Morphology-Aware Legged Locomotion with Reinforcement Learning

Nico Bohlinger[1], Grzegorz Czechmanowski[2,4], Maciej Krupka[2], Piotr Kicki[2,4]
Krzysztof Walas[2,4], Jan Peters[1,3,5], Davide Tateo[1]

***Abstract*— The field of legged robotics is still missing a single learning framework that can control different embodiments—such as quadruped, humanoids, and hexapods—simultaneously and transfer, zero or few-shot, to unseen robot embodiments. To close this gap, we introduce URMA, the Unified Robot Morphology Architecture. Our framework brings the end-to-end Multi-Task Reinforcement Learning approach to the realm of legged robots, enabling the learned policy to control any type of robot morphology. We show that URMA can learn a locomotion policy on multiple embodiments that can be transferred to unseen robots in simulation and the real world.**

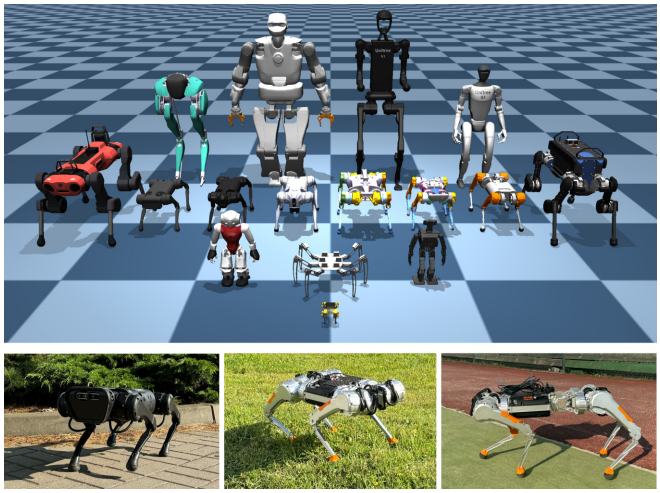

Fig. 1: Top – We train a single locomotion policy for multiple robot embodiments in simulation. Bottom – We can transfer and deploy the policy on three real-world platforms by randomizing the embodiments and environment dynamics during training.

I. INTRODUCTION

Deep Reinforcement Learning (DRL) has enabled legged robots to achieve remarkable locomotion capabilities. Quadruped robots, in particular, have demonstrated highly agile movements, including running at high speeds, jumping obstacles, navigating rough terrain, performing handstands, and even completing parkour courses [1], [2], [3], [4], [5], [6]. A key long-term goal is the development of foundation models for locomotion, enabling zero-shot deployment on any arbitrary robot platform. Achieving this requires adapting the underlying learning architecture to handle diverse tasks and morphologies. Current Multi-Task Reinforcement Learning (MTRL) methods often handle varying observation and action spaces by padding inputs with zeros [7] or employing separate neural network heads for each task [8]. Early work on controlling different morphologies leveraged Graph Neural Networks (GNNs) to represent the morphological and kinematic structure of robots [9], [10], [11]. Although these methods can control various robots, they struggle to generalize to a wide range of embodiments. More recently, Transformer-based architectures have been introduced to address this limitation, using the attention mechanism to aggregate information from varying numbers of joints [12], [13]. However, these methods still lack true generality, as they are often restricted to a predefined set of morphologies.

[1] Department of Computer Science, Technical University of Darmstadt, Germany. [2] Institute of Robotics and Machine Intelligence, Poznan University of Technology, Poland. [3] German Research Center for AI (DFKI), Research Department: Systems AI for Robot Learning. [4] IDEAS NCBR, Warsaw, Poland. [5] Hessian.AI. Corresponding author: nico.bohlinger@tu-darmstadt.de

II. MORPHOLOGY-AWARE LEARNING

We propose the Unified Robot Morphology Architecture (URMA), a complete morphology-aware architecture, that does not require defining the possible morphologies beforehand and can adapt to arbitrary joint configurations with the same network. Figure 2 presents a schematic overview of URMA. To handle observations of any morphology, URMA first splits the observation vector o into robot-specific and general observations o_g, where the former can be of varying size, and the latter has a fixed dimensionality. For locomotion, we subdivide the robot-specific observations into joint and feet-specific observations. In the following text, we describe everything w.r.t. the joint-specific observations, but the same applies to the feet-specific ones as well. Every joint of a robot is composed of joint-specific observations o_j and a description vector d_j. These description vectors are made up of fixed dynamics and kinematics properties of the joint that can uniquely describe the joint, in our case: joint limits, maximum velocity and torque, relative joint position and rotation axis in a nominal configuration, PD gains, etc. The description vectors and joint-specific observations are encoded separately by the Multilayer Perceptrons (MLPs) f_ϕ and f_ψ and are then passed through a simple attention head, with a learnable temperature τ and a minimum temperature ϵ, to get a single latent vector

$$\bar{z}_{\text{joints}} = \sum_{j \in J} z_j, \qquad z_j = \frac{\exp\left(\frac{f_\phi(d_j)}{\tau + \epsilon}\right)}{\sum_{j \in J} \exp\left(\frac{f_\phi(d_j)}{\tau + \epsilon}\right)} f_\psi(o_j), \tag{1}$$

that contains the information of the joint-specific observations of all joints. With the help of the attention mechanism,

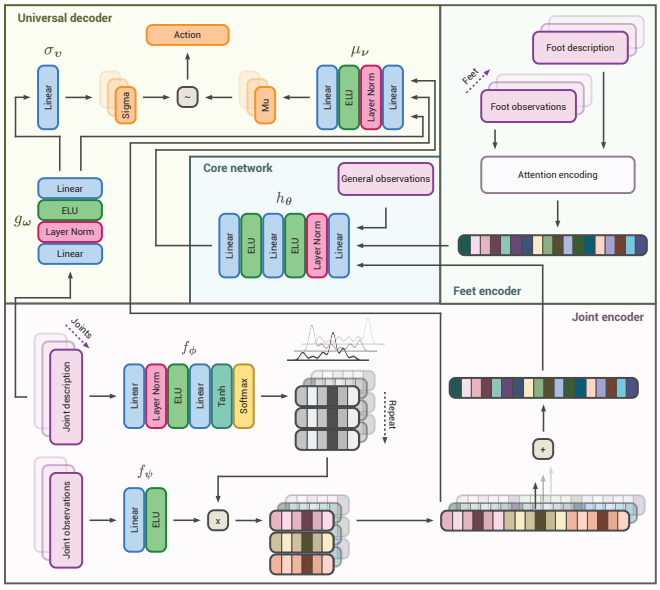

Fig. 2: Overview: Unified Robot Morphology Architecture.

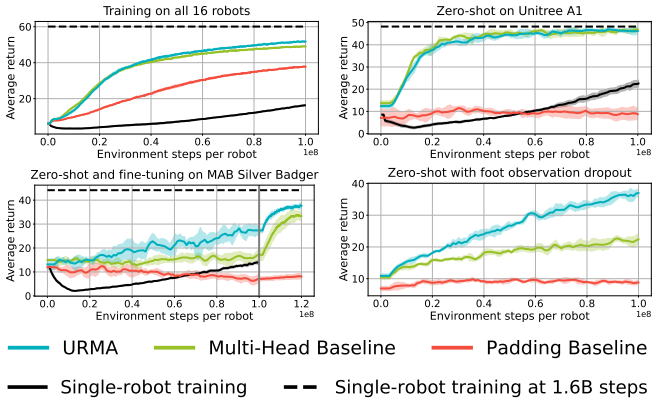

Fig. 3: Top left – Average return of the three architectures during training on all 16 robots compared to the single-robot training setting. Top right – Zero-shot transfer to the Unitree A1 while training on the other 15 robots. Bottom left – Zero-shot transfer to the MAB Robotics Silver Badger while training on the other 15 robots and fine-tuning on only the Silver Badger afterward. Bottom right – Zero-shot evaluation on all 16 robots while removing the feet observations.

the network can learn to separate the relevant joint information and precisely route it into the specific dimensions of the latent vector by reducing the temperature τ of the softmax close to zero. The joint latent vector $\bar{z}_{\text{joints}}$ is then concatenated with the feet latent vector $\bar{z}_{\text{feet}}$ and the general observations o_g and passed to the policies core MLP h_θ to get the action latent vector $\bar{z}_{\text{action}} = h_\theta(o_g, \bar{z}_{\text{joints}}, \bar{z}_{\text{feet}})$. To obtain the final action for the robot, we use our universal morphology decoder, which takes the general action latent vector and pairs it with the set of encoded specific joint descriptions and the single joint latent vectors to produce the mean and standard deviation of the actions for every joint, from which the final action is sampled as

$$a^j \sim \mathcal{N}(\mu_\nu(d_j^a, \bar{z}_{\text{action}}, z_j), \sigma_\upsilon(d_j^a)), \qquad d_j^a = g_\omega(d_j). \tag{2}$$

III. RESULTS

To evaluate the training efficiency of MTRL in our setting, we train URMA and the multi-head and a zero-padding baselines on all 16 robots in the training set simultaneously (100 million steps per robot) and compare the average return to the single-robot training setting, where a separate policy is trained for every robot. Figure 3 confirms the advantage in learning efficiency of MTRL over single-task learning, as URMA and the multi-head baseline learn significantly faster than training only on a single robot at a time. URMA reaches the highest average return at the end. Next, we evaluate the zero-shot transfer on the Unitree A1, a robot whose embodiment is similar to other quadrupeds in the training set. Figure 3 shows the evaluation for the A1 during a training process with the other 15 robots and highlights that both URMA and the multi-head baseline can transfer perfectly well to the A1 while never having seen it during training. To investigate an out-of-distribution embodiment, we use the same setup as for the A1 and evaluate zero-shot on the MAB Robotics Silver Badger robot, which has an additional spine joint in the trunk and lacks feet observations, and then fine-tune the policies for 20 million steps only on the Silver Badger itself. The results show that URMA can handle the additional joint and the missing feet observations better than the baselines and is the only method capable of achieving a good gait at the end of training. To further assess the adaptability of our approach, we evaluate the zero-shot performance in the setting where observations are dropped out, which can easily happen in real-world scenarios due to sensor failures. We train the architectures on all robots with all observations and evaluate them on all robots while completely dropping the feet observations. Figure 3 confirms the results from the previous experiment and shows that URMA can handle missing observations better than the baselines. Finally, we deploy the same URMA policy on the real Unitree A1, MAB Honey Badger, and MAB Silver Badger quadruped robots. Figure 1 shows the robots walking with the learned policy on pavement, grass, and plastic turf terrain with slight inclinations. While the Unitree A1 and the MAB Silver Badger are in the training set, the network is not trained on the MAB Honey Badger. Despite the Honey Badger's gait not being as good as the other two robots, it can still locomote robustly on the terrain we tested, proving the generalization capabilities of our architecture and training scheme.

ACKNOWLEDGMENT

This project was funded by National Science Centre, Poland under the OPUS call in the Weave program UMO-2021/43/I/ST6/02711, and by the German Science Foundation (DFG) under grant number PE 2315/17-1. Part of the calculations were conducted on the Lichtenberg high

performance computer at TU Darmstadt.

REFERENCES

[1] 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.

[2] 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.

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

[4] K. Caluwaerts, A. Iscen, J. C. Kew, W. Yu, T. Zhang, D. Freeman, K.-H. Lee, L. Lee, S. Saliceti, V. Zhuang *et al.*, "Barkour: Benchmarking animal-level agility with quadruped robots," *arXiv preprint arXiv:2305.14654*, 2023.

[5] Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao, "Robot parkour learning," in *Conference on Robot Learning (CoRL)*, 2023.

[6] X. Cheng, K. Shi, A. Agarwal, and D. Pathak, "Extreme parkour with legged robots," in *RoboLetics: Workshop on Robot Learning in Athletics@ CoRL 2023*, 2023.

[7] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *Conference on robot learning*. PMLR, 2020, pp. 1094–1100.

[8] C. D'Eramo, D. Tateo, A. Bonarini, M. Restelli, J. Peters *et al.*, "Sharing knowledge in multi-task deep reinforcement learning," in *8th International Conference on Learning Representations,{ICLR} 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. International Conference on Learning Representations, ICLR, 2020, pp. 1–11.

[9] T. Wang, R. Liao, J. Ba, and S. Fidler, "Nervenet: Learning structured policy with graph neural networks," in *International conference on learning representations*, 2018.

[10] W. Huang, I. Mordatch, and D. Pathak, "One policy to control them all: Shared modular policies for agent-agnostic control," in *International Conference on Machine Learning*. PMLR, 2020, pp. 4455–4464.

[11] J. Whitman, M. Travers, and H. Choset, "Learning modular robot control policies," *IEEE Transactions on Robotics*, 2023.

[12] V. Kurin, M. Igl, T. Rocktäschel, J. Böhmer, and S. Whiteson, "My body is a cage: the role of morphology in graph-based incompatible control," in *International Conference on Learning Representations (ICLR)*, 2021.

[13] B. Trabucco, M. Phielipp, and G. Berseth, "Anymorph: Learning transferable polices by inferring agent morphology," in *International Conference on Machine Learning*. PMLR, 2022, pp. 21 677–21 691.