Accelerating Human-Agent Collaborative Reinforcement Learning

Fotios Lygerakis
University of Texas at Arlington
Arlington, Texas, USA
National Center for Scientific
Research "Demokritos"
Athens, Greece
fotios.lygerakis@mavs.uta.edu

Maria Dagioglou
National Center for Scientific
Research "Demokritos"
Athens, Greece
mdagiogl@iit.demokritos.gr

Vangelis Karkaletsis
National Center for Scientific
Research "Demokritos"
Athens, Greece
vangelis@iit.demokritos.gr

ABSTRACT

In domains such as Human-Robot Collaboration artificial agents must be able to support mutual adaptation and learning. Towards this direction, we use a discrete Soft Actor-Critic agent on a real-time collaborative game with humans. We examine how different allocations of on-line and off-line gradient updates impact the game performance and the total training time. Our results suggest that early allocation of a high number of off-line g/u can accelerate learning while shortening training duration.

CCS CONCEPTS

• **Human-centered computing** → **Collaborative interaction**;
• **Computing methodologies** → **Reinforcement learning**.

KEYWORDS

Human-AI Agent Collaborative learning, Deep RL, Soft Actor-Critic, Off-line gradient updates

ACM Reference Format:
Fotios Lygerakis, Maria Dagioglou, and Vangelis Karkaletsis. 2021. Accelerating Human-Agent Collaborative Reinforcement Learning. In *The 14th PErvasive Technologies Related to Assistive EnvironmentsConference (PETRA 2021), June 29-July 2, 2021, Corfu, Greece.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3453892.3454004

1 INTRODUCTION

Collaborative robots (cobots) are expected to share a common workspace with humans and collaborate with them in industrial work-floors [9, 11], rehabilitation set-ups [3] and many other environments. A crucial robot capability during collaboration is to support mutual learning and adaptation [2]. Deep Reinforcement Learning (RL) methods have recently presented very promising results in real-world learning problems [7], including in Human-Robot Collaboration scenarios [10]. Such frameworks provide the opportunity to study real-time how mutual learning and adaptation between humans and (embodied) Artificial Intelligence (AI) agents develop and which AI or human behaviour aspects can be manipulated to accelerate learning and adaptation.

The purpose of this work is to investigate ways to accelerate collaborative learning between a human and an agent and thus to minimize the time spent by a human collaborator during training. First, we examine two variations of the Soft Actor-Critic [6] training algorithm; one that involves only off-line gradient updates (g/u) in fixed intervals [7] and one that also involves a single g/u after each state transition [10]. Subsequently, we explore the impact of the number of off-line g/u throughout a training. Finally, we provide a graphical RL framework for testing human-agent collaborative settings, similar to the game defined in [10].

2 RELATED WORK

Recently there has been an increasing interest towards human-robot teams working and learning collaboratively to achieve specific goals. Deep RL methods have shown very promising results towards this direction. In [4], sparse non-expert human user feedback is used to help an agent to learn a task. They keep a reward model approximation of the user and train the deep RL agent based on the human preferences. Complex novel behaviors are successfully learned with about an hour of human time. In [1, 12], an interactive policy shaping from human reinforcement signals is proposed in order to collaboratively train an agent's policy, while promoting sample efficiency and avoiding the need for coding a reward function. TAMER outperforms both humans and other RL algorithms with only 15 minutes of training. The Soft Actor-Critic algorithm has also shown impressive results during learning in real-time applications [6, 7] and in real-time human-robot collaborative learning. In [10], a human and a UR3 cobot learn to move a ball to a target by controlling different rotations of a tray. The team manages to learn the task in less than 30 minutes.

All the above scenarios require humans to provide, synchronously or asynchronously, demonstrations and feedback or directly interact with an (embodied) AI agent. Short duration of participation or collaboration is important, considering the mental and physical human load. Towards this direction we explore how several factors of deep RL methods affect the learning and total training duration.

PETRA 2021, June 29-July 2,2021, Corfu, Greece

ACM ISBN 978-1-4503-8792-7/21/06…$15.00
https://doi.org/10.1145/3453892.3454004

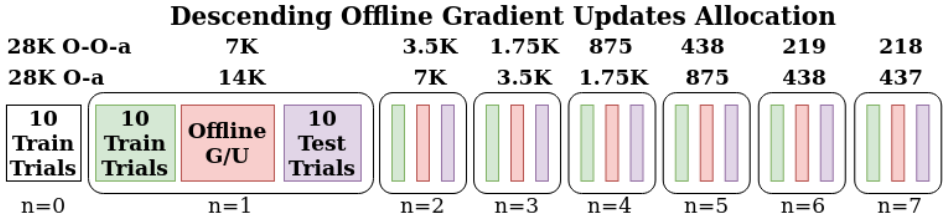

Figure 1: The experimental pipeline and the descending allocation in detail.

3 METHOD

3.1 The virtual environment

Our virtual environment is based on the real-world learning paradigm by [10]. Specifically, we modified a Marble-Maze game[1] which provides a 3D graphical representation of a rotating tray. The human and the agent control the rotation of the tray around one axis each and must collaborate in order to successfully move a ball from a starting point to a goal. The human controls the rotation around the y-axis through the left and right keyboard arrows (counter-clockwise or clockwise rotation). The actions of the agent control the rotation around the x-axis. The integrated game is available at github.com/ligerfotis/maze3d_collaborative.

3.2 RL Set-Up

The state space consists of an 8 dimensional vector that includes the x- and y- linear position and velocity of the ball and the angles and angular velocity of the tray around x and y axes. The agent's action space is 1-dimensional and can take three discrete values -1, 0 and 1 (rotation anti-clockwise, pause or rotation clockwise). The agent is rewarded with 10 when reaching the goal and -1 for each training step passed [10]. The agent is the discrete Soft Actor-Critic (dSAC) algorithm [6]. Our implementation was based on the SLM Lab code [8] and was modified, based on Christodoulou [5] to consider the discrete actions. All the networks of dSAC consist of two hidden layers of 32 nodes each.

3.3 Training Algorithm

We define a *trial* to be the period from the beginning of a game up to either reaching the goal or a maximum of 200 game steps. At every step the actions of the human and the agent a^{agent} change the state s' of the environment and the agent receives a reward r. This transition (s, a^{agent}, r, s') is saved in a replay buffer $\mathcal{D}$. An experiment consists of 70 training trials. Every 10 trials we perform a number of off-line g/u, after which we test the performance of the human-agent over 10 trials. No further training or storing of any transition in $\mathcal{D}$ occurs during these test trials. The score for every testing trial starts from 200 and 1 is subtracted for each game step passed. For each g/u we use a mini-batch of 256 transitions randomly selected from $\mathcal{D}$. We use two variations in the algorithm; one that involves only offline g/u in fixed intervals [7] (Offline algorithm (O-a)) and one that also involves a single g/u after each state transition [10] (Online-Offline algorithm (O-O-a)). In both cases, an offline g/u session is performed every 10 training trials and the total number of g/u is the same (Figure 1).

3.4 Off-line Gradient Updates Allocation

Offline g/u were either distributed evenly across each experiment or followed a *descending allocation* using geometric progression with 1/2 ratio (Figure 1). The latter is expected to accelerate learning while minimizing the total training time. In order to have the same number of total off-line g/u for both on-line and off-line algorithms, we add 2K g/u at each interval of the off-line algorithm, which corresponds to $max\_game\_train\_steps * updates\_interval$. For the first approach, we experimented with two different numbers of total g/u: a)154K (off-line session: 22K in O-a and 20K in O-O-a) and b)28K (off-line session: 4K in O-a and 2K in O-O-a). In the second approach, we only used 28K total g/u distributed with descending allocation.

4 RESULTS

Two participants[2] completed three independent runs of each condition of O-a and O-O-a. Figure 2 shows the mean standard error of the mean (SEM) of the human-agent performance for the different numbers of off-line and of on-line g/u (average of 6 independent runs - three for each participant.) A higher number of off-line g/u

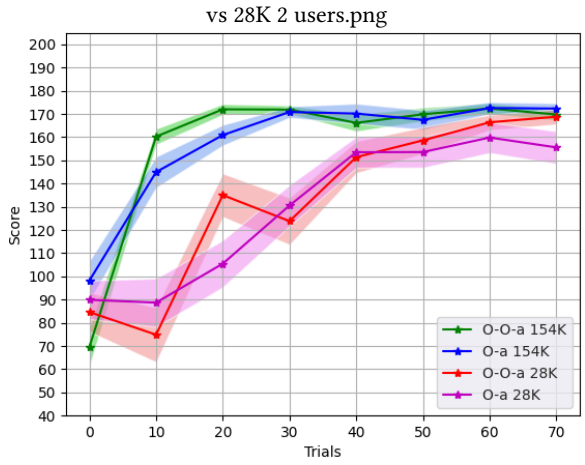

Figure 2: Mean and SEM for 154K vs 28K total g/u. First ten test trials (0) are played with a random agent.

[1] https://github.com/amengede/Marble-Maze

[2] Two different GPUs (GeForce GTX 1050 and a Quadro RTX 4000) were used by each participant.

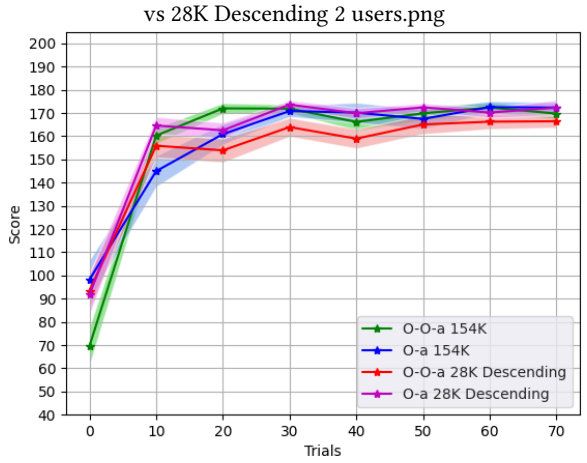

Figure 3: Mean and SEM for 154K vs 28K descending total g/u. First ten test trials (0) are played with a random agent.

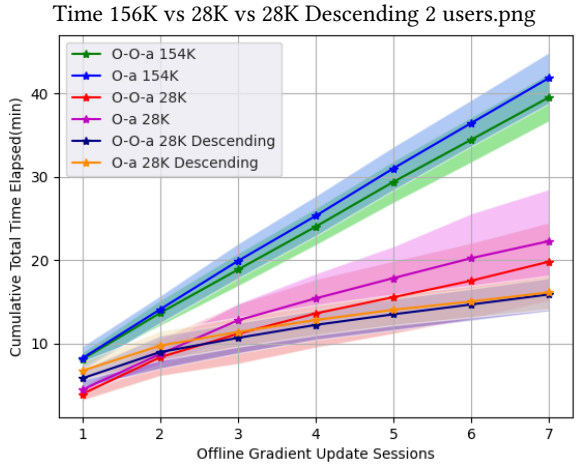

Figure 4: Mean and SEM of Cumulative Total Experiment Time

supports faster and more stable learning. Moreover, adding on-line g/u also seems to lead to higher scores earlier during the training.

Naturally, the higher the number of total g/u, the longer the training duration will be. However, we would like to accelerate the mutual learning while keeping training duration relative short. Towards this, we experimented with variable off-line g/u combining the high performance supported by the higher number of off-line g/u in the beginning of the learning and shortening the total training duration by using a lower number of g/u towards the end of the training.

Figure 3 presents the results of using the descending update allocation (Section 3.4) and compares them with the case of the higher number of off-line g/u in Figure 2. It appears that the descending allocation shows similar learning characteristics with that of the 154K off-line g/u in considerably less total training time (~ 25 minutes less - see Figure 4); there is an early convergence to a high performance and the behaviour does not vary greatly within each testing session. The mean cumulative total time of using descending allocation is even less than of the normal allocation with the same number of total offline g/u, by a factor of 20% and 28%, for O-O-a 28K and O-a 28K respectively.

5 CONCLUSION

Our results indicate that allocating a higher number of off-line g/u early in the training can accelerate learning while minimizing the total training time. Moreover, interpolating on-line g/u after each state transition also appears to accelerate learning. Although the time of interacting with the agent does not change in the different experiments, human collaborator idle periods during off-line g/u is greatly decreased. Naturally, our results are limited by the number of repetitions of each experiment and the number of participants. In the future we plan to repeat the experiments with more participants and expand them to test how other factors such as different reward functions can affect the overall learning.

ACKNOWLEDGMENTS

This work was supported in part by Software and Knowledge Engineering (SKEL) Lab, Institute of Informatics and Telecommunications (IIT), National Center for Scientific Research "Demokritos" (NCSR-D). This work was also supported by the "Stavros Niarchos Foundation" Industrial Post-doc Fellowship on Human-Robot Collaboration: human collaborator representation for robot autonomous decisions, Roboskel lab, SKEL Lab, IIT, NCSR-D.

REFERENCES

[1] W. Bradley Knox and P. Stone. 2008. TAMER: Training an Agent Manually via Evaluative Reinforcement. In *2008 7th IEEE International Conference on Development and Learning*. 292–297. https://doi.org/10.1109/DEVLRN.2008.4640845

[2] Judith Bütepage and Danica Kragic. 2017. Human-robot collaboration: from psychology to social robotics. *arXiv preprint arXiv:1705.10146* (2017).

[3] Giorgia Chiriatti, Giacomo Palmieri, and Matteo Palpacelli. 2020. *A Framework for the Study of Human-Robot Collaboration in Rehabilitation Practices.* 190–198. https://doi.org/10.1007/978-3-030-48989-2_21

[4] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep Reinforcement Learning from Human Preferences. In *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc., 4299–4307. https://proceedings.neurips.cc/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf

[5] Petros Christodoulou. 2019. Soft Actor-Critic for Discrete Action Settings. (10 2019).

[6] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. *CoRR* abs/1801.01290 (2018). arXiv:1801.01290 http://arxiv.org/abs/1801.01290

[7] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. 2018. Soft Actor-Critic Algorithms and Applications. *CoRR* abs/1812.05905 (2018). arXiv:1812.05905 http://arxiv.org/abs/1812.05905

[8] Wah Loon Keng and Laura Graesser. 2017. SLM Lab. https://github.com/kengz/SLM-Lab.

[9] Danica Kragic, Joakim Gustafson, Hakan Karaoguz, Patric Jensfelt, and Robert Krug. 2018. Interactive, Collaborative Robots: Challenges and Opportunities.. In *IJCAI*. 18–25.

[10] Ali Shafti, Jonas Tjomsland, William Dudley, and Aldo Faisal. 2020. Real-World Human-Robot Collaborative Reinforcement Learning. (03 2020).

[11] Valeria Villani, Fabio Pini, Francesco Leali, and Cristian Secchi. 2018. Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics* 55 (2018), 248 – 266. https://doi.org/10.1016/j.mechatronics.2018.02.009

[12] Garrett Warnell, Nicholas R. Waytowich, Vernon Lawhern, and Peter Stone. 2017. Deep TAMER: Interactive Agent Shaping in High-Dimensional State Spaces. *CoRR* abs/1709.10163 (2017). arXiv:1709.10163 http://arxiv.org/abs/1709.10163