2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

DribbleBot: Dynamic Legged Manipulation in the Wild

Yandong Ji*, Gabriel B. Margolis*, and Pulkit Agrawal

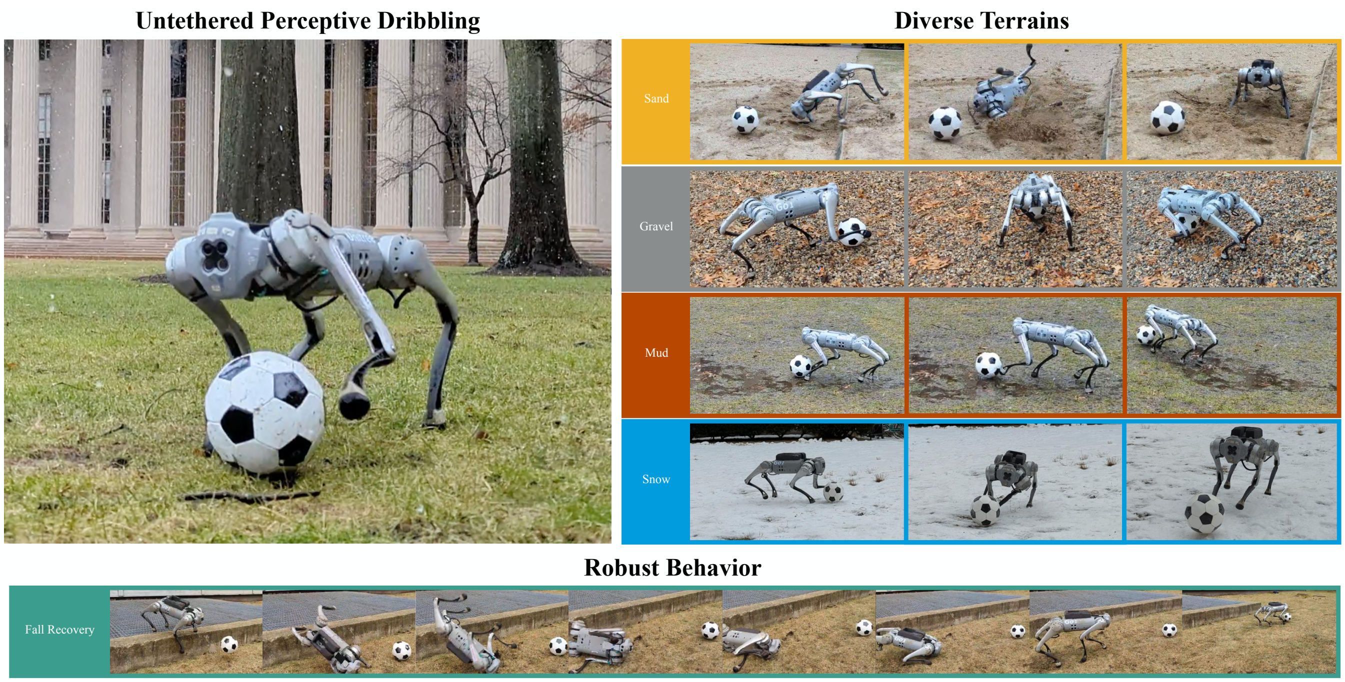


Fig. 1: ***In-the-wild*** **dribbling**. Sim-to-real reinforcement learning enables dynamic soccer ball manipulation in the wild. *Untethered Perceptive Dribbling:* Our system operates using onboard sensors and compute that can go anywhere. *Diverse Terrains:* A sim-to-real control policy adapts to the impact of different terrains on the ball and robot. *Robust Behavior:* A combination of engineered and emergent behaviors aids ball reacquisition after a loss of control.

***Abstract*— DribbleBot (Dexterous Ball Manipulation with a Legged Robot) is a legged robotic system that can dribble a soccer ball under the same real-world conditions as humans. We identify key challenges of in-the-wild soccer ball manipulation, including variable ball motion dynamics and perception using body-mounted cameras. To overcome these challenges, we propose a domain and task specification for learning viable soccer dribbling behaviors in simulation that transfer to real fields. Our system provides promising evidence that current legged robots are physically capable and adequately sensorized for varied and dynamic real-world soccer play. Video is available at `https://gmargo11.github.io/dribblebot`.**

I. INTRODUCTION

Training control policies using reinforcement learning in simulation and deploying them zero-shot in the real world has recently enabled fast, robust legged locomotion across challenging terrains like stairs, hiking trails, sand, and mud [1]–[10]. The same approach has also succeeded at dynamic object manipulation using dexterous hands [11]–[14]. One natural question is whether the successes of sim-to-real reinforcement learning can be extended to compound tasks that combine perception, dynamic locomotion, and object manipulation. This class of tasks contains a number of practical robotics applications in search and rescue, delivery, and sport. As one example, Boston Dynamics has recently developed impressive demonstrations using the humanoid robot Atlas to pick up, run with, and throw heavy objects on a controlled construction site [15]. Despite its relevance, dynamic mobile manipulation is understudied in academic literature due to the typical requirement of expensive, specialized hardware and complex system architecture.

As a case study in dynamic mobile manipulation, we consider the task of soccer dribbling in the wild. Previous studies on robot soccer [16]–[22] assumed a restricted setting: (i) the playing surface was flat and smooth; (ii) external perception was used instead of onboard perception and (iii) interactions largely consisted of static dribbling, where the ball comes to rest before each kick. In contrast, like a human athlete, our system operates from onboard perception and can dynamically control a ball across a wide variety of natural terrains including grass, mud, snow, and pavement.

When training motor control policies for robots, it is critical to choose a task specification that faithfully describes the desired behavior, is feasible to optimize, and promotes move-

* Indicates equal contribution.
All authors are with the Improbable AI Lab, Massachusetts Institute of Technology, USA. Correspondence to: {yandong, gmargo}@mit.edu

ments suitable for sim-to-real transfer. In locomotion [1]–[10] and dexterous manipulation [11]–[14], researchers have gradually developed the effective task specification by way of reward functions, curriculum strategies, choice of observation space, and distribution of training environments. We must adapt these choices to account for the unique considerations of in-the-wild soccer dribbling. One consideration is the ball-terrain dynamics, which varies independently from the robot-terrain dynamics due to the much lighter ball mass and the different nature of rolling contact. We implement a custom ball drag model that results in robustness to different terrains. Another consideration is the limited precision and range of onboard perception systems. We perform object detection directly on raw wide-angle fisheye images to track the ball across a useful dribbling workspace. An additional challenge is potential loss of ball control due to locomotion failure. To address this, we integrate a recovery policy to stand up autonomously after flipping and rely on emergent behavior of the dribbling controller to regain ball control.

The resulting system, DribbleBot (Dexterous Ball Manipulation with a Legged Robot), demonstrates dynamic real-world dribbling maneuvers across a variety of terrains. By providing evidence that existing hardware and sensors are capable of successful behavior, we hope to motivate more work in this promising direction.

II. MATERIALS

Hardware: We use the Unitree Go1 robot for all experiments [23]. This small robot quadruped stands $40\,\mathrm{cm}$ tall. We use two onboard $210°$ field-of-view fisheye cameras to capture images, one facing forward and one facing downward. All computation is performed on two onboard NVIDIA Jetson Xavier NX units. Due to the computation, communication bandwidth, and electrical power limitations of the robot, we critically process full-resolution images locally on each board and send only the ball location estimates to the policy board. We accelerate perception inference using TensorRT. This allows the system to process 400×480 resolution images comfortably at $30\,\mathrm{Hz}$.

Simulator: We simulate the Unitree Go1 robot in Isaac Gym [24] using the manufacturer-provided URDF model. Simulation and training run on a single NVIDIA RTX 3090.

Pretrained Perception Module: We obtain the YOLOv7 [25] model weights from the internet, pretrained on the COCO dataset [26], to perform our own fine-tuning as described in section III-B.1.

III. METHOD

Overview: We train control policy $\mathbf{a}_t = \pi_d(\mathbf{o}_t, \mathbf{c}_t)$ in simulation and transfer it to the real world. The observation $\mathbf{o}_t$ consists of the proprioceptive sensory data and the ball position $\mathbf{b}_t$. The ball velocity command $\mathbf{c}_t$ is represented in the global reference frame. We choose not to train our policy end-to-end on camera images, because of the numerous challenges including slow simulation speed, poor sample efficiency, and increased sim-to-real gap. Instead, we allow the policy to directly observe the ball position during training, and for deployment, we implement a separate perception module $\hat{\mathbf{b}}_t = \mathbf{Y}(\mathbf{o}_t^v)$ fine-tuned on labeled real-world images. The policy π_d outputs actions $\mathbf{a}_t$ which are the 12 joint position targets and updates at $50\,\mathrm{Hz}$. It is trained using Proximal Policy Optimization (PPO) [27] to maximize a shaped reward function incentivizing well-formed dribbling.

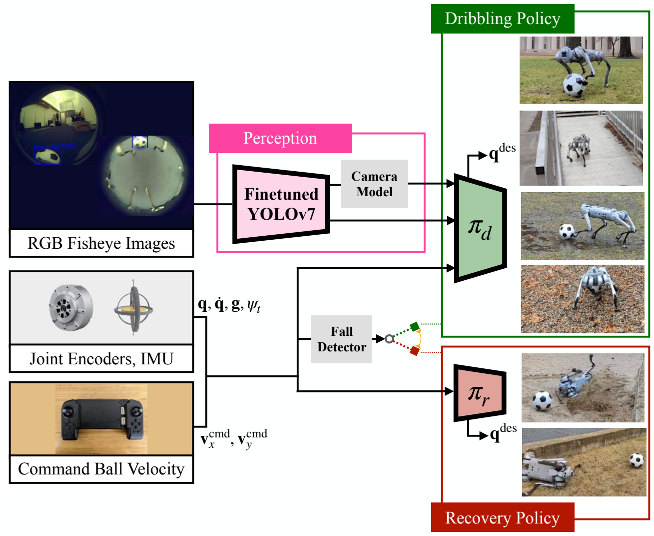


Fig. 2: **System architecture for DribbleBot.**

A sim-to-real gap can arise in training this controller due to several unique considerations in the soccer dribbling task: (i) mismodeling of the ball-terrain dynamics, (ii) limited perception range and precision, and (iii) locomotion failure due to extreme perturbations. Regarding the ball-terrain dynamics, we implement a custom ball drag model and perturb the ball randomly during training. For the perception challenge, we process raw ultra-wide fisheye images by finetuning an object detection network, and introduce reward shaping to incentivize keeping the object in view. To address locomotion failure, we implement a finite state machine that activates a recovery policy π_r to stand up autonomously from a fall.

A. Training the Dribbling Policy

1) Environment Design: We simulate the physics of robot, ball, and terrain. In each environment, the robot's yaw is initialized uniformly at random, and its initial leg positions are randomized around a nominal pose. A soccer ball is initialized at a random position within $2\,\mathrm{m}$ of the robot. The target ball velocity in the global frame is also uniformly randomized. These considerations ensure that the robot learns omnidirectional locomotion and dribbling skills while also being aware of its own global orientation. The episode length is $40\,\mathrm{s}$ with the controller running at $50\,\mathrm{Hz}$.

2) Control Interface for Dribbling in the Wild: Previous works in sim-to-real legged locomotion have mostly chosen the body velocity command or position [28] as the control interface, with some extending the user's options to include multiple gaits [8], [29] or foot placements [30]. However,

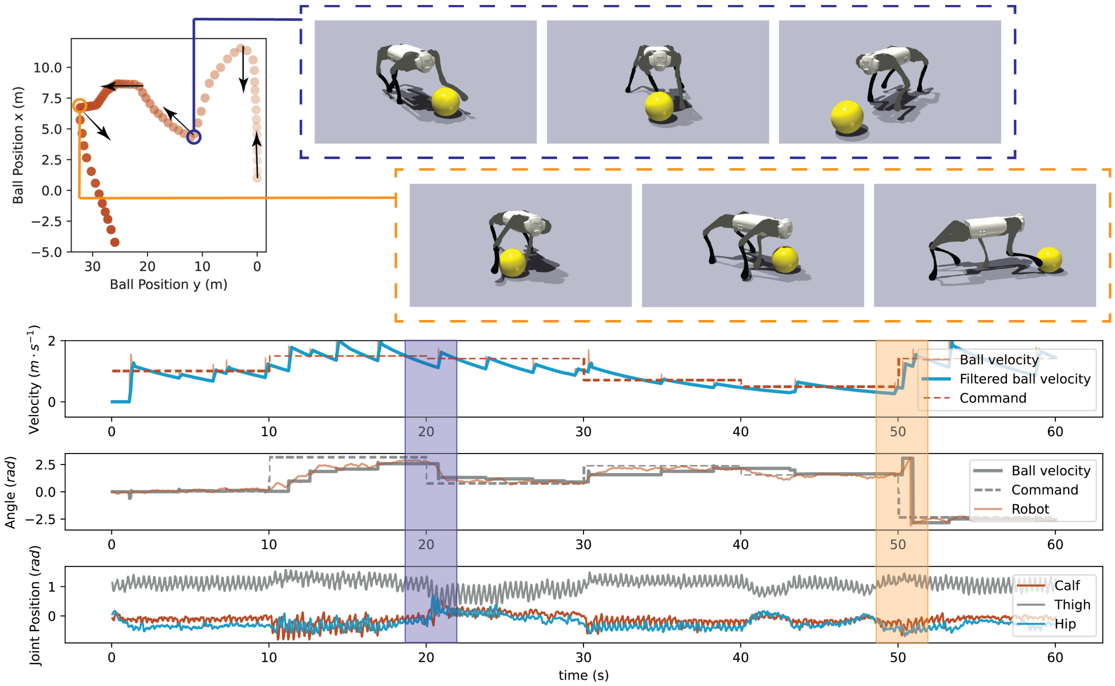


Fig. 3: **Simulated dribbling performance evaluation.** *Top:* Ball position in world frame and turning moment snapshots. The red points indicate the ball position and darken as time elapses. The arrow represents the approximate direction of the commanded velocity. *Upper and middle bottom:* Ball velocity tracking performance in polar coordinates. Here, the robot is first commanded to dribble the ball forward at $1\,\mathrm{m/s}$, and then to execute a sequence of turns at various speeds. *Lower bottom:* We illustrate the joint position of the front right leg, which is typically used for dribbling and for executing left turns. The blue highlight around $20\,\mathrm{s}$ corresponds to the left turn visualized in the first row of images above. The orange highlight around $50\,\mathrm{s}$ corresponds to the right turn visualized in the second row.

neither body-level control nor detailed gait parameterization provides a suitable high-level interface to specify ball dribbling. In addition to adjusting the leg swings to apply targeted forces, the robot needs to orient and position itself relative to a moving ball, shifting the local frame and modulating velocity during dribbling in a way that is not easily prescribed in terms of standard locomotion behaviors. Instead, in our dribbling policy, the command directly specifies the ball linear velocity in the x-y plane. Importantly, we express the command ball velocity in the global reference frame. Because the ball has full rotational symmetry and the robot's orientation can vary rapidly during a kicking maneuver, the local reference frame of each body is constantly changing and less useful for a human operator.

3) Observation and Action Space: The full input to the policy π_d is a 15-step history of command $\mathbf{c}_t$, ball position $\mathbf{b}_t$, joint positions and velocities $\mathbf{q}_t, \dot{\mathbf{q}}_t$, gravity unit vector in the body frame $\mathbf{g}_t$, global body yaw $\boldsymbol{\psi}_t$, and timing reference variables $\boldsymbol{\theta}_t^{\text{cmd}}$ as defined in in [8]. The commands $\mathbf{c}_t$ consist of the target ball velocities $\mathbf{v}_x^{\text{cmd}}, \mathbf{v}_y^{\text{cmd}}$ in world frame. Therefore, the global body yaw observation is necessary for the robot to follow a world frame ball velocity command, as it provides the global transform from the original orientation. The action space $\mathbf{a}_t$ is the twelve target joint positions that are tracked using a PD controller with $k_p = 20.0$, $k_d = 0.5$.

4) Reward Model: Table II (Appendix) provides the task reward terms used for learning agile ball manipulation skills. The task is expressed as tracking a desired ball velocity in the global reference frame. Heuristic reward terms incentivize the robot to be close to the ball and to move at the commanded velocity. To promote ball visibility in the camera, the robot is also rewarded for facing towards the ball. Another set of gait reward terms in the style of [8], [29] guide the robot to adopt a consistent ground contact schedule, generating well-formed gaits without the constraints of a full reference trajectory. Additional safety reward terms, which are standard in the related literature [7], [8], [31], [32], penalize dangerous commands and facilitate sim-to-real transfer.

5) Policy Architecture and Optimization: We use Proximal Policy Optimization (PPO) [27] to train our soccer dribbling policy, an MLP with the hidden layer size $[512, 256, 128]$. The policy converges after 7 billion timesteps, or about 48 hours of training. We also concurrently train a state estimator [32] in simulation using supervised learning where the input is the same as the policy and the hidden layer size is $[256, 128]$. From the observation history, this state estimator infers privileged information that is not available directly in the real world. In our setting, we predict the body velocity, ball velocity, and ball-terrain drag force coefficient as the privileged information.

B. Measures to Mitigate the Sim-to-Real Gap

1) Perception in Fisheye Images: To make dribbling feasible, the robot needs to localize a size-3 soccer ball (diameter $18\,\mathrm{cm}$) using a body-mounted camera when the ball is as close as $20\,\mathrm{cm}$ to the body. This harsh geometry mandates

Scenario	Full	-R	-Y	-D	
Tile	4/4	4/4	0/4	4/4	
Grass	4/4	4/4	0/4	4/4	
Sand	4/4	4/4	0/4	2/4	
Snow	3/4	3/4	0/4		
Curb Step-Down	2/4	1/4			
Ramp	0/4	0/4			

TABLE I: **Real-world dribbling performance evaluation.** The robot executes a fixed dribbling trajectory on each trial. We test each scenario with full system design (`Full`) and with ablations: No recovery controller (`-R`); No YOLO finetuning (`-Y`); No drag model during training (`-D`)

an extremely wide field of view. After testing with rectified images and common cameras like RealSense, we found the visible workspace of these typical options much too small for dribbling. Therefore, we opt to process the raw images from forward-facing and downward-facing fisheye cameras, each with ultra-wide field of view $210°$. We use a base perception model, YOLOv7 [25], pretrained on diverse images, which helps preserve ball detection even in cluttered environments. However, because YOLOv7 is pretrained on rectified images, its performance on our fisheye images is poor. To recover good performance in the entire field of view, we fine-tune the network on 254 hand-labeled images from our robot's camera, including images with the ball at the edge.

After detecting the bounding box of the ball in the camera image, we convert it to a relative pose in the robot body frame. We apply an approximate transformation using the equidistant fisheye lens model to convert the ball profile in pixel space to the 3D ball pose. Given the ball centered at pixel (p_x, p_y), we first compute the angle Ψ from the camera principal axis to the ball center using the equidistant model $r = f\Psi$. Next, observing the ball radius in the image plane and knowing the actual ball radius, we compute the perspective projection ratio and therefore the ball position in camera frame. The observed ball positions from front and bottom cameras are transformed to body frame and we select the one with a higher confidence value from YOLO.

2) Perception Noise Model: Our control policy is trained with observations of the ball position rather than high-dimensional raw camera images. However, in the real world, perceived ball position estimates are noisy. To ensure that the policy is robust to perception noise, we add noise sampled from a uniform distribution to ball position during training. Further, to emulate large changes to ball position that might happen due to a human kicking the ball or the ball going outside field of view, we also randomly teleport the ball in the ground plane. Finally, because the data rate of the camera is limited, we simulate camera communication delay. Noise model details are provided in Section IX (Appendix).

3) Robot System Identification: To mitigate the sim-to-real gap in robot dynamics, we employ two standard and effective system identification measures: (a) Train an actuator network on real-world torque data to account for the non-ideal motor dynamics [8], [14], [33]; and (b) Identify and model the lag between the time the observation is measured and the time the action is applied [8], [14], [34].

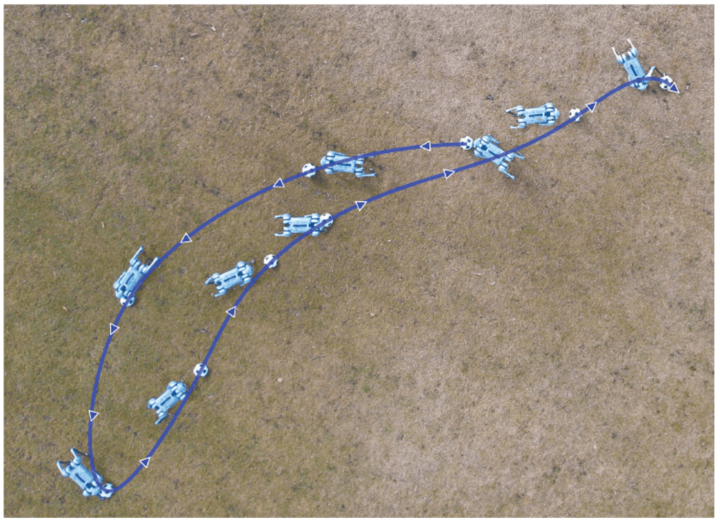
4) Ball-Terrain Interaction Model: Human soccer players can quickly adapt to variations in ball dynamics, due to physical factors like air pressure and size as well as external perturbations to the ball due to uneven terrain or opposing players. We embed similar robustness in DribbleBot by implementing a custom ball drag model and random perturbation scheme in our training environment. Our ball drag model applies drag force proportional to the square of the velocity: $F_D = C_D v^2$, following the standard equation for aerodynamic drag. Different values of C_D serve to emulate terrains with various resistance forces, such as a field with tall grass (high C_D) or pavement (low C_D). In addition, we randomize the ball mass and apply random changes in ball velocity during training. Ball velocity randomization simulates external perturbations to the ball such as human intervention or contact with uneven terrain. The randomization range details are given in Table IV (Appendix).

5) Fall Recovery Controller: If the robot encounters a harsh perturbation such as a shove or a steep curb that results in locomotion failure, it will also cause loss of ball control. In this scenario, we would like the robot to get up from its fall and resume dribbling. Similar to prior works [33], [35], we train a dedicated recovery policy that enables the robot to return to a standing position from diverse fall scenarios. We first generate a set of 1000 initial fall configurations by randomly dropping the robot from different orientations, then train a policy with rewards for body orientation, base height, and action smoothness. The details of the reward function and training procedure for the recovery policy are provided in Table III (Appendix). To transition between dribbling and recovery policy, we define a finite state machine with transitions based on the body orientation. When the roll or pitch angle is larger than $1.0\,\mathrm{rad}$, indicating locomotion failure, the recovery policy triggers to return to the standing pose. When the roll and pitch are smaller than $0.5\,\mathrm{rad}$, the dribbling policy is reactivated.

IV. RESULTS

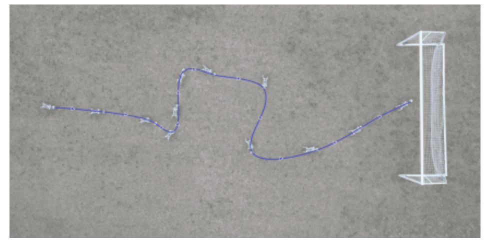
A. Simulation Performance

1) Dribbling Control: We initialize the robot at the origin, initialize the ball $1\,\mathrm{m}$ in front of the robot, and then input a sequence of velocity commands to evaluate the properties of the learned dribbling behavior. Figure 3 shows the performance visualization. The corresponding behavior is also shown in Video S1.



(a) 180° turn maneuver.

(b) Ball control on tile, gravel, and bumpy moss.



(c) Dribble to goal with an evasive turn.

Fig. 4: **Overhead images of DribbleBot during real-world deployment.**

We observe that the robot is able to track dribbling speeds up to $1.5\,\mathrm{m/s}$ and the entire range of turning angles up to 180 degrees. Unlike in ordinary locomotion, the task of turning the ball may extend across a long time horizon, during which the robot establishes control of the ball and executes multiple kicks. This incurs a delay between the change in command and the change in ball state, sometimes as long as several seconds. This lag between commanded and realized ball velocity is visible in Figure 3.

To illuminate the whole-body nature of dribbling and turning, we visualize the joint angles of the front right leg (Figure 3, bottom). The front right leg is used to kick the ball during a left turn (orange highlight), and its motion is substantially changed during this maneuver. Later, when the robot makes a right turn (blue highlight), the left leg is used to kick the ball, but the motion of the right leg also changes to stabilize the maneuver. Coordination of the entire body motion is leveraged to manipulate the ball.

B. Real-world Deployment

1) Dribbling on Diverse Terrains: We test our dribbling controller on a number of terrains with different dynamics. Figure 1 shows grass, mud, slippery gravel, sand terrain, snow, and a steep curb. The robot is able to execute dribbling and turning motions on each terrain under teleoperation, as shown in the supplementary video.

Table I reports the performance of the robot executing a fixed dribbling trajectory on each terrain. The robot is commanded with a predetermined trajectory: dribble forward at $1.5\,\mathrm{m/s}$ for $10\,\mathrm{s}$, then stop the ball for $5\,\mathrm{s}$, then return towards the starting line at $1.5\,\mathrm{m/s}$ until the line is reached. We count the trial as a failure if the robot loses control of the ball, although if the loss is due to the robot falling, we allow it to autonomously recover and continue the attempt. We test each scenario with full system design (`Full`) and with ablations: No recovery controller (`-R`); No YOLO finetuning (`-Y`); No custom ball drag model during training (`-D`). The robot executed four consecutive successful maneuvers using the full system design on tile, grass, and sand. Snow, step-down, and ramp are progressively more challenging and yield lower performance. While the ramp was traversable under teleoperation, it was not completed autonomously. Ablation results show that YOLO finetuning (`-Y`) is critical to performance, and recovery policy (`-R`) improves one run in the challenging curb environment. For the ball drag model, videos show that the policy without drag model (`-D`) maintains control on both grass and tile, but moves substantially less distance on grass despite the equal velocity command. On sand, the policy without drag model fails twice during the turning maneuver after missing a kick on the ball. This suggests that the additional robustness from the drag model may improve response to unexpected ball trajectories.

Figure 4 shows stitched overhead photos to illustrate the real-world dribbling performance. A human teleoperates the robot through (a) 180° turn, (b) a series of diverse terrains including tile, gravel, and bumpy moss, and (c) a $10\,\mathrm{m}$ run towards a soccer goal, with an evasive turning maneuver.

2) Playing with a Human and Emergent Behaviors: We explore the scenario where the robot interactively plays with a human partner on both grassy field and flat ground (Video B1). Unlike the typical locomotion task, the task of controlling ball velocity affords the robot a high degree of freedom in its behavior, even when the user is not changing the command. Successful dribbling is not monolithic: it often involves extended aperiodic movements to seek the ball, orient the body for a kick, and double back if the ball has been lost due to an unexpected perturbation, temporary perception failure, or control failure. We think this may be an interesting context in which to study human-robot interaction. While direct physical interaction with a legged robot is typically limited, interaction through the soccer ball as a shared medium proves rich and fun.

V. RELATED WORK

A. Soccer Skills for Legged Robots

Soccer has long been an area of interest for roboticists. The RoboCup competition, this year in its 26th season, has attracted thousands of annual participants. RoboCup teams have implemented effective rule-based approaches to kicking, passing, and shooting in the past [16]–[18].

Recently, some works have applied learning to legged ball manipulation tasks in simulation [19], [36] and in externally instrumented indoor settings [21], [22], [37]. [37] demonstrated that a quadruped lying on its back can control and reorient a ball with its legs. A number of soccer skills such as dribbling [38] and juggling [36] have been demonstrated for physically simulated characters using reinforcement learning. [20] used imitation learning to perform static dribbling in the real-world indoor setting assisted by motion capture. [21] applied a hierarchical framework to the soccer shooting task in the real world, selecting the front right foot Bézier curve parameters as the low-level command inputs and leveraging a real-world fine-tuning stage to improve the shooting accuracy. [22] trained a control policy for jumping to block an oncoming ball in an instrumented laboratory setting using sim-to-real reinforcement learning.

In addition to low-level skill learning, some work has focused on learning high-level soccer play end-to-end. Notably, [39] approaches the problems of muscle level control and long-horizon decision-making by first pretraining low-level skills using human soccer players' motion-capture video clips and then finding solutions for the multi-agent coordination goal in the low-level control space using reinforcement learning. To learn low-level skills like dribbling, [39] relies on motion capture data of human soccer players, which is not available for the quadruped form factor.

B. Dynamic Object Manipulation

Prior work has explored manipulating objects dynamically using a fixed or fully actuated base. [40] controlled a robotic arm to blindly perform ball juggling using an open-loop policy. Another work on robotic table tennis [41] estimated the ball state using an extended Kalman Filter which internally leverages a model of flight and bouncing behaviors. [42] learned a residue physics model to randomly pick up and throw a rigid object into a box. [43] bootstrapped a human behavior model and trained on both simulated and real data to learn a control policy for a table tennis-playing robot.

Another relevant line of work has investigated manipulating objects using a quadruped with mounted arm. [44] manipulated objects with a quadruped-mounted arm, coordinating the body and arm motion through learned estimation module. [45] implemented a model-based controller to manipulate objects with a quadruped-mounted arm in standing pose. [46] trained an end-to-end controller using reinforcement learning to perform coordinated manipulation with a quadruped-mounted arm under teleoperation and demonstrate vision-guided reaching using AprilTags. These works investigate complementary problems in the space of dynamic mobile manipulation.

VI. DISCUSSION OF LIMITATIONS

Our system has a number of limitations which we hope to explore and improve upon in future work. We enumerate several here, with videos of failure cases available on the project website. *Slow turning response*: our system can execute sharp turns of the ball, but there is lag between the command onset and the actual turn (Figure 3). *Perception sensitivity to lighting*: We found that the perception module can perform poorly in bright, direct sunlight that produces glare from reflection on the ball and cameras. Fine-tuning the perception network with a more diverse set of outdoor images may resolve this problem. *Imprecision at high speeds*: If the ball is accidentally kicked too fast on low-drag terrain, the attempt to stop the ball can fail. *Lack of geometry awareness*: While the robot can dribble on slippery and uneven terrains, it cannot traverse larger obstacles like steep slopes and staircases with good consistency. Moreover, it is not aware of objects in the environment like poles and walls. Future work could incorporate more information about the environment geometry into the controller to improve ball control in cluttered and harsh settings.

VII. CONCLUSION

We have presented DribbleBot, the first dynamic quadrupedal soccer dribbling controller that operates from onboard perception on diverse outdoor terrains. We proposed novel design choices for dealing with variable terrain-ball dynamics and making use of limited onboard perception. Varied motor skills emerge without reference trajectories to capture and control the ball given a simple ball velocity command. We conclude that quadrupedal soccer ball dribbling is a promising domain in which to study dynamic mobile manipulation. The soccer ball can also serve as an interface for playful interaction between legged robots and humans.

ACKNOWLEDGMENT

We thank the members of the Improbable AI lab for helpful discussions and feedback. We are grateful to MIT Supercloud and the Lincoln Laboratory Supercomputing Center for providing HPC resources. This research was supported by the DARPA Machine Common Sense Program, the MIT-IBM Watson AI Lab, and the National Science Foundation under Cooperative Agreement PHY-2019786 (The NSF AI Institute for Artificial Intelligence and Fundamental Interactions, http://iaifi.org/). This research was also sponsored by the United States Air Force Research Laboratory and the United States Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the United States Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation herein.

AUTHOR CONTRIBUTIONS

- **Yandong Ji** contributed to ideation and implementation of the entire system, experimental evaluation, and writing.
- **Gabriel B. Margolis** contributed to ideation and implementation of the entire system, experimental evaluation, and writing.
- **Pulkit Agrawal** advised the project and contributed to its development, experimental design, positioning, and writing.

REFERENCES

[1] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke, "Sim-to-real: Learning agile locomotion for quadruped robots," in *Proc. Robot.: Sci. and Syst. (RSS)*, Pittsburgh, Pennsylvania, USA, June 2018, pp. 1–9.

[2] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Sci. Robot.*, vol. 4, no. 26, p. aau5872, Jan. 2019.

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

[4] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "RMA: Rapid motor adaptation for legged robots," in *Proc. Robot.: Sci. and Syst. (RSS)*, Virtual, July 2021.

[5] J. Siekmann, K. Green, J. Warila, A. Fern, and J. Hurst, "Blind bipedal stair traversal via sim-to-real reinforcement learning," in *Proc. Robot.: Sci. and Syst. (RSS)*, Virtual, July 2021.

[6] T. Miki, J. Lee, J. Hwanbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Sci. Robot.*, vol. 7, no. 62, p. abk2822, Jan. 2022.

[7] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal, "Rapid locomotion via reinforcement learning," in *Proc. Robot.: Sci. and Syst. (RSS)*, June 2022.

[8] G. B. Margolis and P. Agrawal, "Walk these ways: Tuning robot control for generalization with multiplicity of behavior," in *Proc. Conf. Robot Learn. (CoRL)*, Auckland, New Zealand, Dec. 2022.

[9] A. Agarwal, A. Kumar, J. Malik, and D. Pathak, "Legged locomotion in challenging terrains using egocentric vision," in *Proc. Conf. Robot Learn. (CoRL)*, Auckland, New Zealand, Dec. 2022.

[10] S. Choi, G. Ji, J. Park, H. Kim, J. Mun, J. H. Lee, and J. Hwangbo, "Learning locomotion on deformable terrain," *Sci. Robot.*, vol. 8, no. 74, p. eade2256, 2023.

[11] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, *et al.*, "Learning dexterous in-hand manipulation," *Int. J. Robot. Res. (IJRR)*, vol. 39, no. 1, pp. 3–20, 2020.

[12] OpenAI, I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. Tezak, J. Tworek, P. Welinder, L. Weng, Q. Yuan, W. Zaremba, and L. Zhang, "Solving Rubik's cube with a robot hand," *arXiv preprint*, 2019.

[13] T. Chen, J. Xu, and P. Agrawal, "A system for general in-hand object re-orientation," in *Proc. Conf. Robot Learn. (CoRL)*, London, UK, Nov. 2021, pp. 297–307.

[14] T. Chen, M. Tippur, S. Wu, V. Kumar, E. Adelson, and P. Agrawal, "Visual dexterity: In-hand dexterous manipulation from depth," *arXiv preprint arXiv:2211.11744*, 2022.

[15] R. Deits and T. Koolen, "Picking up momentum," Jan 2023. [Online]. Available: https://www.bostondynamics.com/resources/blog/picking-momentum

[16] M. Veloso, W. Uther, M. Fijita, M. Asada, and H. Kitano, "Playing soccer with legged robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, vol. 1. IEEE, 1998, pp. 437–442.

[17] P. Stone, "Intelligent autonomous robotics: A robot soccer case study," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 1, no. 1, pp. 1–155, 2007.

[18] M. Friedmann, J. Kiener, S. Petters, D. Thomas, O. Von Stryk, and H. Sakamoto, "Versatile, high-quality motions and behavior control of a humanoid soccer robot," *International Journal of Humanoid Robotics*, vol. 5, no. 03, pp. 417–436, 2008.

[19] A. F. Muzio, M. R. Maximo, and T. Yoneyama, "Deep reinforcement learning for humanoid robot behaviors," *Journal of Intelligent & Robotic Systems*, vol. 105, no. 1, pp. 1–16, 2022.

[20] S. Bohez, S. Tunyasuvunakool, P. Brakel, F. Sadeghi, L. Hasenclever, Y. Tassa, E. Parisotto, J. Humplik, T. Haarnoja, R. Hafner, M. Wulfmeier, M. Neunert, B. Moran, N. Siegel, A. Huber, F. Romano, N. Batchelor, F. Casarini, J. Merel, R. Hadsell, and N. Heess, "Imitate and repurpose: Learning reusable robot movement skills from human and animal behaviors," *arXiv preprint arXiv:2203.17138*, 2022.

[21] 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," *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, Oct. 2022.

[22] 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," *arXiv preprint arXiv:2210.04435*, 2022.

[23] Unitree Robotics, A1, 2022, https://www.unitree.com/products/a1, [Online; accessed Apr. 2022].

[24] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, "Isaac Gym: High performance GPU-based physics simulation for robot learning," *arXiv preprint*, 2021.

[25] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[26] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.

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

[28] N. Rudin, D. Hoeller, M. Bjelonic, and M. Hutter, "Advanced Skills by Learning Locomotion and Local Navigation End-to-End," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 2497–2503.

[29] J. Siekmann, Y. Godse, A. Fern, and J. Hurst, "Sim-to-real learning of all common bipedal gaits via periodic reward composition," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Xi'an, China, June 2021, pp. 7309–7315.

[30] H. Duan, A. Malik, J. Dao, A. Saxena, K. Green, J. Siekmann, A. Fern, and J. Hurst, "Sim-to-real learning of footstep-constrained bipedal dynamic walking," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, 2022, pp. 10 428–10 434.

[31] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Proc. Conf. Robot Learn. (CoRL)*, London, UK, Nov. 2021, pp. 91–100.

[32] G. Ji, J. Mun, H. Kim, and J. Hwangbo, "Concurrent training of a control policy and a state estimator for dynamic and robust legged locomotion," *IEEE Robot. Automat. Lett. (RA-L)*, vol. 7, no. 2, pp. 4630 – 4637, Apr. 2022.

[33] J. Lee, J. Hwangbo, and M. Hutter, "Robust recovery controller for a quadrupedal robot using deep reinforcement learning," *arXiv preprint arXiv:1901.07517*, 2019.

[34] Z. Xie, X. Da, M. van de Panne, B. Babich, and A. Garg, "Dynamics randomization revisited: A case study for quadrupedal locomotion," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Virtual, May 2021, pp. 4955–4961.

[35] L. Smith, J. C. Kew, X. B. Peng, S. Ha, J. Tan, and S. Levine, "Legged robots that keep on learning: Fine-tuning locomotion policies in the real world," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*. IEEE, 2022, pp. 1593–1599.

[36] Z. Xie, S. Starke, H. Y. Ling, and M. van de Panne, "Learning soccer juggling skills with layer-wise mixture-of-experts," in *ACM SIGGRAPH 2022 Conference Proceedings*, 2022, pp. 1–9.

[37] F. Shi, T. Homberger, J. Lee, T. Miki, M. Zhao, F. Farshidian, K. Okada, M. Inaba, and M. Hutter, "Circus anymal: A quadruped learning dexterous manipulation with its limbs," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*. IEEE, 2021, pp. 2316–2323.

[38] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa, "AMP: Adversarial motion priors for stylized physics-based character control," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, pp. 1–20, 2021.

[39] S. Liu, G. Lever, Z. Wang, J. Merel, S. M. A. Eslami, D. Hennes, W. M. Czarnecki, Y. Tassa, S. Omidshafiei, A. Abdolmaleki, N. Y. Siegel, L. Hasenclever, L. Marris, S. Tunyasuvunakool, H. F. Song, M. Wulfmeier, P. Muller, T. Haarnoja, B. Tracey, K. Tuyls, T. Graepel, and N. Heess, "From motor control to team play in simulated humanoid football," *Science Robotics*, vol. 7, no. 69, p. eabo0235, 2022.

[40] K. Ploeger, M. Lutter, and J. Peters, "High acceleration reinforcement learning for real-world juggling with binary rewards," *arXiv preprint arXiv:2010.13483*, 2020.

[41] K. Mülling, J. Kober, O. Kroemer, and J. Peters, "Learning to select

and generalize striking movements in robot table tennis," *Int. J. Robot. Res. (IJRR)*, vol. 32, no. 3, pp. 263–279, 2013.

[42] A. Zeng, S. Song, J. Lee, A. Rodriguez, and T. Funkhouser, "Tossingbot: Learning to throw arbitrary objects with residual physics," *IEEE Trans. Robot. (T-RO)*, vol. 36, no. 4, pp. 1307–1319, 2020.

[43] S. Abeyruwan, L. Graesser, D. B. D'Ambrosio, A. Singh, A. Shankar, A. Bewley, and P. R. Sanketi, "i-sim2real: Reinforcement learning of robotic policies in tight human-robot interaction loops," *arXiv preprint arXiv:2207.06572*, 2022.

[44] M. Mittal, D. Hoeller, F. Farshidian, M. Hutter, and A. Garg, "Articulated object interaction in unknown scenes with whole-body mobile manipulation," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 1647–1654.

[45] Y. Ma, F. Farshidian, T. Miki, J. Lee, and M. Hutter, "Combining learning-based locomotion policy with model-based manipulation for legged mobile manipulators," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2377–2384, 2022.

[46] Z. Fu, X. Cheng, and D. Pathak, "Deep whole-body control: learning a unified policy for manipulation and locomotion," *arXiv preprint arXiv:2210.10044*, 2022.