Closing the Sim-to-Real Gap for Ultra-Low-Cost, Resource-Constrained, Quadruped Robot Platforms

Jason Jabbour
University of Virginia
jasonjabbour@virginia.edu

Sabrina M. Neuman
Harvard University
sneuman@seas.harvard.edu

Mark Mazumder
Harvard University
markmazumder@g.harvard.edu

Colby Banbury
Harvard University
cbanbury@g.harvard.edu

Shvetank Prakash
Harvard University
sprakash@g.harvard.edu

Brian Plancher
Harvard University
brian_plancher@g.harvard.edu

Vijay Janapa Reddi
Harvard University
vj@eecs.harvard.edu

***Abstract*—Automating robust walking gaits for legged robots has been a long-standing challenge. Previous work has achieved robust locomotion gaits on sophisticated quadruped hardware platforms through the use of reinforcement learning and imitation learning. However, these approaches do not consider the strict constraints of ultra-low-cost robot platforms with limited computing resources, few sensors, and restricted actuation. These constrained robot platforms require special attention to successfully transfer skills learned in simulation to reality. As a step toward robust learning pipelines for these constrained robot platforms, we demonstrate how existing state-of-the-art imitation learning pipelines can be modified and augmented to support low-cost, limited hardware. By reducing our model's observational space, leveraging TinyML to quantize our model, and adjusting the model outputs through post-processing, we are able to learn and deploy successful walking gaits on an 8-DoF, $299 (USD) toy quadruped robot that has reduced actuation and sensor feedback, as well as limited computing resources. A video of our current results can be found at: https://youtu.be/jloya0TOzWA.**

I. Introduction and Related Work

Imitation learning has been used to achieve complicated, natural-looking skills by leveraging real animal locomotion [7], enabling complex quadruped, bipedal, and humanoid robots to achieve a range of skills, including walking, trotting, and back-flipping [6, 8]. Such pipelines utilize motion capture data of a real-life character performing a skill, and retarget the motion onto the robot frame in simulation. The retargeted motion capture data is then used as the reference to train a reinforcement learning policy where the reward function encourages the learned policy to emulate the reference motion.

The imitation learning pipeline developed by Peng et al. [7] has been successfully demonstrated to close the reality gap and perform well on sophisticated quadruped robot platforms such as the Unitree A1 and Laikago robots [13]. Complex locomotion skills, including pace, trot, side-step, turns, and hop-turn, were transferred from simulation onto the physical robots. However, these policies rely on the precise actuators, a broad menu of sensors, and the high maneuverability of these sophisticated robots. They cannot be directly applied to ultra-low-cost resource-constrained robots, which are a promising solution for applications ranging from search and rescue to routine infrastructure monitoring and maintenance [5]. As

	Unitree A1	Petoi Bittle	Ratio
Cost	$10,000 USD	$299 USD	33x
Weight	12 kg	.29 kg	41x
Dimensions	.5 x .3 x .4 m	.2 x .11 x .11 m	2.5x
Degrees of Freedom (DoF)	12 (Leg: 3)	8 (Leg: 2)	1.5x
Battery Capacity	25.2V 4200mAh	7.4V 1000mAh	3x
Motor Resolution	.022°	1°	45x
IMU	Yes	Yes	NA
Motor Feedback	Yes	No	NA
Foot Pressure Sensor	Yes	No	NA
LiDAR	Yes	No	NA
Computing	ARM Cortex-A72 2.5GHz	Nyboard V1 ATMega328P 20MHz	125x
Optional Additional Computing	NVIDIA TX2 1.3GHz	Raspberry Pi Zero 2W 1GHz	1.3x

Fig. 1. Specification comparison between two quadrupeds, the Unitree A1 [13] and the ultra-low-cost Petoi Bittle [11], highlighting the contrast between the capabilities of fully-featured vs. resource-constrained hardware.

such, further work is needed to extend the usability of these imitation learning pipelines to ultra-low-cost robot platforms that lack these features (Fig. 1). While recent work on constrained robot platforms has achieved locomotion skills in simulation [14], and some have even transferred these skills to hardware [4, 10], this success has yet to be achieved through the use of streamlined imitation learning pipelines.

In this work, we identify the key challenges to adapting imitation learning pipelines to ultra-low-cost robots with poor actuation, limited computing resources, and limited sensors; and propose practical solutions to overcome these difficulties. These initial steps help lay the groundwork for a future with globally-accessible, capable, ultra-low-cost robots.

II. Challenges and Solutions

Observability. The use of low-cost actuators increases the difficulty of closing the reality gap. For example, the servo motors on the Bittle robot do not have encoders. As such, their precise position is not observable. This differs from the motors used on more expensive quadrupeds and breaks state-of-the-art imitation learning pipelines which assume that the robot has perfect knowledge of its joints. In fact, these pipelines assume a 120 dimension observation space consisting of IMU

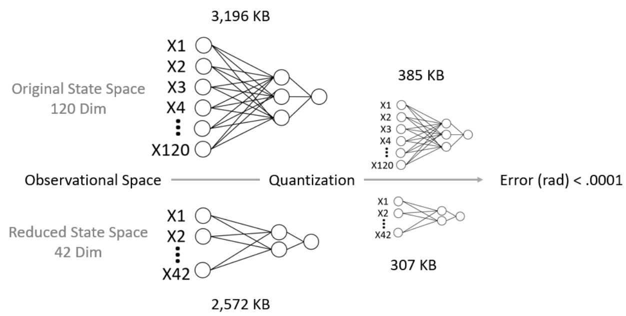

Fig. 2. Reducing observational space, freezing variables, and quantizing model weights significantly decreases model size with minimal accuracy loss.

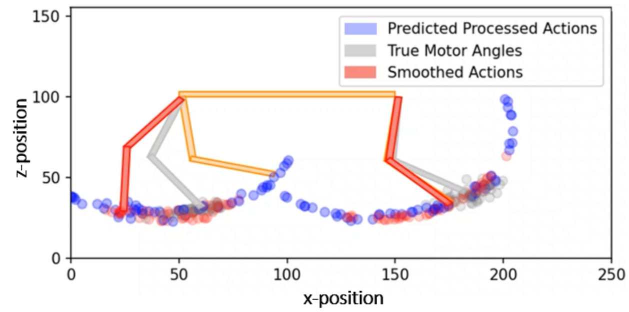

Fig. 3. Smoothing action commands to the real robot helps avoid unrealizable large joint angle changes predicted by the learned model. We show a 2D projection of raw model outputs (blue), actual joint angles reached in simulation (gray), and post-processed smoothed commands (red).

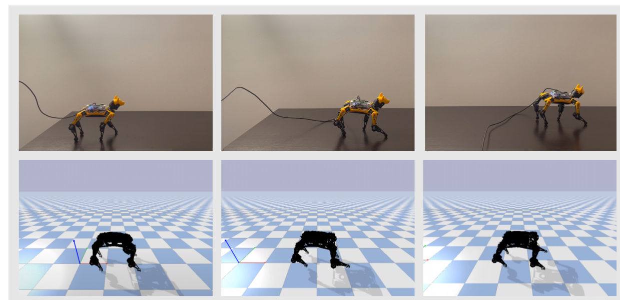

Fig. 4. Adjusting for hardware limitations enables transfer of learned movement from simulation (below) to real ultra-low-cost robots (above).

data (roll, pitch, roll rate, pitch rate), motor angles, previously predicted actions, as well as simulated future reference motion frames. As we cannot run a simulator onboard a robot like Bittle in real time, and cannot observe the joint angles, we adapted our pipeline to use a reduced 42-dimensional space that removes the observed joint angles and future reference motion frames. (Fig. 2). We find that this reduction in the observation space still leads to natural-looking walking gaits.

Computation. The transfer of trained policies from simulation to real-life is more difficult on ultra-low-cost platforms due to their limited onboard computing. The neural networks trained on a server or laptop might not always fit on a constrained platform's computing resources, e.g., Bittle's onboard microcontroller, or internet-of-things (IoT) processors like the RaspberryPi Zero 2W [9] that can be optionally added to Bittle (at the expense of additional power and weight). As such we employ embedded machine learning "TinyML" [1] techniques to reduce the size of the model. By applying graph freezing to convert all variables to constants, and Float16 quantization to convert floating point weights from 32-bit to 16-bit, we reduce the size of the model by a factor of 8x. TinyML techniques have been very successful at reducing the size of ML models without sacrificing accuracy [2], and in our case it also does not have a large impact on accuracy, producing a reconstruction error of less than 0.0001 radians (Fig. 2). Overall, by combining the reduced observation space, graph freezing, and quantization we reduce the model size by 10x.

Controllability. A discrepancy between controlling the simulated Bittle versus the physical Bittle can be found in the actuator precision. The simulated Bittle can reach any joint angle predicted by the policy. However, this is inconsistent with reality. The servo motors found on the physical Bittle can only be commanded by joint actions changes that are 1° or larger. As such, we incorporated a 1-degree dead band zone into the simulation during training. Another issue is that communication with the servo motors using Bittle's source code limits the speed of actions passed to the motors. Commands are only applied to the servo motors if the previous command is completed. Since the simulated Bittle can be commanded a new action while the joints are still in motion, the model predicts large joint angle changes as targets that the simulated robot never reaches. This causes the physical robot to try to achieve motions that are much larger than desired. To address this, we smooth the predicted actions using a trailing moving average (Fig. 3).

III. Hardware Experiments

Methodology. We deployed our model onto a RaspberryPi Zero 2W [9] (an IoT-scale device) placed on top of an 8-DoF Petoi Bittle quadruped [11]. We trained our policy for 60 million steps using the Pybullet physics engine [3] and the stable-baselines PPO1 policy [12]. Once deployed on the real-life Bittle robot, each action is delivered to the servo motors using serial commands. A notable sim-to-real inconsistency was found between the coordinate frame orientations of the simulation and physical Bittle IMU. Since the physical IMU was inverted, we rotated IMU data received from the physical Bittle 180° about the y-axis to synchronize the frames.

Results. As illustrated in Fig. 4, by correcting the IMU data and employing the solutions described in Section II, we were able to achieve a stable walking gate on physical robot hardware. A video of this experiment can be found at https://youtu.be/jloya0TOzWA.

IV. Conclusion and Future Work

In this paper, we demonstrate that sophisticated imitation learning pipelines can be applied to ultra-low-cost robot platforms by understanding and adjusting for hardware limitations. We see many opportunities for future work, including: exploring the development of robust policies that leverage data from additional low-cost sensors; developing policies for more complex movements; and leveraging "helper" policies to adjust for the actuator and sensor limitations of our robot platform.

References

[1] Colby R Banbury, Vijay Janapa Reddi, Max Lam, William Fu, Amin Fazel, Jeremy Holleman, Xinyuan Huang, Robert Hurtado, David Kanter, Anton Lokhmotov, David Patterson, Danilo Pau, Jae-sun Seo, Jeff Sieracki, Urmish Thakker, Marian Verhelst, and Poonam Yadav. Benchmarking tinyml systems: Challenges and direction. *MLSys Workshop on Bench. ML Work. Emerging HW*, 2020.

[2] Sevda Ozge Bursa, Ozlem Durmaz Incel, and Gulfem Isiklar Alptekin. Transforming Deep Learning Models for Resource-Efficient Activity Recognition on Mobile Devices. In *2022 5th Conference on Cloud and Internet of Things (CIoT)*, pages 83–89, 2022. doi: 10.1109/CIoT53061.2022.9766512.

[3] Erwin Coumans and Yunfei Bai. Bullet Real-Time Physics Simulation, 2016-2019. URL https://pybullet.org/.

[4] Shimpei Masuda and Kuniyuki Takahashi. Sim-to-Real Learning of Robust Compliant Bipedal Locomotion on Torque Sensor-Less Gear-Driven Humanoid. *arXiv preprint arXiv:2204.03897*, 2022.

[5] Sabrina M. Neuman, Brian Plancher, Bardienus P. Duisterhof, Srivatsan Krishnan, Colby Banbury, Mark Mazumder, Shvetank Prakash, Jason Jabbour, Aleksandra Faust, Guido C. H. E. de Croon, and Vijay Janapa Reddi. Tiny robot learning: challenges and directions for machine learning in resource-constrained robots. In *2022 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*. IEEE, 2022.

[6] Xue Bin Peng, Angjoo Kanazawa, Jitendra Malik, Pieter Abbeel, and Sergey Levine. SFV: Reinforcement Learning of Physical Skills from Videos. *ACM Trans. Graph.*, 37(6), November 2018. doi: 10.1145/3272127.3275014. URL https://doi.org/10.1145/3272127.3275014.

[7] Xue Bin Peng, Erwin Coumans, Tingnan Zhang, Tsang-Wei Lee, Jie Tan, and Sergey Levine. Learning Agile Robotic Locomotion Skills by Imitating Animals. 2020. doi: 10.48550/ARXIV.2004.00784. URL https://arxiv.org/abs/2004.00784.

[8] Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. AMP: Adversarial Motion Priors for Stylized Physics-Based Character Control. 40(4), 2021. ISSN 0730-0301. doi: 10.1145/3450626.3459670. URL https://doi.org/10.1145/3450626.3459670.

[9] Raspberry Pi. Raspberry Pi Zero 2W Your Tiny, Tiny $15 Computer, 2021. URL https://www.raspberrypi.com/products/raspberry-pi-zero-2-w/.

[10] Maurice Rahme, Ian Abraham, Matthew L. Elwin, and Todd D. Murphey. Linear Policies are Sufficient to Enable Low-Cost Quadrupedal Robots to Traverse Rough Terrain. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8469–8476, 2021. doi: 10.1109/IROS51168.2021.9636011.

[11] Li Rongzhong. Petoi Bittle Robot, Accessed 2022. URL https://www.petoi.com/products/petoi-bittle-robot-dog.

[12] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

[13] Unitree. Unitree A1 Robot, Accessed 2022. URL https://www.unitree.com/products/a1/.

[14] Chong Zhang, Wanming Yu, and Zhibin Li. Accessibility-Based Clustering for Efficient Learning of Locomotion Skills. *arXiv preprint arXiv:2109.11191*, 2021. URL https://arxiv.org/abs/2109.11191.