Improving Policy Generalization for Teacher-Student Reinforcement Learning

Gong Xudong[1], Jia Hongda[1], Zhou Xing[1], Feng Dawei[1], Ding Bo[1](✉), and Xu Jie[2]

[1] National University of Defense Technology, Changsha, China
dingbo@nudt.edu.cn
[2] University of Leeds, Leeds, UK

Abstract. Teacher-student reinforcement learning is a popular approach that aims to accelerate the learning of new agents with advice from trained agents. In these methods, budgets are introduces to limit the amount of advice to prevent over-advising. However, existing budget-based methods tend to use up budgets in the early training stage to help students learn initial policies fast. As a result, initial policies are some kind solidified, which is not beneficial for improving policy generalization. In this paper, to overcome advising intensively in the early training stage, we enable advising in the entire training stage in a decreasing way. Specifically, we integrate advice into reward signals and propose an advice-based extra reward method, and integrate advice into exploration strategies and propose an advice-based modified epsilon method. Experimental results show that the proposed methods can effectively improve the policy performance on general tasks, without loss of learning speed.

Keywords: Reinforcement learning · Agent training · Advising strategy · Policy generalization

1 Introduction

Multi-agent reinforcement learning (MARL) [3] has been widely used in dynamic learning problems in multi-agent systems (MAS) and has gained considerable success in real time strategy games, e.g. DOTA2 [10]. In the application of MARL, new agents should be deployed to extend system capability or to replace failed agents. In these situations, the system ability to resume is determined by how fast newly entering agents can learn their policies. Thus, researches on how to speed up the learning of newly entering agents are vital challenges in MAS.

Teacher-student reinforcement learning [14] was proposed to meet the above challenges. In these methods, an experienced "teacher" agent helps accelerate the "student" agents learning by providing advice on which action to take next. [1]. Besides, helping students learn policies with strong generalization [11] should

G. Li et al. (Eds.): KSEM 2020, LNAI 12275, pp. 39–47, 2020.
https://doi.org/10.1007/978-3-030-55393-7_4

also be considered, otherwise, students can copy teacher policies immediately. Therefore, budgets are introduced to constrain the amount of advice [12,13]. However, existing budget-based methods tend to use up budgets in the early training stage, which means advising imposes an intensive impact to students' exploration in the early training stage. This leads to that students learn relatively solid initial policies fast, which, however, are not beneficial for learning policies with strong generalization. Similar inspiration can be found in pedagogy: if students follow the guidance too much in the early learning stage, they may lack the motivation for change and innovation in the future [2].

The main idea of this paper highlights that advising should be enabled in the entire training stage in a decreasing way, so that advice can provide students continuous reference to learn better policies. Based on this idea, we investigate the framework of reinforcement learning and find that reward signals and exploration strategies are two functional units that take effect in the entire training stage. Thus, we propose the advice-based extra reward (ER) method where we extend reward signals by providing a student with an extra reward if he selects an action that is similar to advice. And propose the advice-based modified epsilon (ME) method where we modify exploration strategies by asking for advice with a descending probability when a student decides to explore the environment.

We test the two proposed methods on the coordinated multi-agent object transportation problem (CMOTP) [3] and a variation of the CMOTP, that is, the r-CMOTP. Comparisons conducted with state-of-the-art advising strategies show that the two proposed methods can improve policy performance on general tasks effectively, without loss of learning speed.

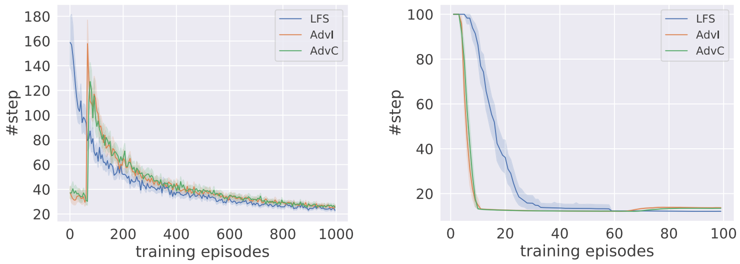
The remainder of this paper is organized as follows. Section 2 presents the necessary background and related works. Section 3 introduces the advice-based ER and ME method. Section 4 compares the proposed methods with state-of-the-art methods on the CMOTP and r-CMOTP. Section 5 concludes the paper.

2 Background and Related Work

2.1 Motivated Scenario

As over advising hinders student learning [13], existing advising methods are generally designed with budgets [8,13] to limit the amount of advice. We apply existing budget-based methods to the CMOTP [3] (detailed in Sect. 4.1). For convenience, a CMOTP **task** refers to two agents allocated to certain specific positions aiming to transport goods to a home area. Different tasks are marked by different initial positions of the two agents.

Figure 1 shows the results when initial positions of the two agents are fixed in the training and testing. Figure 1(a) indicates that budgets are used in the early training stage, as students can finish a training episode quickly in the early stage and slowly in middle and late stages, while Fig. 1(b) illustrates that students can perform well in this specific task even when budgets have not been depleted. This process indicates that advice in budget-based methods takes effect in the early training stage to help students learn initial policies.



(a) performance in training

(b) performance in testing

Fig. 1. LFS means learning from scratch without advice, AdvI is the abbreviation of the advice importance strategy [13], and AdvC is the abbreviation of the advice correct strategy [13], both are budget-based methods. The y-axis represents the mean step that agents take to finish the task in testing, while the x-axis represents the amount of training episodes that have been used to train the policy. We terminate an episode if the episode length exceeds 100 in the tests. The faster a curve drops, the faster a student learns its policy.

In further experiments (detailed in Sect. 4.2), we find that training after budgets are depleted worsens policy performance on specific tasks, but is necessary to enhance policy performance on general tasks. However, as our experimental results demonstrate, policy performance on specific tasks and policy performance on general tasks obtained by budget-based methods are both worse than method without advising when training finished. These results suggest that although it is able to accelerate the learning of students, budget-based methods suffers from a low policy generalization problem.

2.2 Teacher-Student Reinforcement Learning

Teacher-student reinforcement learning was proposed by Clouse et al. [4]. The two roles in these methods are teachers that have learned their policies, and students that do not have their policies yet. The thought of these methods is that teachers can give students some advice basing on some heuristics (e.g. ask uncertain heuristic [4], advice importance heuristic [13] etc.) to accelerate their learning [5–7,13], but the amount of advice should be limited [13]. Over-advising is a major trouble in this setting, since it may hinder students' learning and consume too many communication resources [13]. When students receive advice, they execute these advice immediately and evolve their policies based on the reward signal from the environment. Generally, teachers and students can take different representing types of environment states and different learning algorithms, but share a common action set [13].

3 The Proposed Method

To improve the policy generalization of students, we propose two methods which enable advising in the entire training stage in a decreasing way. The first one uses an advice-based extra reward to integrate advising into reward signals. The second one employs an advice-based modified epsilon method to integrate advising into exploration strategies.

3.1 Advice-Based ER Method

In reinforcement learning framework, agents evolve their policies with direct feedback from reward function, which plays a vital role in the entire training stage. To distribute advice in the entire training stage in a decreasing way, we extend the reward function by considering information from advice.

Equation 1 shows that when the action chosen either by the exploration strategy or by the evolving action policy equals the advice from a teacher, the teacher provides the student with an extra reward, which is calculated by Eq. 2,

$$r'_{st}(s, \mathbf{a}) = r_{st}(s, \mathbf{a}) + \varphi(s, \mathbf{a}, t), \tag{1}$$

$$\varphi(s, \mathbf{a}, t) = \begin{cases} \omega + \mu e^{-\nu t}, if\, a_{st} = \pi_{tc}(s), \\ 0, else \end{cases} \tag{2}$$

where s is the state of the environment, $\mathbf{a}$ is the joint action of all the agents in the environment, t is the iteration that the policy has been trained, $r_{st}(s, \mathbf{a})$ is the reward function of the student, subscript st denotes the student, and subscript tc denotes the teacher, $\omega \in \mathbb{R}, \mu \in [0, +\infty), \nu \in [0, +\infty)$.

3.2 Advice-Based ME Method

In reinforcement learning framework, exploration strategies also play an important role in the entire training stage in helping agents learn an optimal and strong-generalization policy. To distribute advice in the entire training stage in a decreasing way, we let students ask for advice with a decreasing probability when he uses exploration strategy.

Equation 3 shows that when a student uses exploration strategies to interact with the environment, he asks for advice from a teacher with a specified probability,

$$a_{st} = \begin{cases} \pi_{st}(s), if\, x \in [\varepsilon(t), 1] \\ \pi_{tc}(s), if\, x \in [0, \varepsilon(t))\, and\, x' \in [0, g(t)], \\ \pi_{\varepsilon}(s), else \end{cases} \tag{3}$$

where t is the iteration that the policy has been trained, $x \sim U(0, 1), x' \sim U(0, 1)$ are two random variables, $\varepsilon(t)$ is the exploring probability at iteration t, and $g(t) = \omega + \mu e^{-\nu t}$, where $\omega \in [0, 1], \mu \in [0, 1 - \omega], \nu \in [0, +\infty)$ is the asking-for-advice probability at iteration t when the student explores.

4 Evaluation

4.1 CMOTP Environment and Experimental Settings

We evaluate the proposed methods on the CMOTP [3], and a variation of the CMOTP (the initial positions of the two agents are randomized, denoted as r-CMOTP below). As the main purpose of our work is to study the advising methods, for convenience, we implement independent learners [3,9] with Q-learning algorithm as our test algorithm. Available actions for each agent, state representation, network output, and environment restrictions are all same as [3]. First, we train the two agents in the r-CMOTP. Next, we set one of the trained agents as the teammate and the other as the teacher. For the proposed methods, we set $\omega = 0.01, \mu = 0.09, \nu = 0.0001$ for the ER and $\omega = 0.1, \mu = 0.1, \nu = 0.001$ for the ME. We conduct experiments with these settings.

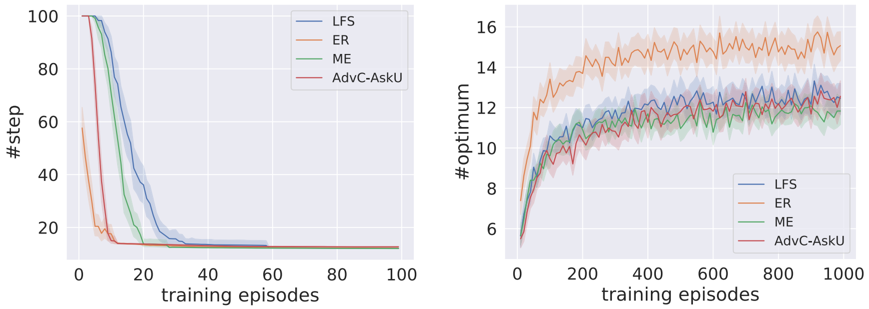
4.2 Evaluation on the CMOTP

We train the student with a specific task, in which the initial positions of the two agents are fixed. We use 100 random seeds in the training to obtain 100 different policies for each method. First, we test the student agent with the same specific task to demonstrate policy quality in this specific task. In the training, we train policy with 10,000 episodes and conduct a test every 10 episodes and terminate a test if the episode length exceeds 100. Figure 2(a) and Table 1 exhibit the corresponding results. Next, we test each policy with 100 different tasks, record the number of tests in which the policy performs optimally. Average value on the 100 different tests is shown to demonstrate the policy generalization. Figure 2(b) and Table 2 demonstrate the corresponding results.

Table 1 exhibits that the policy quality in the specific task is high by the time-point when budgets are depleted for AdvI, AdvC, AskI, and AdvC-AskI. However, this finding does not mean that we can terminate training at this time-point, because Table 2 shows that policy generalization is low for all budget-based methods at this time-point. This result indicates that initial policies learned by budget-based methods are poor in generalization. Consequently, further training is required to enhance policy generalization. However, when the training is complete, the highest policy generalization obtained by the budget-based methods (13.08 by AdvC-AskU) is lower than 13.29 of the LFS method. This finding indicates existing budget-based methods are not beneficial to learn policies with better generalization than LFS.

4.3 Evaluation on the R-CMOTP

We further conduct experiments on a general case, that is, the r-CMOTP, to show differences in policy generalization. In this section, the initial agent position is randomized in the training and testing. We train policy with 55,000 episodes and measure the policy quality by testing each policy with 100 fixed tasks and record the average steps the policy need to complete these tasks. Figure 3(a) and Table 3



(a) Comparison on the steps to finish the task

(b) Comparison on #test where the policy performs optimally out of 100 test

Fig. 2. Comparison among LFS, ER, ME, and AdvC-AskI on the CMOTP. For (a), the y-axis represents the mean step that agents take to finish the task, while the x-axis represents the amount of training episode that has been used to train the policy. For (b), the y-axis represents the number of test in which the policy performs optimally out of 100 test, while the x-axis is same as (a). For both figures, the solid line is the result averaged on 100 different random seeds, and the shaded area is the standard deviation.

Table 1. Comparison on the policy quality on the specific task among different methods[a]

Methods	LFS	AdvI [13]	AdvC [13]	AskI [1]	AskU [4]	AdvC-AskI [1]	AdvC-AskU [1]	ER	ME
#OB[b]	–	89	90	97	14	92	37	–	–
#OT[c]	100	33	32	83	98	90	88	**100**	**100**
#EB[d]	–	62.00	69.11	55.41	55.65	75.66	–	–	–

[a] The initial position of the two agents are fixed in both the training and the testing.

[b] #policy that performs optimally when budgets are depleted (the AdvC-AskU method is tested at 80th episode).

[c] #policy that performs optimally when training is complete.

[d] #episode that has been used to train the policy when budgets are depleted (the AdvC-AskU does not use up budgets by the end of training but rarely uses them since 80 episodes).

Table 2. Comparison on the policy generalization among different methods[a]

Methods	LFS	AdvI	AdvC	AskI	AskU	AdvC-AskI	AdvC-AskU	ER	ME
#OB[b]	–	4.47	5.09	7.21	8.49	6.04	9.04	–	–
#OT[c]	13.29	8.56	8.75	11.50	11.17	12.34	13.08	**15.51**	11.98

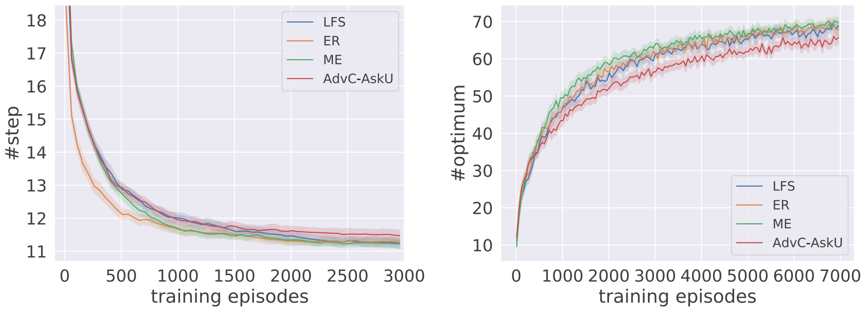
[a] The initial positions of agents are fixed in training and random in testing.

[b] #test where the policy performs optimally when budgets are depleted

[c] #test where the policy performs optimally when training complete.

exhibit the corresponding results. Meanwhile, we measure policy generalization by testing each policy with 100 different tasks and record the number of tasks in which the policy performs optimally (the final value is averaged on 100 random seeds). Figure 3(b) and Table 3 show the corresponding results.

Figure 3(a) shows that ER and ME achieve faster rates in improving policy quality compared with AdvC-AskU. As can be observed in the first row of Table 3, the final average number of steps to finish a task is 9.39 for ER and 9.47 for ME, respectively, both of which are lower than those of budget-based methods. This finding suggests that ER and ME can improve policy quality. Figure 3(b) demonstrates that ER and ME have faster speeds in improving policy generalization compared with AdvC-AskU. In addition, the second row of Table 3 shows that the number of optima a policy obtains in 100 different tasks is 76.06 for ER and 75.77 for ME. Both are higher than LFS and budget-based methods, which indicates that ER and ME can improve policy generalization.



(a) Comparison on the steps to finish the task

(b) Comparison on #test where the policy performs optimally out of 100 test

Fig. 3. Comparison among LFS, ER, ME, and AdvC-AskI on the r-CMOTP. The illustration of (a) is same as Fig. 2(a) while the illustration of (b) is same as Fig. 2(b).

Table 1 demonstrates that the two proposed methods perform best on the specific task of the CMOTP. Meanwhile, Table 2 illustrates that ER achieves the best policy generalization (i.e., 15.51, which is higher than the LFS and budget-based methods). This finding suggests that ER can effectively improve policy generalization on the specific task of the CMOTP. Nevertheless, ME does not perform better than all budget-based methods, the reason may be that for the specific task, exploration is significantly important for students to improve policy generalization, however, asking for advice takes a certain proportion in exploration strategies.

Table 3. Comparison on the policy quality and generalization among different methods[a]

Methods	Optimal	Teacher	LFS	AdvI	AdvC	AskI	AskU	AdvC-AskI	AdvC-AskU	ER	ME
#St[b]	8.88	10.44	9.43	9.60	9.66	9.64	9.54	9.59	9.50	**9.39**	9.47
#Opt[c]	–	73.01	74.85	70.01	72.69	73.66	73.53	74.68	73.81	**76.06**	75.77

[a] The initial position of the two agents are random in both the training and the testing.
[b] The average steps for each policy on 100 specific tasks.
[c] The average number of test in which the policy performs optimally when training complete.

5 Conclusions

This study investigates methods of advising agents to accelerate learning and improve policy generalization. We propose the advice-based extra reward method and the advice-based modified epsilon method and conduct experiments on the coordinated multi-agent object transportation problem. Experimental results show that the proposed methods can effectively improve policy generalization compared with existing methods in the teacher-student reinforcement learning.

References

1. Amir, O., Kamar, E., Kolobov, A., Grosz, B.J.: Interactive teaching strategies for agent training. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, pp. 804–811. AAAI Press (2016)
2. Arnold, K.D.: Academic achievement - a view from the top. The Illinois valedictorian project. Academic Achievement, p. 76 (1993)
3. Buşoniu, L., Babuška, R., De Schutter, B.: Multi-agent reinforcement learning: an overview. In: Srinivasan, D., Jain, L.C. (eds.) Innovations in Multi-agent Systems and Applications - 1. SCI, vol. 310, pp. 183–221. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14435-6_7
4. Clouse, J.A.: On integrating apprentice learning and reinforcement learning (1997)
5. Cruz, F., Magg, S., Weber, C., Wermter, S.: Improving reinforcement learning with interactive feedback and affordances. In: 4th International Conference on Development and Learning and on Epigenetic Robotics, pp. 165–170. IEEE (2014)
6. Cruz, F., Magg, S., Weber, C., Wermter, S.: Training agents with interactive reinforcement learning and contextual affordances. IEEE Trans. Cogn. Dev. Syst. **8**(4), 271–284 (2016)
7. Griffith, S., Subramanian, K., Scholz, J., Isbell, C.L., Thomaz, A.L.: Policy shaping: integrating human feedback with reinforcement learning. In: Advances in Neural Information Processing Systems, pp. 2625–2633 (2013)
8. Ilhan, E., Gow, J., Perez-Liebana, D.: Teaching on a budget in multi-agent deep reinforcement learning. In: 2019 IEEE Conference on Games, pp. 1–8. IEEE (2019)
9. Matignon, L., Laurent, G.J., Le Fort-Piat, N.: Independent reinforcement learners in cooperative Markov games: a survey regarding coordination problems. Knowl. Eng. Rev. **27**(1), 1–31 (2012)
10. OpenAI: Openai five (2018). https://blog.openai.com/openai-five/
11. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (2018)

12. Taylor, M.E., Carboni, N., Fachantidis, A., Vlahavas, I., Torrey, L.: Reinforcement learning agents providing advice in complex video games. Connect. Sci. **26**(1), 45–63 (2014)
13. Torrey, L., Taylor, M.: Teaching on a budget: agents advising agents in reinforcement learning. In: Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems, pp. 1053–1060 (2013)
14. Zimmer, M., Viappiani, P., Weng, P.: Teacher-student framework: a reinforcement learning approach. In: AAMAS Workshop Autonomous Robots and Multirobot Systems (2014)