Learning Robot Locomotion for Multiple Embodiments

Nico Bohlinger[1], Grzegorz Czechmanowski[2, 3] Maciej Krupka[2] Piotr Kicki[2, 3]
Krzysztof Walas[2, 3] Jan Peters[1, 4, 5, 6] Davide Tateo[1]
[1]Department of Computer Science, Technical University of Darmstadt, Germany
[2]Institute of Robotics and Machine Intelligence, Poznan University of Technology, Poland
[3]IDEAS NCBR, Warsaw, Poland
[4]German Research Center for AI (DFKI), Research Department: Systems AI for Robot Learning
[5]Hessian.AI

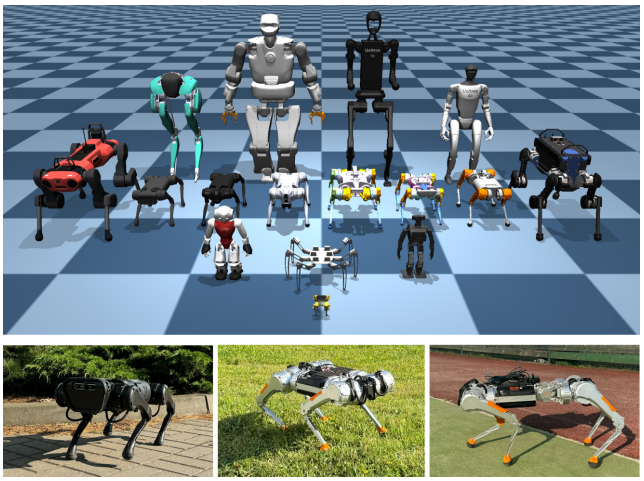

Figure 1: Top – We train a single locomotion policy for multiple robot embodiments in simulation. Bottom – We can transfer and deploy the policy on three real-world platforms by randomizing the embodiments and environment dynamics during training.

1 Introduction

In recent years, Deep Reinforcement Learning (DRL) techniques are achieving state-of-the-art results in robust legged locomotion [1, 2, 3, 4]. While there exists a wide variety of legged platforms such as quadruped, humanoids, and hexapods, the field is still missing a single learning framework that can control all these different embodiments easily and effectively and possibly transfer, zero or few-shot, to unseen robot embodiments. In practice, the number of joints and feet of a legged robot determines the size of its action and observation space, which can differ for every new robot. This often prevents a straightforward transfer of existing policies as the learning architecture fully depends on the specific robot platform. To tackle this problem, we introduce Unified Robot Morphology Architecture (URMA). Our framework brings the end-to-end Multi-Task Reinforcement Learning (MTRL) approach to the realm of legged robots, enabling the learned policy to control any type of robot morphology. The key idea of our method is to allow the network to learn an abstract locomotion controller that can be seamlessly shared between embodiments thanks to our morphology-aware encoders and decoders. This flexible architecture can be seen as a potential first step in building a foundation model for legged robot locomotion. Our experiments show that URMA can learn a locomotion policy on multiple embodiments that can be easily transferred to unseen robot platforms in simulation and the real world.

2 Related Workd

With the help of DRL techniques, legged robots can perform impressive locomotion skills. There are numerous examples of highly agile locomotion with quadrupedal robots [1, 3, 5, 6, 7, 8], learning to run at high speeds, jumping over obstacles, walking on rough terrain, performing handstands, and completing parkour courses. Similar methods have been applied to generate robust walking gaits for bipedal and humanoid robots [2, 9, 10]. However, an important long-term objective is to develop foundation models for locomotion, allowing zero-shot (or few-shot) deployment to any arbitrary robot platform. To reach this objective, it is fundamental to adapt the underlying learning system to support different tasks and morphologies. To handle differently sized observation and action spaces, MTRL baselines often resort to padding the observations and actions with zeros to fit a maximum length [11] or to using a separate neural network head for each task [12]. These methods allow for efficient training but can be limiting when trying to transfer to new tasks or environments. Earlier work on controlling different robot morphologies is based on the idea of using Graph Neural Networks (GNNs) to capture the morphological and kinematic structure of the robots [13, 14, 15]. These approaches can control different robots even when removing some of their limbs, but they struggle to generalize to many different morphologies. Transformer-based architectures have been proposed to overcome those limitations by using the attention mechanism to globally aggregate information of varying numbers of joints [16, 17]. These methods still lack substantial generality as they are limited to morphologies that were defined a priori.

3 Unified Robot Morphology Architecture

We propose the 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

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

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

$$a^j \sim \mathcal{N}(\mu_v(d_j^a, \bar{z}_{\text{action}}, z_j), \sigma_v(d_j^a)), \qquad d_j^a = g_\omega(d_j). \qquad (2)$$

4 Experiments

To evaluate the training efficiency of MTRL in our setting, we train URMA, a multi-head [12] and a zero-padding [11] architecture 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.

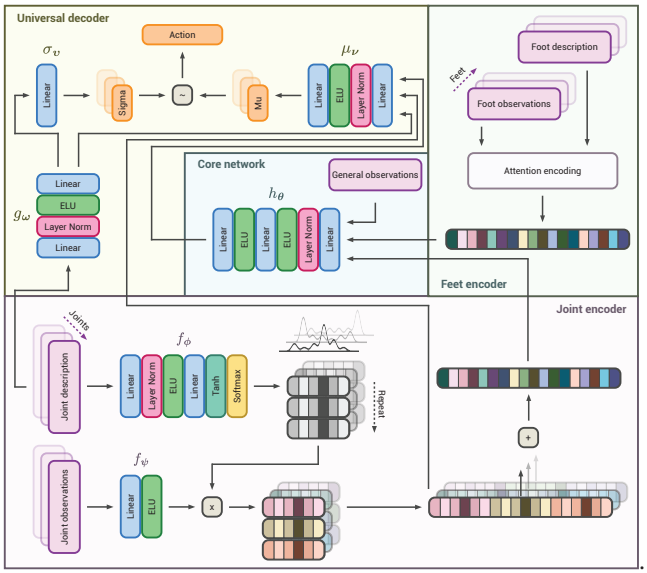

Figure 2: Overview of the URMA architecture.

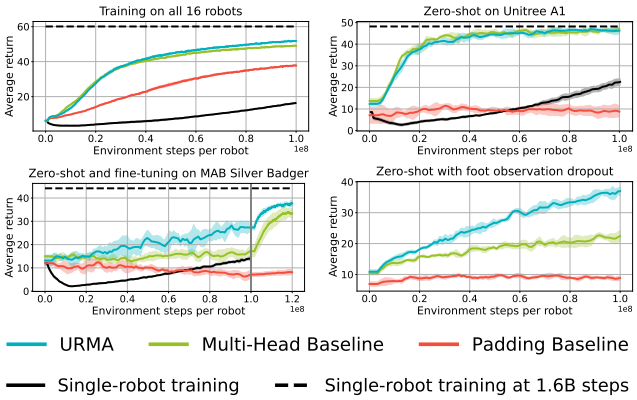

Figure 3: Top left – Average return of the three architectures during training on all 16 robots compared to the single-robot training setting. Top right – Zero-shot transfer to the Unitree A1 while training on the other 15 robots. Bottom left – Zero-shot transfer to the MAB Robotics Silver Badger while 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.

5 Conclusion

We presented URMA, an end-to-end framework to learn robust locomotion for different types of robot morphologies with a single neural network architecture. Our morphology-aware encoders and decoders allow URMA to learn a single control policy for 16 different embodiments from three different legged robot morphologies. In practice, URMA reaches higher final performance when training with all embodiments, shows higher robustness to observation dropout, and better zero-shot capabilities to new robots compared to MTRL baselines. Furthermore, we deploy the same policy zero-shot on two known and one unseen quadruped robot in the real world. We argue that this multi-embodiment learning setting can be easily extended to more complex scenarios and can serve as a basis for locomotion foundation models that can act on the lowest level of robot control. Finally, the URMA architecture is general enough to be applied to not only any robot embodiment but also any control task, making task generalization, also for non-locomotion tasks, an interesting avenue for future research.

6 Acknowledgments

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] Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62):eabk2822, 2022.

[2] Ashish Kumar, Zhongyu Li, Jun Zeng, Deepak Pathak, Koushil Sreenath, and Jitendra Malik. Adapting rapid motor adaptation for bipedal robots. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1161–1168. IEEE, 2022.

[3] Gabriel B Margolis and Pulkit Agrawal. Walk these ways: Tuning robot control for generalization with multiplicity of behavior. In *Conference on Robot Learning*, pages 22–31. PMLR, 2023.

[4] Gilbert Feng, Hongbo Zhang, Zhongyu Li, Xue Bin Peng, Bhuvan Basireddy, Linzhu Yue, Zhitao Song, Lizhi Yang, Yunhui Liu, Koushil Sreenath, et al. Genloco: Generalized locomotion controllers for quadrupedal robots. In *Conference on Robot Learning*, pages 1893–1903. PMLR, 2023.

[5] Suyoung Choi, Gwanghyeon Ji, Jeongsoo Park, Hyeongjun Kim, Juhyeok Mun, Jeong Hyun Lee, and Jemin Hwangbo. Learning quadrupedal locomotion on deformable terrain. *Science Robotics*, 8(74):eade2256, 2023.

[6] Ken Caluwaerts, Atil Iscen, J Chase Kew, Wenhao Yu, Tingnan Zhang, Daniel Freeman, Kuang-Huei Lee, Lisa Lee, Stefano Saliceti, Vincent Zhuang, et al. Barkour: Benchmarking animal-level agility with quadruped robots. *arXiv preprint arXiv:2305.14654*, 2023.

[7] Ziwen Zhuang, Zipeng Fu, Jianren Wang, Christopher Atkeson, Sören Schwertfeger, Chelsea Finn, and Hang Zhao. Robot parkour learning. In *Conference on Robot Learning (CoRL)*, 2023.

[8] Xuxin Cheng, Kexin Shi, Ananye Agarwal, and Deepak Pathak. Extreme parkour with legged robots. In *RoboLetics: Workshop on Robot Learning in Athletics@ CoRL 2023*, 2023.

[9] Jonah Siekmann, Kevin Green, John Warila, Alan Fern, and Jonathan Hurst. Blind bipedal stair traversal via sim-to-real reinforcement learning. In *Robotics: Science and Systems*, 2021.

[10] Ilija Radosavovic, Tete Xiao, Bike Zhang, Trevor Darrell, Jitendra Malik, and Koushil Sreenath. Real-world humanoid locomotion with reinforcement learning. *arXiv:2303.03381*, 2023.

[11] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020.

[12] 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*, pages 1–11. International Conference on Learning Representations, ICLR, 2020.

[13] Tingwu Wang, Renjie Liao, Jimmy Ba, and Sanja Fidler. Nervenet: Learning structured policy with graph neural networks. In *International conference on learning representations*, 2018.

[14] Wenlong Huang, Igor Mordatch, and Deepak Pathak. One policy to control them all: Shared modular policies for agent-agnostic control. In *International Conference on Machine Learning*, pages 4455–4464. PMLR, 2020.

[15] Julian Whitman, Matthew Travers, and Howie Choset. Learning modular robot control policies. *IEEE Transactions on Robotics*, 2023.

[16] Vitaly Kurin, Maximilian Igl, Tim Rocktäschel, JW Böhmer, and Shimon Whiteson. My body is a cage: the role of morphology in graph-based incompatible control. In *International Conference on Learning Representations (ICLR)*, 2021.

[17] Brandon Trabucco, Mariano Phielipp, and Glen Berseth. Anymorph: Learning transferable polices by inferring agent morphology. In *International Conference on Machine Learning*, pages 21677–21691. PMLR, 2022.