GLIDE-RL: Grounded Language Instruction through DEmonstration in RL

Extended Abstract

Chaitanya Kharyal
Microsoft
Hyderabad, India
chaitanyajee@gmail.com

Sai Krishna Gottipati
AI Redefined
Montreal, Canada
sai@ai-r.com

Tanmay Kumar Sinha
Microsoft Research
Bangalore, India
tanmaysinha18@gmail.com

Srijita Das
University of Michigan-Dearborn
Dearborn, USA
sridas@umich.edu

Matthew E. Taylor
AI Redefined
University of Alberta
Edmonton, Canada
matt@ai-r.com

KEYWORDS

Reinforcement Learning, Curriculum Learning, Grounded Language

ACM Reference Format:
Chaitanya Kharyal, Sai Krishna Gottipati, Tanmay Kumar Sinha, Srijita Das, and Matthew E. Taylor. 2024. GLIDE-RL: Grounded Language Instruction through DEmonstration in RL: Extended Abstract. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 3 pages.

1 INTRODUCTION

A critical capability in the development of complex human-AI collaborative systems is the ability of AI agents to understand the natural language and perform tasks accordingly. However, training efficient Reinforcement Learning (RL) agents grounded in natural language has been a long-standing challenge due to the complexity and ambiguity of the language and sparsity of the rewards, among other factors. Advances in curriculum learning [1, 9], continual learning [7], and language models [2, 4, 8], have all independently contributed to effective training of grounded agents in various environments. Leveraging these developments, and building upon our previous work on curriculum learning in teacher-student settings [6], we present a novel algorithm, Grounded Language Instruction through DEmonstration in RL (GLIDE-RL) that introduces a teacher-instructor-student curriculum learning framework. This three-agent setting lets an RL agent learn to follow natural language instructions that can generalize to new tasks and even to novel language instructions.

2 PROBLEM SETUP AND ALGORITHM

The proposed framework of GLIDE-RL is illustrated in Figure-1. The final objective is to have a goal conditioned RL agent (Student) capable of following the natural language instructions in a simulated environment with sparse reward. We have three types of agents: *Teacher, Instructor and Student*. The teacher and student agents are trained in an adversarial setup. While the *student* is a goal-conditioned RL agent that aims to complete the tasks provided to it as natural language instructions, *teachers* are trained to propose tasks/goals by acting in the environment that student agent cannot achieve — this results in teachers providing a curriculum of incrementally harder goals for the student agent to train on. We train multiple teacher agents to assist in better generalization of the student agent by proposing diverse goals. Note that the Teacher agents by themselves are not capable of describing what they have done or instructing the agents. It is the role of the *instructor* agent to describe the teacher's trajectory or key events in natural language and then convert it to a form of instruction for the student agent to act and train upon. Formalizing our **problem set-up** as below:
Given: A student agent S, a set of teacher agents $\{T_1, T_2, \cdots, T_N\}$ and an instructor agent I
To-do: Learn an optimal goal-conditioned policy π_S for the student agent that can follow natural language instructions generated by I by observing the evolving teacher policies $\{\pi_{T_1}, \pi_{T_2}, \cdots, \pi_{T_N}\}$
Assumptions: We make the following assumptions (1) All the teacher agents start from scratch with a random policy and only learn from feedback (reward) related to the student's performance (2) Instructor agent is capable of describing the actions of the teacher in natural language and is equipped with a pre-trained LLM to convert these descriptions to several synonymous instructions.

Algorithm: In every student-teacher rollout, one of the teachers T_i acts in the environment by choosing an action according to its policy π_{T_i} based on the current observation until the end of episode (of predefined length). The trajectory of the teacher is then used by the Instructor to describe in natural language the course of events that the teacher has triggered. In one episode rollout, the teacher could trigger multiple events. The instructor first describes these events in natural language (e.g., "you are standing in-front of red ball") and then converts this description to the form of an instruction (e.g., "go to the red ball"). It then uses a pre-trained language model ϕ_I to generate m synonymous instructions. These events then become the goal for the student agent. Events are

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

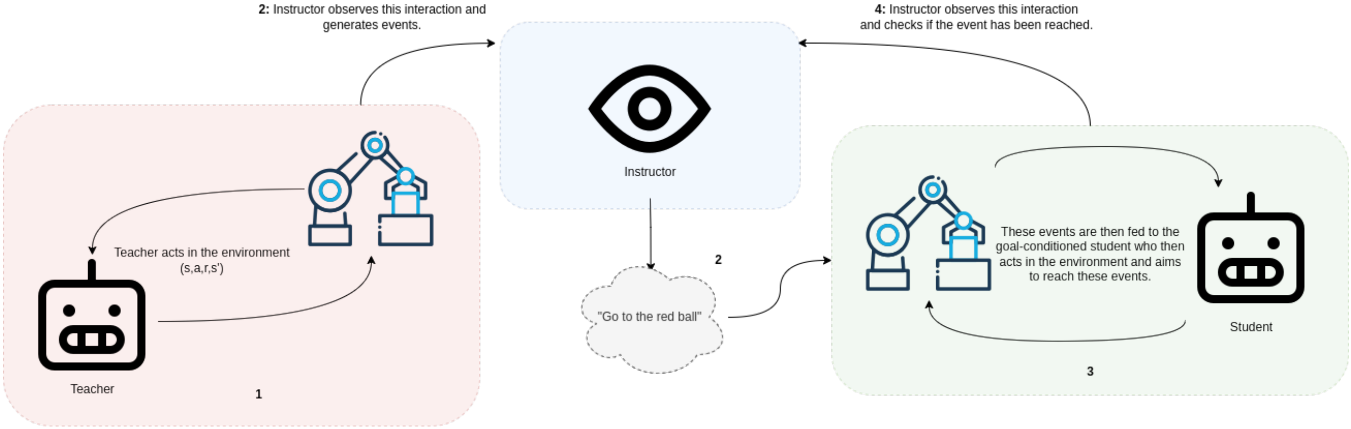
2: Instructor observes this interaction and generates events.
4: Instructor observes this interaction and checks if the event has been reached.
Instructor
Teacher acts in the environment (s,a,r,s')
Teacher
1
2
"Go to the red ball"
These events are then fed to the goal-conditioned student who then acts in the environment and aims to reach these events.
Student
3

Figure 1: GLIDE-RL framework with three independently functioning parts: ***the teacher, instructor and student.***

fed to the student agent one at a time, in the exact same order as the teacher has triggered them, in the form of natural language instructions. Thus, in addition to its current observation, the goal-conditioned student agent also takes in randomly sampled task/goal from the instruction set. The language model ϕ_L transforms the natural language instruction to an embedding (a tensor) which is then concatenated with the input observation. This combined input representation is then passed through Deep-Q-Network to obtain the Q-values for every action. The action corresponding to the maximum Q-value is chosen. At each time step, the student gets 0 reward if it doesn't finish the task/goal. If it finishes, it gets a positive reward. The student continues to act in the environment until it finishes all the goals or until the maximum episode length. The student and the teacher network is updated using standard D3QN loss [5]. An additional behavior cloning loss [10] is used to update the student agent.

3 RESULTS

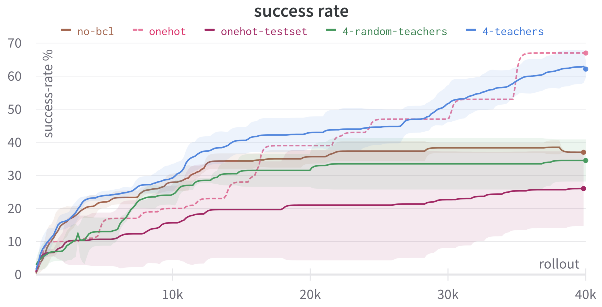

Figure 2: Success rate of different variations on the test set. The plot shows the mean and standard deviation over 5 seeds

Firstly, we train a student conditioned on one-hot goals (`onehot` in figure 2) as a baseline. Teachers' functionality doesn't change here. But, for the student, instead of receiving language embeddings from a language model as inputs, it receives pre-designed one-hot encoding for each event. This baseline gives us an estimate of the upper bound of success rate achievable. Also note that, with one-hot encodings, the agent does not have any generalization capabilities as the size of the encodings cannot be increased to accommodate the unseen goals.

Next, we train the student conditioned on the embeddings from the language model using GLIDE-RL (`4-teachers`). We present the student with the synonymous events while training as described before. The aim of this is to gauge how well can the student perform on the test set as compared to previous baseline. To show the importance of the curriculum generated by the teachers, we train a student with random teacher agents (`4-random-teachers`), and another one-hot student trained directly on the test set but without any teachers or curriculum (`onehot-testset`). While the random-teachers don't learn adversarially with the student, and hence provide no curriculum, the onehot-testset baseline doesn't have the notion of teachers. We introduced onehot-testset baseline to understand how challenging the task is without a curriculum set by the teachers. To understand the necessity of Behavioral Cloning Loss (BCL), we train another baseline (`no-bcl`).

Figure-2 shows the importance of teachers (and curriculum) for the student's performance as the students trained without the teachers' curriculum fail to perform well (measured in terms of success rate), even when trained directly on the test set. Furthermore, we see that the student with goals conditioned as language embeddings is able to perform comparable to the one with one-hot goals (upper-bound). Moreover, with the no-bcl baseline, we establish the importance of the behavioural cloning loss during the training process. As future work, we would like to extend this work to more complex environments including training the instructor agent. Part of the experiments were run using the Cogment [3, 11]. For more detailed experimental results and video demonstrations, **visit our project page at https://ai-r.com/research/gliderl**

ACKNOWLEDGMENTS

Part of this work has taken place in the Intelligent Robot Learning (IRL) Lab at the University of Alberta, which is supported in part by research grants from the Alberta Machine Intelligence Institute (Amii); a Canada CIFAR AI Chair, Amii; Compute Canada; Huawei; Mitacs; and NSERC.

REFERENCES

[1] Yuqing Du, Pieter Abbeel, and Aditya Grover. 2022. It Takes Four to Tango: Multiagent Selfplay for Automatic Curriculum Generation. In *10th International Conference on Learning Representations, ICLR 2022*. 1515–1528.

[2] Yuqing Du, Olivia Watkins, Zihan Wang, Cédric Colas, Trevor Darrell, Pieter Abbeel, Abhishek Gupta, and Jacob Andreas. 2023. Guiding Pretraining in Reinforcement Learning with Large Language Models. arXiv:2302.06692 [cs.LG]

[3] Sai Krishna Gottipati, Luong-Ha Nguyen, Clodéric Mars, and Matthew E. Taylor. 2023. Hiking up that HILL with Cogment-Verse: Train & Operate Multi-agent Systems Learning from Humans. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems* (London, United Kingdom) *(AAMAS '23)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 3065–3067.

[4] Prasoon Goyal, Scott Niekum, and Raymond J Mooney. 2019. Using natural language for reward shaping in reinforcement learning. *arXiv preprint arXiv:1903.02020* (2019).

[5] Ying Huang, GuoLiang Wei, and YongXiong Wang. 2018. V-D D3QN: the Variant of Double Deep Q-Learning Network with Dueling Architecture. In *2018 37th Chinese Control Conference (CCC)*. 9130–9135. https://doi.org/10.23919/ChiCC.2018.8483478

[6] Chaitanya Kharyal, Tanmay Sinha, Sai Krishna Gottipati, Fatemeh Abdollahi, Srijita Das, and Matthew E. Taylor. 2023. Do As You Teach: A Multi-Teacher Approach to Self-Play in Deep Reinforcement Learning. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems* (London, United Kingdom) *(AAMAS '23)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2457–2459.

[7] Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. 2022. Towards continual reinforcement learning: A review and perspectives. *Journal of Artificial Intelligence Research* 75 (2022), 1401–1476.

[8] Jelena Luketina, Nantas Nardelli, Gregory Farquhar, Jakob Foerster, Jacob Andreas, Edward Grefenstette, Shimon Whiteson, and Tim Rocktäschel. 2019. A survey of reinforcement learning informed by natural language. In *International Joint Conference on Artificial Intelligence (IJCAI-19)*.

[9] Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E. Taylor, and Peter Stone. 2020. Curriculum Learning for Reinforcement Learning Domains: A Framework and Survey. *Journal of Machine Learning Research* 21, 181 (2020), 1–50. http://jmlr.org/papers/v21/20-212.html

[10] OpenAI OpenAI, Matthias Plappert, Raul Sampedro, Tao Xu, Ilge Akkaya, Vineet Kosaraju, Peter Welinder, Ruben D'Sa, Arthur Petron, Henrique P. d. O. Pinto, Alex Paino, Hyeonwoo Noh, Lilian Weng, Qiming Yuan, Casey Chu, and Wojciech Zaremba. 2021. Asymmetric self-play for automatic goal discovery in robotic manipulation. arXiv:2101.04882 [cs.LG]

[11] A. I. Redefined, Sai Krishna Gottipati, Sagar Kurandwad, Clodéric Mars, Gregory Szriftgiser, and François Chabot. 2021. Cogment: Open Source Framework For Distributed Multi-actor Training, Deployment & Operations. *CoRR* abs/2106.11345 (2021). arXiv:2106.11345 https://arxiv.org/abs/2106.11345