t^u, {\mathbf{a}_{t-1}^u}^T \in \mathbb{R}^9$ include the 9 motors of the upper body (3 shoulder and 1 elbow motors on each side, 1 waist motor). The action $\mathbf{a}_t^u$ is the target motor positions of the 9 motors we just mentioned. The rewards for π^u remain the same as in the original paper except we remove the root movement goal rewards.

The observation $\mathcal{S}^l$ for lower body policy π^l include $s_t^l = [\omega_t, r_t, p_t, \Delta y, q_t^l, \dot{q}_t^l, {\mathbf{a}_{t-1}^l}^T]$. $q_t^l, \dot{q}_t^l, {\mathbf{a}_{t-1}^l}^T \in \mathbb{R}^{10}$ include the 10 motors of the lower body (3 hip, 1 knee and 1 ankle motors on each side). The action $\mathbf{a}_t^l$ is the target motor positions of the 10 motors we just mentioned. The rewards for π^l remain the same as in the original paper except we remove the expression goal rewards. ω_t is the robot root's angular velocity, r_t, p_t is roll and pitch.

C. Dataset Visualization

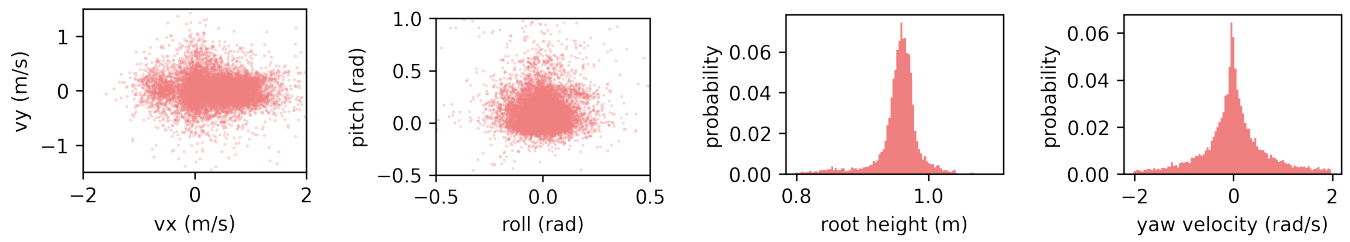
D. Additional Real World Results Visualization

We provide detailed visualization for some motions evaluated in the real world. Fig. 14 presents 8 motions from CMU MoCap and 2 motions from text2motion diffusion model. The diffusion model target motions are first generated through MDM [64] 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. Fig. 15 tracks motions recorded with Move One [3] using both online [65] and self-recorded videos.

Real-world quantatative evaluation with AMP baseline. We test a series of motions in the real world as shown in Tab. IX 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.

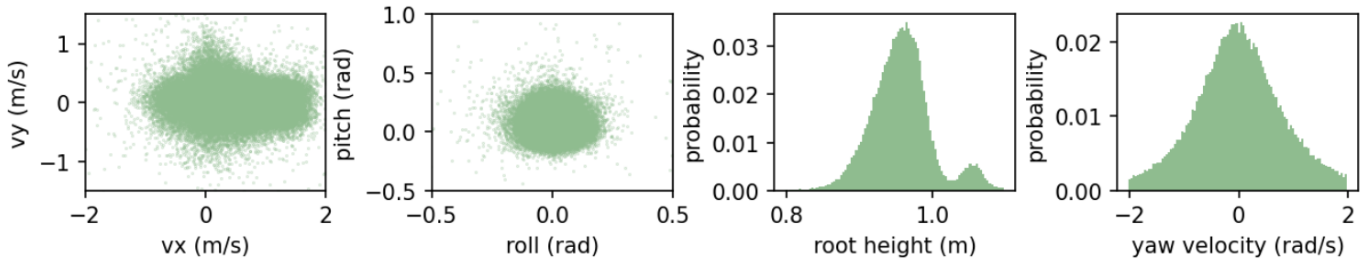
Motions	Ours	Ours+AMP
Walk, Exaggerated Stride	0.054	0.087
Zombie Walk	0.072	0.11
Wave Hello	0.062	0.095
Walk Happily	0.037	0.074
Punch	0.052	0.055
Direct Traffic, Wave, Point	0.037	0.094
Highfive	0.04	0.084
Basketball Signals	0.045	0.081
Adjust Hair Walk	0.042	0.09
Russian Dance	0.063	0.1
Mummy Walk	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 IX: We report the mean absolute roll and pitch angle for a 10-second test in the real world for each motion.

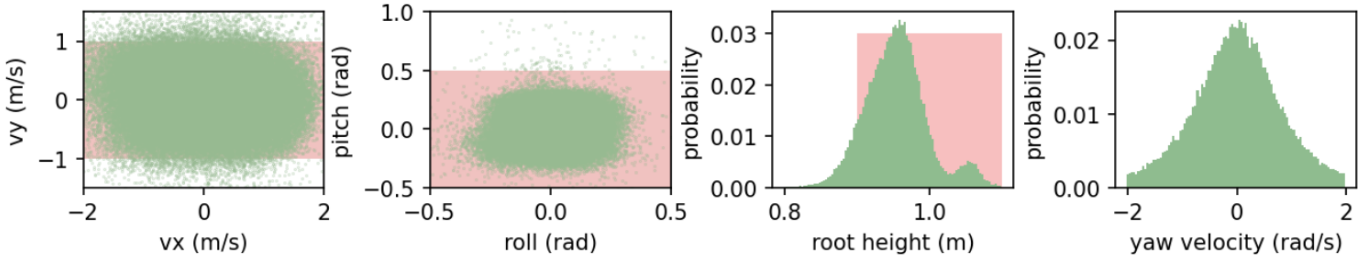


(a) (b) (c) (d)

Fig. 12: 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.



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



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

Fig. 13: 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, the 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.

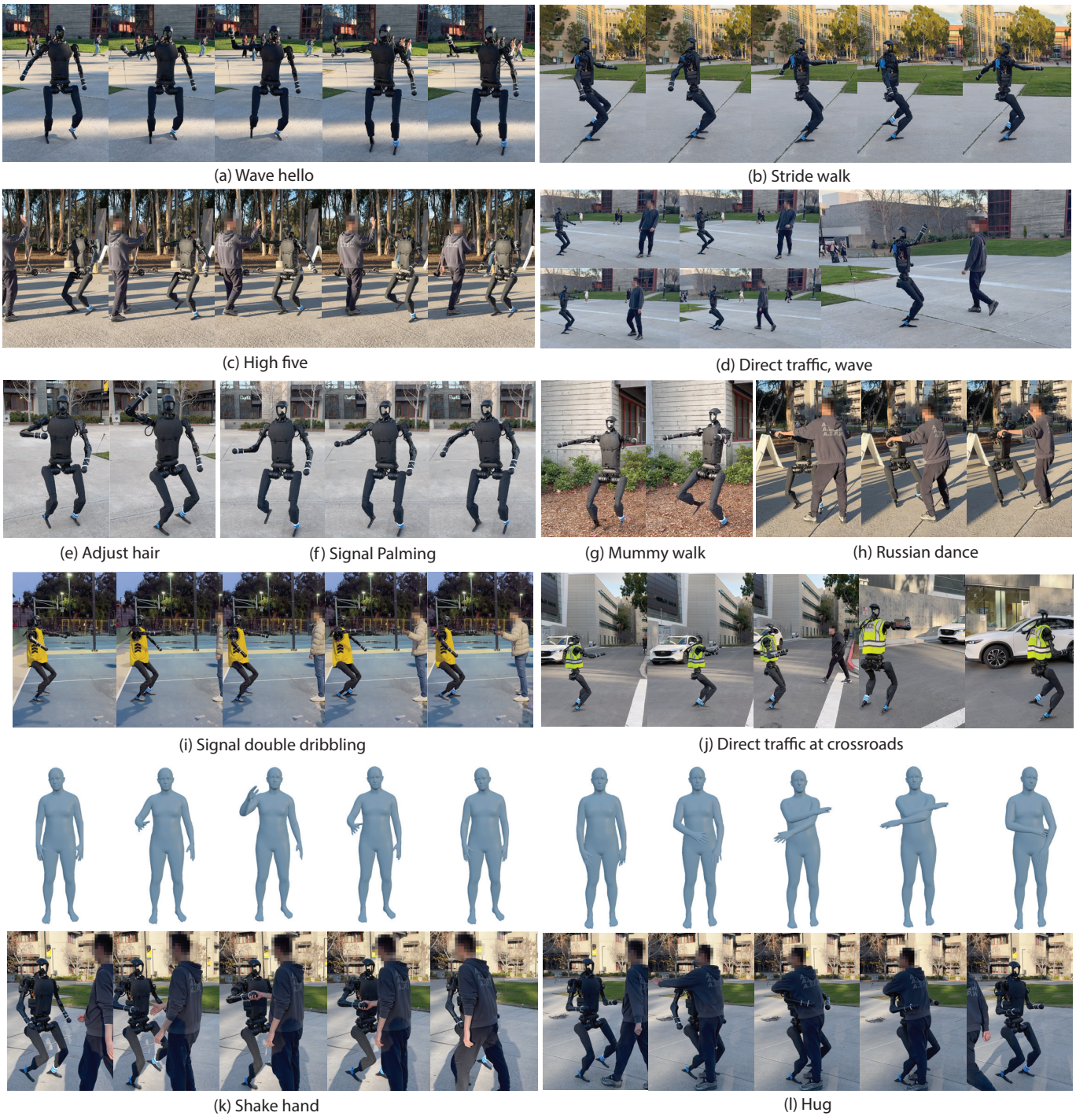


Fig. 14: Expressive motion evaluation in the real world. Target motions of (a)-(j) are from CMU MoCap. Target motions of (k) and (l) are prompted using MDM [64]. The prompts respectively are ”moving arm out to shake hands” and ”a person crosses their arms and then puts them back to their side”.

(a) Exaggerated greeting

(b) Put on backpack

(c) Dance: hiphop

(d) Dance: uptown funk

Fig. 15: Video-to-Motion evaluation. (a,b) The videos were self-recorded and subsequently processed offline using Move One [3] to create custom motions. (c,d) To assess the robustness and tracking performance of the policy, we select two challenging dance videos from the Internet.