ES-sim2real optimization

Giacomo Spigler g.spigler@tilburguniversity.edu; Cesare Maria Dalbagno c.m.dalbagno@tilburguniversity.edu

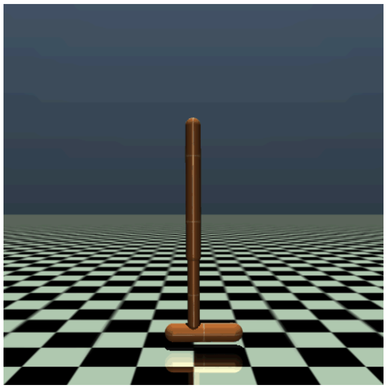
Tilburg University, Department of Cognitive Science and Artificial Intelligence

Introduction

Robotics is one of the hottest topics in technology for its rapid development and potential social impact. Latest advancements such as the Figure humanoid robot (Figure [@Figure_robot], 2024) and NVIDIA's GROOT project (*Project GR00T*, n.d.) seem to lay solid foundations for a future in which robotics is increasingly more integrated in everyday life, improving the living conditions of many.

All of these advances in achieving such complex tasks with high levels of uncertainty and many degrees of freedom have been possible thanks to Deep Reinforcement Learning (François-Lavet et al., 2018). However, this method remains very unstable and difficult to apply to real world due to what is known as 'sim-to-real gap'. That is: the algorithm trains on extensive data in a simulated environment with perfect information but, when transferred to the real world, even slight deviations in the parameters which the robot has adapted to lead to a plunge in performance (Zhao et al., 2020). To make up for this, different techniques have been investigated: in particular, the implementation of a Meta Learning approach seems to help for generalization outside of the task distribution the robot has encountered during training, therefore making it more robust when tested in a real world scenario (Song, Yang, et al., 2020).

Building on top of the results obtained by Song et al. we propose a method which, after training in a randomized simulated environment, does not expect the robot to perform well out of the box. However, allowing it for few interaction episodes with the environment we expect a quick adaptation to the new task. The combination of Meta-Reinforcement Learning with domain randomization techniques will hopefully lead to more reliable and stable algorithms, well suited for human-robot interaction.



Methods

We have run our experiments in the Gymnasium environment '*Hopper-v4*', implemented in the physics simulator Mujoco (Erez et al., 2012; Todorov et al., 2012; Towers et al., 2023). We trained a fully connected neural network using reinforcement learning using Stable Baselines3 to handle training in the simulated environment using PPO (Raffin et al., 2021; Schulman et al., 2017). The task the one-legged robot is trying to achieve is to move as quickly as possible in one direction without falling. We have tested performances in three distinct learning

Hopper	μ	σ	max	min
torso mass	6	1.5	9.5	1.5
ground friction	2	0.25	2.5	1.5

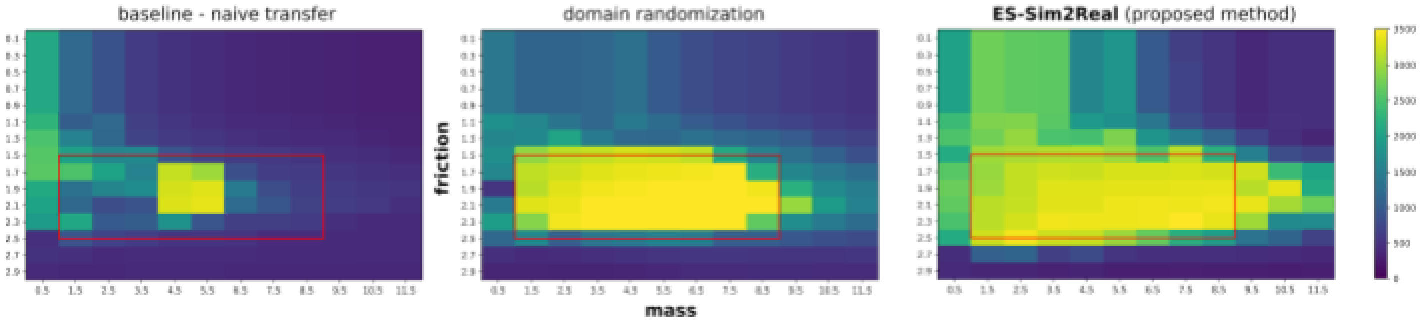
Table 1

conditions. In the first one we used two fixed values for the *torso mass* and *ground friction* parameters in the environment (mass=6, friction=2), secondly, we realized a simple domain randomization by sampling the parameters from two uniform distributions parameterized as *Table 1* displays (Weng, 2019) and lastly we implemented ES-MAML (400 meta iterations, 6 inner loop episodes each) in the same randomization setup (Song, Gao, et al., 2020). All three models were later tested in an environment with parameters outside of the training distribution (mass range = [0.5, 12]; friction range = [0.1, 3]) to resemble the sim-to-real gap. The hypothesis was that we would observe an increase in average performance of the models going from the first to the last training conditions, since the task distribution was larger in the randomized environments and since in the ES-MAML setup the model could adapt to the test environment for 3 episodes.

Results

As Figure 1 reports, when averaged over 5 random seeds, domain randomization in training significantly improves performance compared to the fixed parameters condition, while the preliminary results with our method prove a much better adaptation for out-of-domain tasks.

Figure 1



Analysis of sim2sim generalization performance of three agents. Trained agents are evaluated in a wide range of simulated physics environments that differ in the mass of parts of the agent's body [0.5, 11.5] (kg) and in its foot-ground friction coefficient [0.1, 2.9].

Baseline-naive-transfer agent: an agent is trained in a fixed physics simulator with parameters mass=6kg, friction=2.
As common in naive sim2real transfers, the performance of the agent drops quickly when the simulation parameters are changed.

Domain randomization: an agent is trained over a collection of simulated environments where parameters are sampled uniformly at random within [1, 9] (mass) and [1.5, 2.5] (friction). The range of randomization parameters is shown as a red box in the plots.

Proposed method (ES-Sim2Real): agents are trained using meta-reinforcement-learning to explicitly optimize for quick adaptation to new simulation parameters.

Conclusions

As future directions we would like to apply this proof of concept to a sim-to-real task with a physical robot. Supposedly, even though much more computationally expensive, the ES-sim2real optimization would allow for even more significant improvements with harder tasks. Since ES scales almost linearly with the number of workers, we believe our method could lead to better results with a reasonable computation overload, leading to better robots for everyone.

References

Erez, T., Tassa, Y., & Todorov, E. (2012). *Infinite-Horizon Model Predictive Control for Periodic Tasks with Contacts*. https://doi.org/10.7551/mitpress/9481.003.0015

Figure [@Figure_robot]. (2024, March 13). *With OpenAI, Figure 01 can now have full conversations with people -OpenAI models provide high-level visual and language intelligence -Figure neural networks deliver fast, low-level, dexterous robot actions Everything in this video is a neural network: Https://t.co/OJzMjCv443* [Tweet]. Twitter. https://twitter.com/Figure_robot/status/1767913661253984474

François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., & Pineau, J. (2018). An Introduction to Deep Reinforcement Learning. *Foundations and Trends® in Machine Learning*, *11*(3–4), 219–354. https://doi.org/10.1561/2200000071

Project GR00T. (n.d.). NVIDIA Developer. Retrieved March 31, 2024, from https://developer.nvidia.com/project-gr00t

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., & Dormann, N. (2021). Stable-Baselines3: Reliable Reinforcement Learning Implementations. *Journal of Machine Learning Research*, *22*(268), 1–8. http://jmlr.org/papers/v22/20-1364.html

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). *Proximal Policy Optimization Algorithms* (arXiv:1707.06347). arXiv. https://doi.org/10.48550/arXiv.1707.06347

Song, X., Gao, W., Yang, Y., Choromanski, K., Pacchiano, A., & Tang, Y. (2020). *ES-MAML: Simple Hessian-Free Meta Learning* (arXiv:1910.01215). arXiv. https://doi.org/10.48550/arXiv.1910.01215

Song, X., Yang, Y., Choromanski, K., Caluwaerts, K., Gao, W., Finn, C., & Tan, J. (2020).

Rapidly Adaptable Legged Robots via Evolutionary Meta-Learning. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3769–3776. https://doi.org/10.1109/IROS45743.2020.9341571

Todorov, E., Erez, T., & Tassa, Y. (2012). MuJoCo: A physics engine for model-based control. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5026–5033. https://doi.org/10.1109/IROS.2012.6386109

Towers, M., Terry, J. K., Kwiatkowski, A., Balis, J. U., Cola, G., Deleu, T., Goulão, M., Kallinteris, A., KG, A., Krimmel, M., Perez-Vicente, R., Pierré, A., Schulhoff, S., Tai, J. J., Tan, A. J. S., & Younis, O. G. (2023). *Gymnasium* (v0.29.1) [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.8269265

Weng, L. (2019, May 5). *Domain Randomization for Sim2Real Transfer*. https://lilianweng.github.io/posts/2019-05-05-domain-randomization/

Zhao, W., Queralta, J. P., & Westerlund, T. (2020). Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: A Survey. *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, 737–744. https://doi.org/10.1109/SSCI47803.2020.9308468