Speeding up Deep Reinforcement Learning through Influence-Augmented Local Simulators

Extended Abstract

Miguel Suau*, Jinke He, Matthijs T. J. Spaan, and Frans A. Oliehoek
Delft University of Technology
*m.suaudecastro@tudelft.nl

ABSTRACT

Learning effective policies for real-world problems is still an open challenge for the field of reinforcement learning (RL). The main limitation being the amount of data needed and the pace at which that data can be obtained. In this paper, we study how to build lightweight simulators of complicated systems that can run sufficiently fast for deep RL to be applicable. We focus on domains where agents interact with a reduced portion of a larger environment while still being affected by the global dynamics. Our method combines the use of local simulators with learned models that mimic the influence of the global system. The experiments reveal that incorporating this idea into the deep RL workflow can considerably accelerate the training process and presents several opportunities for the future.

KEYWORDS

Simulation; Influence; Deep Reinforcement Learning.

ACM Reference Format:
Miguel Suau*, Jinke He, Matthijs T. J. Spaan, and Frans A. Oliehoek. 2022. Speeding up Deep Reinforcement Learning through Influence-Augmented Local Simulators: Extended Abstract. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022*, IFAAMAS, 3 pages.

1 INTRODUCTION

In this work, we design lightweight versions of large simulators with the goal of speeding up the overall training process. The method we propose applies to domains where agents only interact with a reduced local part of a larger environment, yet they are indirectly being affected by the global dynamics. Traffic control is one example of such environments. Say, for instance, that we wanted to train an agent to control the traffic lights of a particular intersection in a very large city. To do so we could build a small local simulator that captures only the information that is directly relevant to the agent (traffic density in the neighborhood [7]). However, after training, we may find out that an agent that does very well in the small simulator, performs poorly in the real intersection. The performance gap would be caused by a data distribution shift [1, 9]. Even though the simulator might be able to closely mimic the local dynamics (i.e. cars moving within the intersection), it would fail to account for the interactions of the local neighborhood with the rest of the city. Thus, the agent learns a policy based on certain transition dynamics that turn out to be very different in the real world. Alternatively, we could try to model the dynamics of a sufficiently large portion of the city, but this would surely result in a very slow simulator.

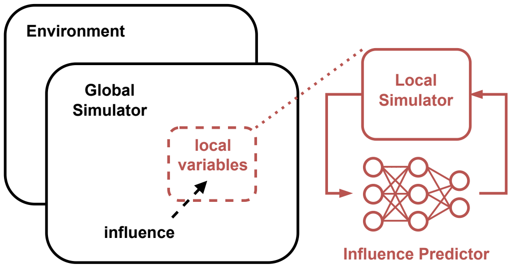

Figure 1: Diagram Influence-Augmented Local Simulator

One important property of the traffic domain is that, although the agent's local problem may be *affected* by many external variables, it is only *directly influenced* by the road segments that connect the intersection with the rest of the city. Hence, we can simply monitor the traffic densities at these road segments since, from the agent's local perspective, they summarize the effect of all the external variables. This insight is not specific to the traffic domain. In fact, in most networked systems (warehouse commissioning, [2], heating systems [4], telecommunication [11]) interactions between different components often occur through a few number of variables.

Contributions: Supported by the formal framework of *influence-based abstraction* (IBA) [8], we exploit the above property to build influence-augmented local simulators (IALS), which mirror the response of the global system through the so called *influence predictor*. In previous work [5] we demonstrated the advantage of this approach for online planning in two discrete toy problems. Here we extend the method to high dimensional problems and study how to integrate the IBA framework with Deep RL. Moreover, while in [5] we showed that the IALS outperforms the global simulator only when the time budget is limited, our results reveal that the IALS can train policies in a fraction of the time and that these can match the same performance as policies trained on the GS, without imposing any time constraints, and despite the IALS is only approximate.

2 INFLUENCE-AUGMENTED LOCAL SIMULATORS FOR DEEP RL

In the following, we describe how we use the IBA formulation to design lightweight simulators that can speed up the long training times imposed by neural network policies. Figure 1 shows a diagram of the influence-augmented local simulator (IALS) [5], which is composed of a *local simulator* and an *approximate influence predictor* (AIP). Please refer to the full version of this paper [12] for a more detailed description of the method.

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online.

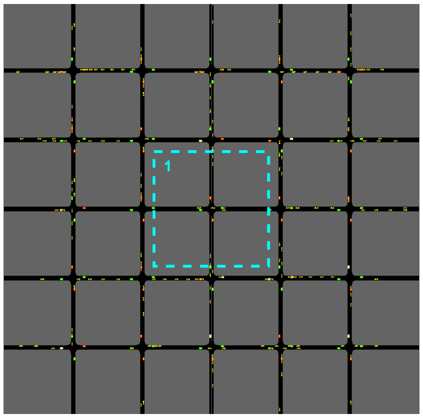

Figure 2: A screenshot of the traffic environment.

***Local simulator (LS):*.** As opposed to a global simulator (GS), which should closely reproduce the dynamics of every state variable, the LS is an abstracted version that only models a small portion of it. The LS focuses on characterizing the transitions of those state variables x_t that are direct parents of rewards and observations. Moreover, as mentioned in the introduction, although local state transitions may be affected by many external (non-local) variables y_t, in many structured settings local variables will only be *directly influenced* by a subset of those which we call influence sources u_t. Hence, we can simulate local transitions as $\dot{T}(x_{t+1}|x_t, u_t, a_t)$ where $\dot{T}$ denotes the local simulator.

***Approximate influence predictor (AIP):* .** The AIP monitors the response of the the external variables y_t to the current action-local state history (ALSH) $l_t = \langle x_1, a_1 ..., a_{t-1}, x_t \rangle$ by estimating $I(u_t|l_t)$. Due to combinatorial explosion, computing the exact distribution $I(u_t|l_t)$ is generally intractable [8]. We write $\hat{I}_\theta$ to denote the AIP, where θ are the parameters, which need to be learned from data. Replacing the true influence distribution with an approximation implies that we are no longer guaranteed to find the optimal policy [3]. Nonetheless, as we show in our experiments, it is often worth trading accuracy for computational efficiency. We model $\hat{I}_\theta$ using a neural network, which we train on a dataset of N samples of the form (l_n, u_n) collected from the GS. We formulate the task as a classification problem. The neural network is optimized using the expected cross-entropy loss.

3 EXPERIMENTS

The goal of this experiments is to study whether we can reduce training times by replacing the GS with the IALS while still achieving comparable learning performances. Agents are trained separately with PPO [10] on (1) the global simulator (GS), (2) the influence-augmented local simulator (IALS) with a pretrained AIP, and (3) an IALS with an untrained AIP (untrained-IALS). To measure the agent's performance, training is interleaved with periodic evaluations on the GS. The results are averaged over 5 random seeds.

Figure 2 shows a grid-like traffic network composed of 25 intersections. The agent controls the traffic lights at the intersection highlighted by the blue dashed box. The rest of the traffic lights are controlled by fixed actuators that use sensors to adapt to the traffic Wu et al. [13]. The goal is to maximize the average speed of cars entering the intersection's neighborhood. Cars are only visible to the agent when they enter the dashed box.

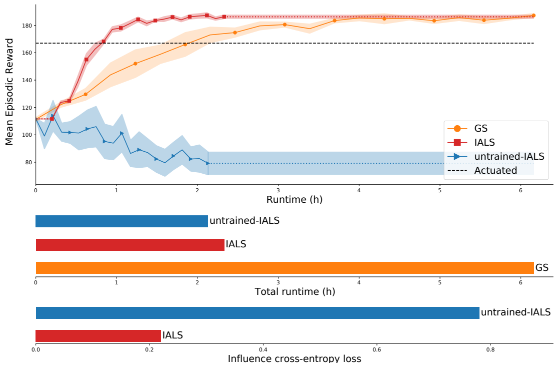

Figure 3: Top: Learning curves of agents trained with the GS, the IALS and the untrained-IALS as a function of wall-clock time. Middle: Total runtime of training for 2M training steps on the three simulators. Bottom: Cross entropy loss.

***GS, LS and AIP:*.** The GS and LS are built using Flow [13] and SUMO [6]. The GS simulates the entire traffic grid while the LS only models the local neighborhood of the intersection. The influence sources u_t are binary variables indicating whether or not a car will be entering the simulation from each of the four incoming lanes at the current timestep. The AIP $\hat{I}_\theta$ is a feedforward neural network trained offline on a dataset of (l_t, u_t) pairs collected from the GS.

Results: The plot at the top of Figure 3 are the learning curves of agents trained with the GS, the IALS, and the untrained-IALS. The plot shows the mean episodic reward as a function of real wall-clock time. Agents are trained for 2M timesteps on all three simulators. The dotted horizontal lines at the end of the red and blue curves show the agent's final performance. The short horizontal line at the beginning of the red curve represents to the AIP's training time. The black horizontal line indicates the performance of the actuated traffic light controller. The two bar charts at the bottom show the total training time when using each of the three simulators, and the AIP's accuracy with and without training. The results suggest that policies trained on the IALS (red) can match the performance of those trained on the GS (orange) in about 1/3 of the total training time, despite the IALS is not as accurate as the GS. On the other hand, since the distribution $\hat{I}_\theta(u_t|l_t)$ induced by the untrained AIP is very different from the true distribution $I(u_t|l_t)$, as evidenced by the high cross entropy loss (blue bar bottom chart), agents trained on the untrained-IALS (blue) perform much worse. More experiments can be found in the full version [12].

ACKNOWLEDGEMENTS

This project received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 758824 —INFLUENCE).

REFERENCES

[1] Martin Arjovsky. 2021. Out of Distribution Generalization in Machine Learning. *arXiv preprint arXiv:2103.02667* (2021).

[2] Daniel Claes, Frans Oliehoek, Hendrik Baier, Karl Tuyls, et al. 2017. Decentralised online planning for multi-robot warehouse commissioning. In *Proceedings of the 16th international conference on autonomous agents and multiagent systems*. 492–500.

[3] Elena Congeduti, Alexander Mey, and Frans A. Oliehoek. 2021. Loss Bounds for Approximate Influence-Based Abstraction. In *AAMAS21*.

[4] Anchal Gupta, Youakim Badr, Ashkan Negahban, and Robin G Qiu. 2021. Energy-efficient heating control for smart buildings with deep reinforcement learning. *Journal of Building Engineering* 34 (2021), 101739.

[5] Jinke He, Miguel Suau, and Frans Oliehoek. 2020. Influence-Augmented Online Planning for Complex Environments. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 4392–4402.

[6] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. 2018. Microscopic Traffic Simulation using SUMO, In The 21st IEEE International Conference on Intelligent Transportation Systems. *IEEE Intelligent Transportation Systems Conference (ITSC)*. https://elib.dlr.de/124092/

[7] Elise van der Pol and Frans A. Oliehoek. 2016. Coordinated Deep Reinforcement Learners for Traffic Light Control. Submitted to NIPS'16 Workshop on Learning, Inference and Control of Multi-Agent Systems.

[8] Frans Oliehoek, Stefan Witwicki, and Leslie Kaelbling. 2021. A sufficient statistic for influence in structured multiagent environments. *Journal of Artificial Intelligence Research* 70 (2021), 789–870.

[9] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. 2009. *Dataset shift in machine learning*. The MIT Press.

[10] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[11] Miguel Suau, Alexandros Agapitos, David Lynch, Derek Farrell, Mingqi Zhou, and Aleksandar Milenovic. 2021. Offline Contextual Bandits for Wireless Network Optimization. *arXiv preprint arXiv:2111.08587* (2021).

[12] Miguel Suau, Jinke He, Matthijs T. J. Spaan, and Frans A. Oliehoek. 2022. Influence-Augmented Local Simulators: A Scalable Solution for Fast Deep RL in Large Networked Systems. *Preprint* (2022).

[13] Cathy Wu, Aboudy Kreidieh, Kanaad Parvate, Eugene Vinitsky, and Alexandre M Bayen. 2017. Flow: A Modular Learning Framework for Autonomy in Traffic. *arXiv preprint arXiv:1710.05465* (2017).