Learning Agile Bipedal Motions on a Quadrupedal Robot

Yunfei Li[1], Jinhan Li[1], Wei Fu[1] and Yi Wu[1,2]

***Abstract*— Can a quadrupedal robot perform bipedal motions like humans? Although developing human-like behaviors is more often studied on costly bipedal robot platforms, we present a solution over a lightweight quadrupedal robot that unlocks the agility of the quadruped in an upright standing pose and is capable of a variety of human-like motions. Our framework is with a bi-level structure. At the low level is a motion-conditioned control policy that allows the quadrupedal robot to track desired base and front limb movements while balancing on two hind feet. The policy is commanded by a high-level motion generator that gives trajectories of parameterized human-like motions to the robot from multiple modalities of human input. We for the first time demonstrate various bipedal motions on a quadrupedal robot, and showcase interesting human-robot interaction modes including mimicking human videos, following natural language instructions, and physical interaction.**

I. Introduction

Empowering robots with versatile motions like humans has been an important research topic to allow them to better coexist and interact with humans [1]. Developing bipedal robot systems has attracted much interest since they have an appealing potential to mimic human behaviors thanks to their structural similarity to human beings [2]. However, existing bipedal robots are typically expensive, heavy, and power-consuming [3], [4], [5]. In contrast, quadrupedal robots are much cheaper and more lightweight and have recently demonstrated impressive sporting capabilities in various domains [6], [7], [8]. This naturally raises an interesting question: *Is it possible for quadrupedal robots to demonstrate agile human-like motions as an affordable alternative of humanoid robots?*

Enabling a quadrupedal robot to perform agile bipedal motions poses significant control challenges. Since quadrupedal robots are designed for dog-like behaviors with four legs on the ground, they must first stand upright on two feet from a four-leg pose at rest to unlock the motions of bipedal creatures. The stand-up procedure requires an agile control policy to gain enough momentum to swing up the robot and avoid flipping over at the same time. Furthermore, the bipedal motions are inherently unstable over common quadrupedal robots with spherical feet, thus the robot must actively adjust all its body parts to stay balanced once it stands up. Previous work utilized external mechanical support for the robot to stand up [9], while we aim to control a quadrupedal robot to mimic bipedal motions without any external hardware.

Another notable challenge is how to master a wide range of human-like motions. Due to the difference in kinematics and dynamics between humans and quadrupedal robots, it is difficult to directly track human motion capture data while keeping the quadruped robot balanced [10]. Therefore, motion encodings that can both represent versatile human-like behaviors and are also feasible for the embodiment of a quadrupedal robot require careful design.

[1]Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China. liyf20@mails.tsinghua.edu.cn, jxwuyi@gmail.com
[2]Shanghai Qi Zhi Institute, Shanghai, China.

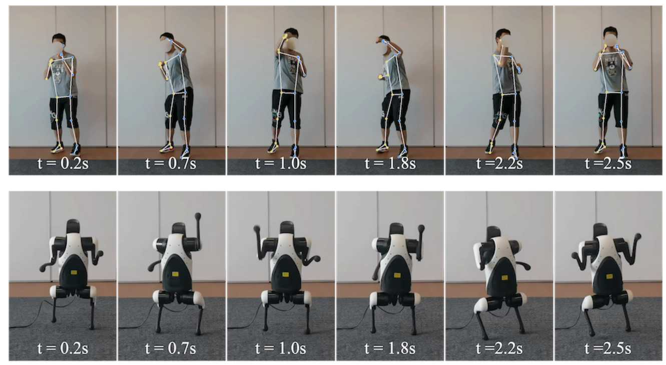

Fig. 1: A quadrupedal robot demonstrates human-like motions with only hind feet on the ground. The top row shows the reference human boxing video. The bottom row shows the robot mimicking the human motion to perform multiple punches and uppercuts at a high speed.

In this work, we present a bi-level framework that enables agile bipedal motions on a quadrupedal robot. At the low level, we train a motion-conditioned policy with model-free reinforcement learning (RL) that is capable of balancing the quadrupedal robot on its two toes while tracking reference motions at the same time. We represent motions as a sequence of the desired state of the robot base and end effectors of the front limbs, which is flexible enough to encode a spectrum of behaviors and is plausible for the quadrupedal robot to execute. The policy is trained in a calibrated simulator and then transferred to the real robot. At the high level, a motion generator parameterizes human-like motions from videos or natural language descriptions into a sequence of desired targets for the low-level policy.

We demonstrate the whole framework on an affordable quadruped platform Xiaomi CyberDog2 (∼\$1800) [11] and showcase the successful deployment of a variety of agile bipedal maneuvers and interactions with humans such as mimicking human videos to practice boxing (Fig. 1) and ballet dance, greeting commanded by natural language instructions, and walking hand-in-hand with a human. To the best of our knowledge, these agile bipedal motions are made possible on a quadrupedal robot for the first time.

II. Related Work

Learning agile skills with quadrupedal robots: There has been tremendous progress in training a variety of agile skills on quadrupedal robot platforms with reinforcement learning, such as jumping over obstacles [12], [13], [14], [15], [16], landing [17], soccer shooting [8], and goalkeeping [18], but the motions are mainly with four legs on the ground. Only a few works study the possibility of bipedal motions on quadrupedal robots [19], [16]. Besides standing up and locomotion, we consider a wider range of motions that involve the base and hand movements to make more interesting interactions with humans. [20] trained a wheeled-legged robot to stand up and navigate and [9] used an external mechanical support to stand up, while we work on a canonical quadrupedal robot without special hardware.

As for learning robot motions, one popular line of work is motion imitation from reference trajectories. These works either directly mimic the reference motions [21], [22] or adopt an adversarial approach [23] to produce motions with the same style as the source dataset [24], [25], [26]. Another research direction is to smartly parameterize motions of interest and track them with reward engineering [8], [18], [27]. Since the reference trajectories of bipedal motions for quadrupeds are not readily available, we choose to parameterize the motions using the base and front limb movements, which is versatile enough to represent a broad range of bipedal motions.

Developing controllers for bipedal motions: Traditional model-based methods like model predictive control [28], [29] and trajectory optimization [30], [31] can obtain bipedal walking controllers that follow predefined gaits. They require accurate modeling of the robot dynamics and state estimation, and run intensive optimization online to achieve good performances. Model-free reinforcement learning (RL) is another direction to obtain such controllers that alleviates the burden of heavy engineering in dynamics modeling and has demonstrated superior performance in bipedal velocity following [32] and jumping [33]. In this work, we develop a model-free RL method to learn bipedal skills, but over an affordable *quadrupedal robot* that is not specialized for bipedal motions. We also work beyond locomotion tasks and demonstrate more versatile motions after freeing the robot's front limbs from walking.

Sim-to-real transfer: Domain randomization is a powerful technique to bridge the sim-to-real gap when deploying a policy trained in simulation to a real robot [34], [35], [36]. The applied randomization range typically requires expertise to design [37]. System identification using real data is another direction [38], [39], such as learning a motor model to fit its complex dynamics [40]. There are also works that iteratively calibrate the simulator using learned trajectories and optimize the policy with new simulation parameters, and demonstrate successful transfer in precise robot arm manipulation [41] and bipedal motions [42]. We similarly leverage some real-world data to tune the randomization range of critical parameters in simulation to reduce the discrepancy between simulation and the real world for a successful policy transfer.

III. Preliminary

We parameterize the quadrupedal robot's motions using two components: the base velocities and the positions of the front hands in the robot base frame. We formulate the problem of tracking the desired bipedal motions as a Markov Decision Process (MDP) defined by $(\mathcal{S}, \mathcal{A}, \mathcal{T}, r, \gamma, \rho_0)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{T}: \mathcal{S} \times \mathcal{A} \mapsto \mathcal{S}$ is the transition function, $r: \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is the reward function, γ is the discount factor, and ρ_0 is the initial state distribution. The objective is to train a policy π^* which could lead to maximum discounted accumulative reward $\pi^* = \arg\max_\pi \mathbb{E}_{s_0 \sim \rho_0, a_t \sim \pi(\cdot|s_t)} \left[\sum_{t \geq 0} \gamma^t r(s_t, a_t) \right]$.

We adopt proximal policy optimization (PPO) [43], a state-of-the-art RL algorithm to solve the MDP. PPO jointly trains a parametrized policy network $\pi_\theta(a|s)$ and a critic function V_ϕ when optimizing the RL objective.

We consider bipedal motions with an upright standing pose in this work, and parameterize them with the linear velocity of the robot base, the base heading, and the positions of the end effectors in front limbs relative to the base.

IV. Method

We propose a bi-level framework to learn bipedal motions on a quadruped. We first train a motion-conditioned policy in simulation that allows the robot to stand on hind toes while tracking random motions. Since the desired bipedal motions are highly agile and are sensitive to physical parameters, we calibrate the simulator via a simple real-to-sim process to enable successful deployment on the real robot. Afterwards, we generate the sequence of motion targets from multiple modalities of human inputs, and command the RL policy to accomplish human-like agile bipedal maneuvers.

A. Learning a motion-conditioned policy with RL

We design a model-free RL approach to obtain a control policy that empowers a quadrupedal robot with the ability to stand up and track motions. The policy is trained in a massively parallel GPU-based simulator Isaac Gym [44].

Observations and actions: Our policy observes a history of proprioceptive information as its input and predicts the PD control target for all 12 motors. Specifically, the observation concatenates 3 frames of sensory input at $[t - 0.04s, t - 0.02s, t]$. Each frame consists of joint positions, the orientation of the robot base, the last applied actions, the desired linear and angular velocity of the robot base, and the desired positions of front toes in the base coordination. We encode the base orientation using the projection of the vectors $(0, 0, -1)$ and $(1, 0, 0)$ from the world coordination to the current robot base coordination. The policy predicts the target joint positions for the PD controller with parameters $K_p = 30$ and $K_d = 3$ at 50Hz. The critic function takes the policy observation and other privileged information that is only accessible in simulation such as the joint friction and damping as its input.

Reward design: The reward function is a summation of three categories of terms $r = r^{\text{stand}} + r^{\text{track}} + r^{\text{reg}}$ to achieve the following objectives: maintaining the standing pose with only two toes contacting the ground, tracking desired motions as accurately as possible, and avoiding drastic behaviors that are dangerous when deployed in the real world.

We define the standing reward similar to [16] as $r^{\text{stand}} = r^{\text{height}} + r^{\text{pitch}} + r^{\text{collision}}$, where r^{height} encourages the robot to lift up its body, r^{pitch} credits the robot to maintain a certain pitch angle so as to stand upright, and $r^{\text{collision}}$ penalizes all parts of the robot except its rear feet touching the ground.

The tracking reward r^{track} aims to match the linear velocity, the heading of the robot base, and the relative position of the front toes to their targets. $r^{\text{track}}(s) = r^{\text{track}}_{\text{base\_v}}(s) + r^{\text{track}}_{\text{heading}}(s) + r^{\text{track}}_{\text{hand}}(s)$, $r^{\text{track}}_x(s) = \alpha_x c_x(s) \exp\left(-\frac{e_x(s)}{\sigma_x}\right)$, $x = \{\text{base\_v}, \text{heading}, \text{hand}\}$. α_x and σ_x are weighting constants, $e_x(s)$ is the tracking error. $c_x(s)$ is a dynamic scaling factor in the range of $[0, 1]$ conditioned on the standing performance and only reaches 1 after the robot has stood upright. If c is a constant, we find it challenging to obtain a policy that both stands up high and tracks motion since reducing tracking error is easier to exploit compared to the standing reward. We observe that the robot would get stuck at a strategy that tracks hand motions while sitting on the hind legs (see Sec. V-C).

The regularization reward r^{reg} sums up necessary shaping terms that penalize unrealistic behaviors. r^{reg} includes commonly used terms from [45] that penalize drastic joint motions, joint positions and torques close to limits, and large action rates. In addition, we regulate rear foot movements to follow a trotting gait with a height of 5cm with $r^{\text{reg}}_{\text{gait}}$ and penalize slipping on the ground with $r^{\text{reg}}_{\text{slip}}$.

Episode termination: The robot resets from a sitting pose with four feet touching the ground. An episode terminates after a maximum number of 1000 steps is reached or under any of the following early-termination conditions: (a) the robot base or the front limbs is in collision after the first 30 steps, (b) any joint reaches its position limit.

Tracking targets: The agent is commanded to track random motion targets during RL training. The desired base linear velocity along the x-axis (forward and backward) is sampled from discretized bins ranging between [-0.3, 0.3]m/s and discretized at an interval of 0.1m/s every 10 seconds. The desired velocity in the y-axis (side) is fixed as 0. The desired heading direction is sampled from $[-\pi/2,\ \pi/2]$ radians relative to the current heading every 10 seconds, and the desired angular velocity (which is the observation of the policy) is updated every step using the difference between the current and desired heading directions. We update future goal positions of front limb end-effectors every 3 seconds, and compute the desired positions of front toes in each step (the observed targets) by interpolating between the last and future goal positions linearly. We make sure that the goal positions of end-effectors are reachable within the joint limit.

A sit-down policy for safe ending: We train a separate sit-down policy that can transit the robot from random stand-up poses to a quadrupedal landing pose, and append it after the motion-conditioned policy for safe termination during deployment on the real robot. The sit-down is also trained with RL. The initial state distribution for the policy is upright standing poses with random facing directions and random front limb motor positions sampled within their limits. The reward function is designed to encourage the robot's belly to face downward and to credit the joint positions for being close to those in a nominal quadrupedal standing pose.

B. Sim-to-real transfer via real-to-sim calibration

Domain randomization is a powerful sim-to-real technique and we also adopt it for deployment. However, bipedal motions are inherently unstable and are sensitive to physical parameters. The RL policy would fail to find a solution that fits all physical parameters (see Sec. V-C) if too much randomization is applied. To ensure good transfer performance with only slight randomization, we conduct real-to-sim calibration that searches for the simulation parameters that can best explain the real robot trajectories.

Specifically, we apply the same sequence of actions both in simulation and on the real robot for the calibration. The sequence used to probe the real world is from rollouts of policies trained in the early development stage of this project and lasts for 120 seconds in total. Since the policies trained from an uncalibrated simulator can hardly succeed in the real world, we choose to hang up the robot and run the sequence in an open loop during calibration to avoid hardware damage, and save the joint position readings $\mathbf{q}_i^{\text{real}}|_{i=0}^N$ from 12 motors at 200 Hz. Given a specific configuration of simulation parameters ξ, we fix the base link in simulation and collect joint positions $\mathbf{q}_i^{\text{sim}}(\xi)|_{i=0}^N$ by applying the same action sequence. We spawn 8192 simulation environments and search for the parameters with minimal discrepancy to the real world as

$$\xi^\star = \arg\min_\xi \sum_{i=0}^N |\mathbf{q}_i^{\text{sim}}(\xi) - \mathbf{q}_i^{\text{real}}|^2. \quad (1)$$

We calibrate joint friction, joint damping, limb mass, and system delay in this process. Other parameters that are not probed (e.g., those related to contacts) are randomized naively. All randomized physical parameters and their ranges are listed in Table I. We remark that our calibration process is more end-to-end and does not require additional instruments compared to fitting a single module such as motors in [40].

C. Generating reference motions for human-like maneuvers

We study the generation of human-like motions for a quadrupedal robot from two modalities: one is to mimic human videos, and another is to convert natural language instructions into motions with a large language model (LLM).

For mimicking human videos, we focus on retargeting the human front limb motions to the robot. We first adopt an off-the-shelf human pose detector [46], [47] to estimate the 3D skeleton and obtain the vectors of wrists relative to the body p_{human} for each frame extracted from the video. The desired hand positions for the robot p_{robot} are then calculated by scaling down p_{human} to compensate for the differences in the size and working space between the human and the robot.

TABLE I: Randomized simulation parameters and ranges.

Name	Range
Joint friction	[0.03, 0.08]
Joint damping	[0.02, 0.06]
Rigid body friction	[1.0, 3.0]
Rigid body restitution	[0.0, 0.4]
Base mass offset	[-0.5kg, 0.5kg]
Hip mass offset	[0kg, 0.1kg]
Thigh mass offset	[-0.05kg, 0.05kg]
Calf mass offset	[-0.05kg, 0.05kg]
Foot mass offset	[0kg, 0.01kg]
Center of mass displacement	[-0.01m, 0.01m]
K_p, K_d	[80%, 120%]
Delay	[0.005s, 0.03s]

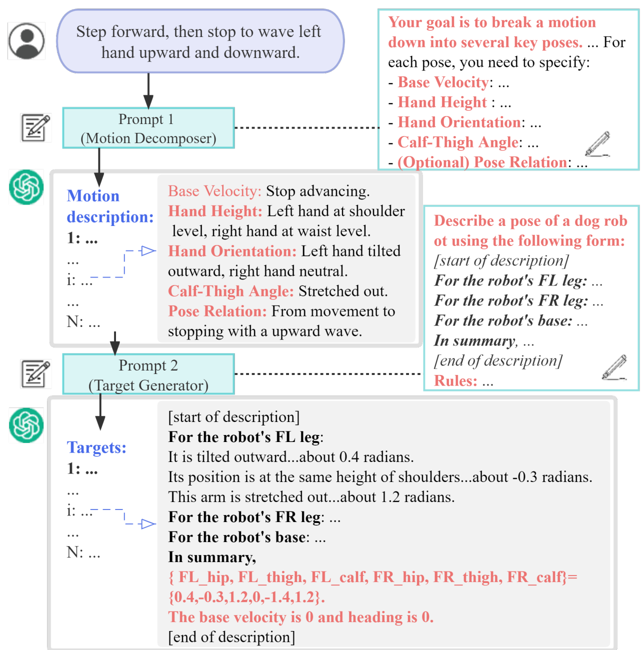

Fig. 2: The workflow of generating reference motions from human language instructions with an LLM. The language command from the user is first decomposed into a sequence of motion descriptions, then converted to targets consisting of base velocity, heading, and front limb joint positions. The example outputs by the LLM in both steps are in grey boxes, and the prompts we use are in cyan boxes.

As for the language input, we leverage the common sense knowledge of a pretrained LLM to generate reference motions that can fulfill the natural language instruction. As illustrated in Fig. 2, the generation is done with two rounds of conversation. In the first round, we prompt the LLM to decompose an abstract instruction into a sequence of key frames and to give detailed descriptions for each frame along the axis of the base velocity, hand height, hand orientation, calf-thigh joint, and the relationship with the previous frame. In the second round, the LLM is prompted to format the description of each frame as the precise target motion for the robot, including the base velocity, the heading direction, and the joint positions of 6 motors on the front limbs. The joint positions are finally converted to positions of end effectors with forward kinematics.

Since the LLM has limited domain knowledge concerning the specification of our quadrupedal robot, it is challenging to directly generate precise desired motions. To help the LLM generate reasonable values, we provide it with a set of rules specific to the kinematics of the robot in the prompt similar to [48], [49], such as "if the FL hand is tilted outward, then set FL_hip_joint within range (0.1,0.57) radians".

V. Experiments

We conduct experiments with a quadrupedal robot Xiaomi CyberDog2 [11]. The robot is driven by 12 identical CyberGear [50] motors. All computation in deployment is run on the internal board of the robot to reduce the communication latency.

A. Performance of the motion-conditioned policy

We test whether the control policy trained with RL can accomplish bipedal locomotion tasks with manually written desired motions. When commanded to track zero velocity and zero relative heading direction, the robot stands upright from the initial lying pose within 1 second as illustrated in the left half of Fig. 3. Fig. 4a shows the robot stably walks forward and backward following the target linear velocity $v_x = \pm 0.3$m/s after standing up. In Fig. 4b, we set the desired heading direction to 90 and -60 degrees relative to the initial pose, and the policy effectively controls the robot to turn around following these commands. After each bipedal motion, the robot is controlled by the same sit-down RL policy to settle down back to a resting pose with four legs on the ground (see the right half of Fig. 3).

We visualize the tracking performance of front toes in simulation during segments of "waving hand" (left) and "ballet dance" (right) in Fig. 5. The intended positions (blue) and the achieved positions (orange) are closely aligned, showcasing the effectiveness of front-toe tracking.

Results of real-to-sim calibration: As is described in Sec. IV-B, we conduct real-to-sim calibration to reduce the discrepancy between the simulator and the real robot. In the left plot of Fig. 6, we show the relation between simulated joint frictions and the sim-to-real discrepancy in joint positions per control step averaged over 120s of the open-loop trajectory. The sim-to-real gap is the smallest in our calibrated joint friction range, and the error goes up in both directions out of this range. In the right plot, we visualize the positions of three motors on the rear left leg in simulation and the corresponding real robot data recorded during open-loop control. The trajectory generated from the best-calibrated physical parameters (purple) matches the real data (red) much better than the uncalibrated one (blue).

B. Performing human-like motions when combined with generated motion targets

We then command the low-level policy with target commands parsed from human videos or natural languages and verify whether our system can enable human-like motions on the quadruped robot. We invite two human participants to

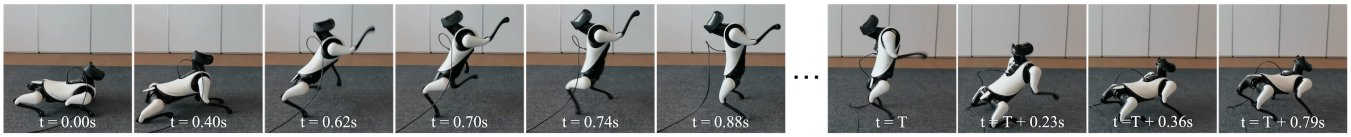

Fig. 3: Our RL policy brings up the quadrupedal robot from a lying pose to a stabilized bipedal standing pose. The separate sit-down policy then controls the robot from the upright standing pose to settle down with four legs on the ground. The learned policies demonstrate great agility, using less than 1 second to stand up and sit down, while are sufficiently robust to work on the real robot.

(a) Track different linear velocities of the robot base.

(b) Track different heading directions of the robot base.

Fig. 4: The quadrupedal robot demonstrates bipedal locomotion following target linear velocities $v_x = \pm 0.3$m/s to walk forward or backward, and tracking target heading directions 90 degrees to the left or 60 degrees to the right.

TABLE II: Ablation studies on the key designs for training our RL policy. The mean and standard deviation over three seeds are reported for each variant. Real2sim calibration, the dynamic scaling factor in the tracking reward, and the feet regularization reward all contribute to achieving the best balance over the stand-up performance, the tracking accuracy, and the feet clearance.

Method	$r^{\text{track}}_{\text{base\_v}}$	$r^{\text{track}}_{\text{heading}}$	$r^{\text{track}}_{\text{hand}}$	r^{height}	$r^{\text{reg}}_{\text{slip}}$	Episodic reward	Episode length
Ours	**0.221±0.022**	**0.173±0.009**	**0.679±0.010**	**0.389±0.005**	**-0.122±0.005**	**30.53±0.58**	**826.68±12.01**
w/o real2sim	0.080±0.013	0.074±0.013	0.318±0.059	0.194±0.036	-0.128±0.020	12.10±2.35	413.14±73.35
w/o dynamic scale	0.177±0.058	0.137±0.050	0.378±0.327	0.300±0.098	-0.126±0.027	21.92±9.43	767.41±106.08
w/o feet reg.	0.211±0.040	0.153±0.007	0.612±0.012	0.350±0.014	-0.207±0.076	23.82±1.59	749.31±20.00

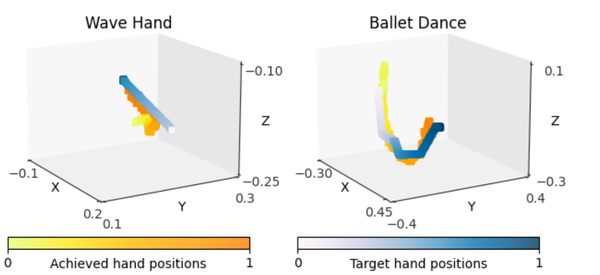

Fig. 5: The visualization of front-toe tracking during segments of "wave hand" (left) and "ballet dance" (right) in simulation. Blue dots represent the desired positions and the orange dots represent the achieved positions. The varying hues indicate the progression of time.

provide motion clips and map their hand trajectories to the quadruped as described in Sec. IV-C. Fig. 1 demonstrates our robot mimicking fast boxing motion. The commands are extracted every 0.1s from the video. The robot manages to keep balance while performing punches and an uppercut at high speed. The human reference frames annotated with detected skeletons are illustrated in the front row, and the robot execution trajectories are in the bottom row. Fig. 7 shows a slow ballet dance that lasts for more than 10 seconds. The motions are extracted every 0.5s. The robot follows the human guidance to gracefully lift up both hands from its waist to above its head, then drops down the right arm and opens it up with its best effort, and finally moves the left hand to the side. The robot does open its arm to the same extent as the human due to the position limit of its hip joints.

We also try to generate motion sequences to fulfill bipedal motions with an LLM. We prompt GPT-3.5 [51] to decompose a natural language instruction into a sequence of target base velocity, heading direction, and front limb joint positions, then convert joint positions to hand positions with forward kinematics. Fig. 8 shows an example of using LLM-generated commands to follow a language instruction "step forward, then stop to wave left hand upward and downward".

Finally, we show an interesting example of physical interaction in Fig. 9 where a human holds the lifted front limb of the standing quadruped and physically guides its position. Our policy is sufficiently robust to keep balanced at all times.

C. Ablation studies

We compare our method with the following variants: (a) w/o real2sim calibration, which randomizes the joint friction according to the value provided in the URDF to illustrate an uncalibrated simulation environment; (b) replacing the dynamic scales $c_x(s)$ in the tracking reward terms that

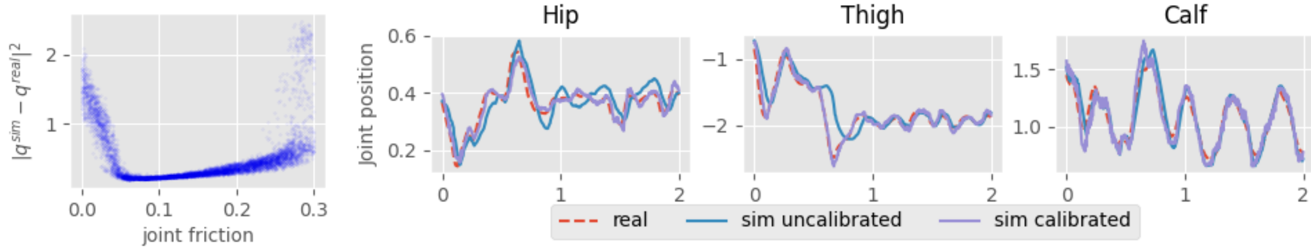

Fig. 6: The results of real-to-sim calibration. Left: the relationship between the sim-to-real difference and the simulated joint friction. The calibrated randomization range is chosen in the region with low errors. Right: the real motor positions w.r.t. the time on the rear left leg and the simulated ones using uncalibrated and calibrated physical parameters. The simulated trajectory matches the real world better after real-to-sim calibration.

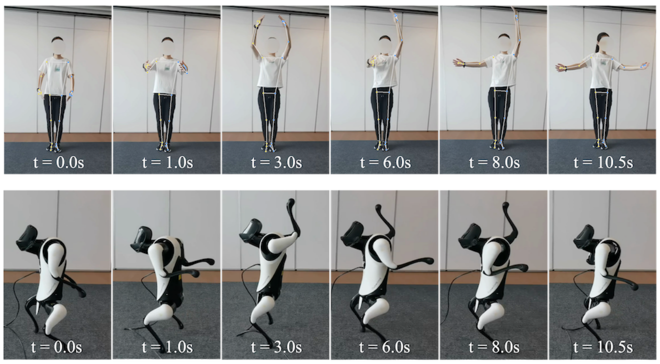

Fig. 7: The quadruped robot follows a human to perform ballet. It keeps balanced and tracks the hand poses at best effort during the long motion that lasts for more than 10s.

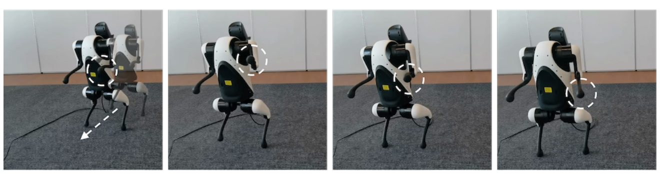

Fig. 8: Using LLM to generate motions for walking forward a few steps and then waving the left hand up and down.

are conditioned on the standing up performance with static coefficients 1; (c) removing the reward terms $r_{\text{gait}}^{\text{reg}}$ and $r_{\text{slip}}^{\text{reg}}$ that regulate the gait of rear feet. All variants are trained for three seeds. Each run is evaluated for 50 episodes in the same calibrated environment using the checkpoint trained for 18000 iterations. The performances are measured using both overall episodic metrics and detailed rewards regarding the base height, the tracking error, and the feet clearance.

As is shown in Table II, using an uncalibrated joint friction range [0, 0.2] (the second row) significantly degrades the performance in all metrics compared with the calibrated

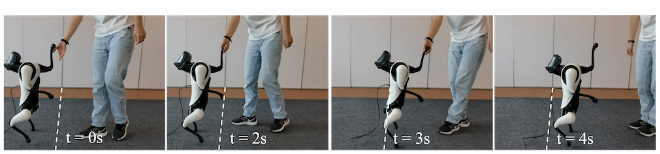

Fig. 9: A human walks hand-in-hand with the standing quadruped robot and drags its front leg to change its position.

Fig. 10: The feet heights recorded along the execution of different policies in simulation. Darker colors indicate the lower heights. Top: The left and right foot heights of the policy trained with our feet regularization reward terms, which demonstrates nice feet clearance. Bottom: A policy trained without these regularization terms tends to slip on the ground and can hardly be transferred to the real robot.

range of [0.03, 0.08] (the first row), indicating the necessity of an appropriate randomization range. The variant "w/o dynamic scale" also performs worse than our original version. It results in a lower base height since the robot finds a sub-optimal strategy that "sits" on the hind calf. The tracking reward and the base height of the policy trained without the gait regularization are comparable to the main result, but the feet slippery metric is much worse. We also compare the foot heights of the policies trained with or without the gait regularization in Fig. 10. Without the regualization, the robot only lifts up the hind legs slightly above the ground and demonstrates irregular contact patterns, thus is difficult to be deployed in the real world where the materials of the ground and the feet are not perfectly rigid bodies.

VI. Conclusion

We study the problem of enabling a quadrupedal robot to perform agile human-like bipedal motions and propose a bi-level framework. The low level is a motion-conditioned RL policy that tracks the desired states of the robot base and the front limbs while balancing on hind toes. At the high level, we generate human-like motion sequences to command the low-level policy from human videos or natural language instructions. Currently, we consider motions that are feasible with proprioceptive states only. Augmenting the robot with environmental perception to perform more complex interactions with humans and objects is an interesting future work.

References

[1] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, "The development of honda humanoid robot," in *Proceedings. 1998 IEEE international conference on robotics and automation (Cat. No. 98CH36146)*, vol. 2. IEEE, 1998, pp. 1321–1326.

[2] S. Shamsuddin, L. I. Ismail, H. Yussof, N. I. Zahari, S. Bahari, H. Hashim, and A. Jaffar, "Humanoid robot nao: Review of control and motion exploration," in *2011 IEEE international conference on Control System, Computing and Engineering*. IEEE, 2011, pp. 511–516.

[3] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous robots*, vol. 40, pp. 429–455, 2016.

[4] S. Shigemi, "Asimo and humanoid robot research at honda."

[5] O. Stasse, T. Flayols, R. Budhiraja, K. Giraud-Esclasse, J. Carpentier, J. Mirabel, A. Del Prete, P. Souères, N. Mansard, F. Lamiraux *et al.*, "Talos: A new humanoid research platform targeted for industrial applications," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 689–695.

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

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

[8] Y. Ji, Z. Li, Y. Sun, X. B. Peng, S. Levine, G. Berseth, and K. 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)*. IEEE, 2022, pp. 1479–1486.

[9] C. Yu and A. Rosendo, "Multi-modal legged locomotion framework with automated residual reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 312–10 319, 2022.

[10] S. Xu, H. Wang, J. Gao, Y. Ouyang, C. Yu, and Y. Wu, "Language-guided generation of physically realistic robot motion and control," *arXiv preprint arXiv:2306.10518*, 2023.

[11] Xiaomi, "Cyberdog2," https://www.mi.com/cyberdog2, 2023, accessed: Aug. 2023.

[12] G. Bellegarda and Q. Nguyen, "Robust quadruped jumping via deep reinforcement learning," *arXiv preprint arXiv:2011.07089*, 2020.

[13] G. B. Margolis, T. Chen, K. Paigwar, X. Fu, D. Kim, S. bae Kim, and P. Agrawal, "Learning to jump from pixels," in *Conference on Robot Learning*. PMLR, 2022, pp. 1025–1034.

[14] H.-W. Park, P. M. Wensing, and S. Kim, "Jumping over obstacles with mit cheetah 2," *Robotics and Autonomous Systems*, vol. 136, p. 103703, 2021.

[15] ——, "Online planning for autonomous running jumps over obstacles in high-speed quadrupeds," in *2015 Robotics: Science and Systems Conference, RSS 2015*. MIT Press Journals, 2015.

[16] L. Smith, J. C. Kew, T. Li, L. Luu, X. B. Peng, S. Ha, J. Tan, and S. Levine, "Learning and adapting agile locomotion skills by transferring experience," *arXiv preprint arXiv:2304.09834*, 2023.

[17] N. Rudin, H. Kolvenbach, V. Tsounis, and M. Hutter, "Cat-like jumping and landing of legged robots in low gravity using deep reinforcement learning," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 317–328, 2021.

[18] X. Huang, Z. Li, Y. Xiang, Y. Ni, Y. Chi, Y. Li, L. Yang, X. B. Peng, and K. Sreenath, "Creating a dynamic quadrupedal robotic goalkeeper with reinforcement learning," 2022.

[19] Y. Fuchioka, Z. Xie, and M. Van de Panne, "Opt-mimic: Imitation of optimized trajectories for dynamic quadruped behaviors," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5092–5098.

[20] E. Vollenweider, M. Bjelonic, V. Klemm, N. Rudin, J. Lee, and M. Hutter, "Advanced skills through multiple adversarial motion priors in reinforcement learning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5120–5126.

[21] X. B. Peng, A. Kanazawa, J. Malik, P. Abbeel, and S. Levine, "Sfv: Reinforcement learning of physical skills from videos," *ACM Trans. Graph.*, vol. 37, no. 6, Nov. 2018.

[22] X. B. Peng, E. Coumans, T. Zhang, T.-W. E. Lee, J. Tan, and S. Levine, "Learning agile robotic locomotion skills by imitating animals," in *Robotics: Science and Systems*, 07 2020.

[23] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa, "Amp: Adversarial motion priors for stylized physics-based character control," *ACM Trans. Graph.*, vol. 40, no. 4, Jul. 2021. [Online]. Available: http://doi.acm.org/10.1145/3450626.3459670

[24] C. Li, M. Vlastelica, S. Blaes, J. Frey, F. Grimminger, and G. Martius, "Learning agile skills via adversarial imitation of rough partial demonstrations," in *Conference on Robot Learning*. PMLR, 2023, pp. 342–352.

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

[26] E. Vollenweider, M. Bjelonic, V. Klemm, N. Rudin, J. Lee, and M. Hutter, "Advanced skills through multiple adversarial motion priors in reinforcement learning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 5120–5126.

[27] G. B. Margolis and P. Agrawal, "Walk these ways: Tuning robot control for generalization with multiplicity of behavior," in *Conference on Robot Learning*. PMLR, 2023, pp. 22–31.

[28] E. Daneshmand, M. Khadiv, F. Grimminger, and L. Righetti, "Variable horizon mpc with swing foot dynamics for bipedal walking control," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2349–2356, 2021.

[29] E. Dantec, M. Naveau, P. Fernbach, N. Villa, G. Saurel, O. Stasse, M. Taix, and N. Mansard, "Whole-body model predictive control for biped locomotion on a torque-controlled humanoid robot," in *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*. IEEE, 2022, pp. 638–644.

[30] T. Apgar, P. Clary, K. Green, A. Fern, and J. W. Hurst, "Fast online trajectory optimization for the bipedal robot cassie." in *Robotics: Science and Systems*, vol. 101. Pittsburgh, Pennsylvania, USA, 2018, p. 14.

[31] A. Hereid, O. Harib, R. Hartley, Y. Gong, and J. W. Grizzle, "Rapid trajectory optimization using c-frost with illustration on a cassie-series dynamic walking biped," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 4722–4729.

[32] Z. Li, X. Cheng, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Reinforcement learning for robust parameterized locomotion control of bipedal robots," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 2811–2817.

[33] Z. Li, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Robust and versatile bipedal jumping control through multi-task reinforcement learning," *arXiv preprint arXiv:2302.09450*, 2023.

[34] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 23–30.

[35] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 3803–3810.

[36] B. Mehta, M. Diaz, F. Golemo, C. J. Pal, and L. Paull, "Active domain randomization," in *Conference on Robot Learning*. PMLR, 2020, pp. 1162–1176.

[37] Q. Vuong, S. Vikram, H. Su, S. Gao, and H. I. Christensen, "How to pick the domain randomization parameters for sim-to-real transfer of reinforcement learning policies?" *arXiv preprint arXiv:1903.11774*, 2019.

[38] L. Ljung, "System identification," in *Signal analysis and prediction*. Springer, 1998, pp. 163–173.

[39] S. Kolev and E. Todorov, "Physically consistent state estimation and system identification for contacts," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 1036–1043.

[40] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019.

[41] Y. Chebotar, A. Handa, V. Makoviychuk, M. Macklin, J. Issac, N. Ratliff, and D. Fox, "Closing the sim-to-real loop: Adapting simulation randomization with real world experience," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8973–8979.

[42] J. Tan, Z. Xie, B. Boots, and C. K. Liu, "Simulation-based design of dynamic controllers for humanoid balancing," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 2729–2736.

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

[44] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa *et al.*, "Isaac gym: High performance gpu based physics simulation for robot learning," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

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

[46] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, "Blazepose: On-device real-time body pose tracking," *arXiv preprint arXiv:2006.10204*, 2020.

[47] H. Xu, E. G. Bazavan, A. Zanfir, W. T. Freeman, R. Sukthankar, and C. Sminchisescu, "Ghum & ghuml: Generative 3d human shape and articulated pose models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6184–6193.

[48] W. Yu, N. Gileadi, C. Fu, S. Kirmani, K.-H. Lee, M. G. Arenas, H.-T. L. Chiang, T. Erez, L. Hasenclever, J. Humplik *et al.*, "Language to rewards for robotic skill synthesis," *arXiv preprint arXiv:2306.08647*, 2023.

[49] Y. Tang, W. Yu, J. Tan, H. Zen, A. Faust, and T. Harada, "Saytap: Language to quadrupedal locomotion," *arXiv preprint arXiv:2306.07580*, 2023.

[50] Xiaomi, "Cybergear," https://www.mi.com/shop/buy/detail?product_id=19086, 2023, accessed: Aug. 2023.

[51] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27 730–27 744, 2022.