Bridging the Reality Gap with PiCRL

Marta Freixo Lopes[1], Roy L.M. Wang[2], Sami Jullien[1], and Stijn Verdenius[2]

[1] University of Amsterdam, Amsterdam, NL
[2] WAIR, Luchtvaartstraat 4, Amsterdam, NL

Abstract. Training a Reinforcement Learning agent in real-world settings requires large amounts of data, which can be both challenging and costly to obtain. To address this, researchers use simulations, emulating real-world conditions for safer training. However, these simulations can't perfectly replicate all real-world processes, leading to a "Reality Gap". In this work, we tackle this problem by developing Pretrained In-Context Reinforcement Learning (PiCRL), a method that allows the training of a robust agent capable of deployment in real scenarios, even with imperfect simulations.

Keywords: Reinforcement Learning · Domain Randomization · System Identification.

1 Method

Our model integrates Domain Randomization (DR) [1] and System Identification (SI) [2] to bridge the Reality Gap [3]. We first develop the Context Transformer (CT) that performs System Identification, followed by the use of Domain Randomization to create a robust agent that leverages information from the CT for improved decision-making.

Context Transformer The Context Transformer (CT) is pretrained using a Transformer [4] architecture to identify the target system from an offline dataset. By randomizing the parameters, we generate various simulations that allow the CT to learn to distinguish between them, supervised by the parameters that define each simulation.

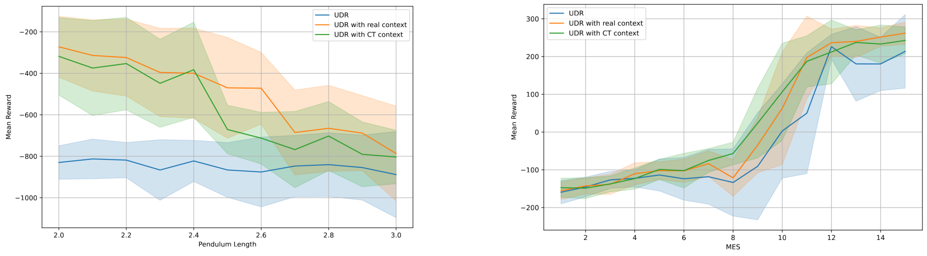
Pretrained in-Context RL (PiCRL) PiCRL trains a robust RL agent capable of adapting to new environments. Similar to Uniform Domain Randomization (UDR), the parameters of the simulator are uniformly randomized at the start of each episode, creating a new environment. We then generate a dataset of transitions from this environment, which the CT uses to create an environmental representation. This representation is then appended to the state and we use this new state for the training of the Reinforcement Learning agent.

2 Results

We conducted experiments with the Pendulum and LunarLander environments from the Gymnasium package, with results presented in Figure 1.

For the Pendulum environment, we pretrained the Context Transformer on instances with randomized Pendulum lengths within $[2.2, 3.0]$ and trained the Reinforcement Learning agent on the same randomization space. Figure 1a shows that our agent consistently outperformed the one trained with standard UDR when tested on environments with lengths randomized within $[2.0, 3.0]$.

For LunarLander, both CT and PiCRL were trained by randomizing the $MainEngineStrength$ within $[8, 15]$ and tested in environments with this parameter uniformly randomized on $[0, 11]$. Although the performance difference between our method and UDR are less pronounced here, as shown in Figure 1b, our approach still proves to be more robust to new environments.



(a) Mean Reward of agent trained on Pendulum with $length \sim U(2.2, 3.0)$, tested with $length \sim U(2.0, 3.0)$.

(b) Mean Reward of agent trained on LunarLander with $MES \sim U(2.2, 3.0)$, tested with $MES \sim U(2.0, 3.0)$.

Fig. 1: Sim-to-Sim experiments.

These experiments demonstrate that incorporating environmental information into RL agent training is crucial for effective policy transfer. Adding this information produces a robust agent with performance nearly equivalent to one that has access to the real simulation parameters.

3 Conclusion

With this project, we successfully designed and trained an end-to-end solution for the Reality Gap. We effectively trained two agents capable of being deployed in unseen environments, performing better than the agents trained with traditional UDR. By using datasets generated by simulations, our method proves to be data-efficient, enabling the creation of a robust agent that can be directly applied to real world scenarios.

Acknowledgments. I sincerely thank my supervisors, Roy and Sami, for their guidance and support throughout this project. I am also grateful to WAIR and Stijn for the opportunity to complete my thesis internship with their team.

References

1. J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, P. Abbeel: Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 2017, pp. 23-30. https://doi.org/10.1109/IROS.2017.8202133.
2. Ljung, L.: System Identification. In: Procházka, A., Uhlíř, J., Rayner, P.W.J., Kingsbury, N.G. (eds) Signal Analysis and Prediction. Applied and Numerical Harmonic Analysis. Birkhäuser, Boston, MA. https://doi.org/10.1007/978-1-4612-1768-8_11
3. Jakobi, N., Husbands, P., Harvey, I.: Noise and the reality gap: The use of simulation in evolutionary robotics. In: Morán, F., Moreno, A., Merelo, J.J., Chacón, P. (eds) Advances in Artificial Life. ECAL 1995. Lecture Notes in Computer Science, vol 929. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-59496-5_337
4. AA. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, et al: Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, 6000–6010.