Heterogeneous Multi-unit Control with Curriculum Learning for Multi-agent Reinforcement Learning

Jiali Chen[1], Kai Jiang[2], Rupeng Liang[2], Jing Wang[2], Shaoqiu Zheng[2(✉)], and Ying Tan[1,3,4,5(✉)]

[1] School of Intelligence Science and Technology, Peking University, Beijing 100871, China
ytan@pku.edu.cn
[2] Nanjing Research Institute of Electronic Engineering, Nanjing 210007, China
zhengshaoqiu1214@foxmail.com
[3] Key Laboratory of Machine Perception (MOE), Peking University, Beijing 100871, China
[4] Institute for Artificial Intelligence, Peking University, Beijing 100871, China
[5] Nanjing Kangbo Intelligent Health Academy, Nanjing 211100, China

Abstract. Heterogeneous Multi-unit control is one of the most concerned topic in multi-agent system, which focuses on controlling agents of different type of functions. Methods that utilize parameter or replay-buffer sharing are able to address the problem of combinatorial explosion under isomorphism assumption, but may lead to divergence under heterogeneous setting. This work use curriculum learning to bypass the barrier of a needle in a haystack that is faced by either joint-action learner or independent learner. According to the experiment on heterogeneous force combat engagements, the independent learner outperforms the baseline learner by 10% of evaluation metrics with curriculum learning, which empirically shows that curriculum learning is able to discover a novel learning trajectory that is not followed by conventional multi-agent learners.

Keywords: Heterogeneous control · Curriculum learning · Multi-agent system

1 Introduction

A series of benchmark has been proposed for determining the performance of multi-agent reinforcement learning, with more agents, more complex agent architecture and sparser rewards indicating better algorithm needed to solve the problem. The StarCraft Multi-Agent Challenge (SMAC [33]) based on the StarCraft

This work is supported by the National Natural Science Foundation of China (Grant No. 62250037, 62276008 and 62076010), and partially supported by Science and Technology Innovation 2030 - 'New Generation Artificial IntelligenceMajor Project (Grant Nos.: 2018AAA0102301 and 2018AAA0100302).

Y. Tan and Y. Shi (Eds.): DMBD 2022, CCIS 1744, pp. 3–16, 2022.
https://doi.org/10.1007/978-981-19-9297-1_1

II Learning Environment (SC2LE [45]) is currently one of the most wildly used benchmark for multi-agent reinforcement learning algorithms, as shown in Fig. 1 However, the performance on SMAC is mostly determined by micromanagement of every single agent rather than collaborative joint actions. Moreover, the agents are with similar action spaces, move and attack, with difference only in attacking range and defense. Yet such setting is still too weak for realistic applications because isomorphism is often violated and requires collaborative joint actions of heterogeneous agents. Besides, tasks are multi-stage and multi-target rather than simply defeating the opponent by elimination.

Fig. 1. A snapshot of 27 m vs. 30 m challenge in SMAC

Recently more complex and realistic environments that focuses on heterogeneous agents are being aggressively explored to develop algorithms for policies that are more robust to the dynamics of environment. Different from environments with isomorphism agents, heterogeneity indicates decision structures with little similarity among agents. Agents have different observation spaces and action spaces, thus, neural networks with fixed sizes of input and output can not be directly applied. Furthermore, the optimal parameters and network structures are unique for every agent, which means that a normal technique that tries to reduce the computational cost by sharing information among agents may fail. For example, as shown in Fig. 2, in a heterogeneous force combat engagement task, the view of a tank may be blocked by a nearby forest, and it is only able to move along the facing direction or turn. However, a helicopter is with higher motility and different action space, a control structure different from that of a tank is needed.

Conventional methods like parameter sharing are reconsidered [41–43] since the assumption of similar decision structures is violated under heterogeneous settings. With carefully redesign of the observation spaces, action spaces and

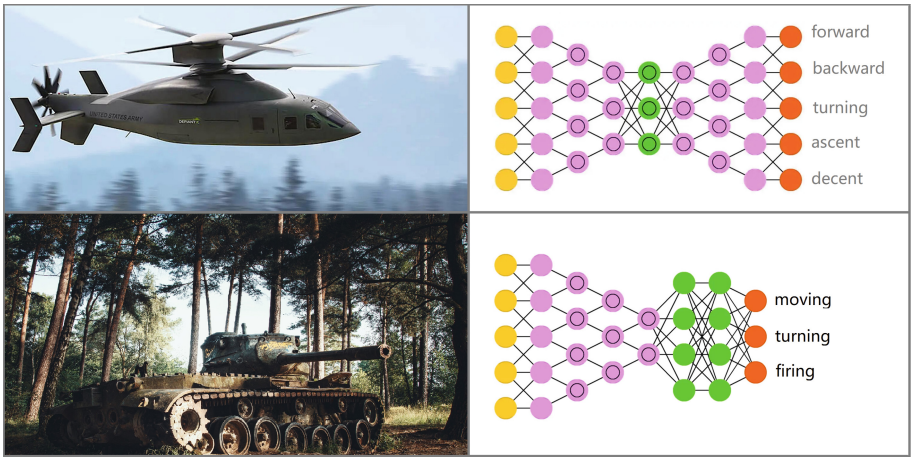

Fig. 2. A tank and a helicopter are heterogeneous agents

neural network structures among all agents, the technique of parameter sharing is able to applied under heterogeneous setting. Meanwhile, this reveals the fact most techniques that developed under the assumption of isomorphism agents failed to directly generalize to heterogeneous setting. As a consequence, methods that fit those benchmarks well are needed to reconsidered if it is over-fitted to isomorphism agents.

2 Related Works

A series of methods is developed under the widely used framework of centralized training and decentralized execution (CTDE [27]). These methods are classified in mainly two classes based on their optimization method, value decomposition or policy gradient.

Value based methods try to fit a value function $Q(s, \boldsymbol{a})$ that is able to evaluate the accumulative rewards of an immediate joint action $\boldsymbol{a}$, by iterating the bellman equation, that is

$$Q(s, \boldsymbol{a}) \leftarrow r + \gamma \max_{\boldsymbol{a}'} Q(s', \boldsymbol{a}') \tag{1}$$

which is a direct utilization of value iteration in single agent reinforcement learning. However, Q function suffers from the curses of dimensions with the increase of the number of agents. To address this problem, alternatives of decomposing the centralized value function are aggressively explored. Linear decomposition [38] assumes that the centralized value function is equal to a linear summation of functions on all agents. QMix [32] consider the monotonic aggregation based on the Individual-Global-Max (IGM) condition. QTRAN [36] claims that monotonic aggregation is not necessary condition for IGM and proposed a method

based on affine transformation, showing better adaptation on complex games. WQMix [31] claims that QMix may lead to divergence or underestimation, and therefore proposed a weighted version of QMix. QPLEX [46] decomposes the value function with dueling structure, and claims that it is able to learn every decomposable value function satisfying the IGM condition.

While policy gradient method optimizes the parameters directly from an estimation of accumulative rewards. To formalize, the gradient of neural network weights

$$\nabla J(\theta) = R(\tau)\nabla P(\tau|\theta) \approx \frac{1}{N}\sum_{n=1}^{N} R(\tau_n)\nabla \ln P(\tau_n|\theta) \tag{2}$$

yet it is also necessary to decompose the gradients onto every agent, which is also known as a credit assignment problem [25]. MADDPG [22] uses a centralized critic network to distribute gradients among training agents. COMA [7] introduces counterfactual baseline to reduce the variance of gradients. MAPPO [51] generalizes proximal policy optimization (PPO [35]) to multi-agent environments with 5 proposed techniques. MATRPO [19] generalizes trust region policy optimization (TRPO [34]) to multi-agent environments to acquire a better theoretical guarantee multi-agent reinforcement learning algorithm.

However, methods based on CTDE framework is often assumed that the agents are isomorphism. Although in benchmarks like SMAC there are multiple races or units in an environment, the goals of the challenges and the decision structures of agents are of little difference. Thus algorithms with significant empirical results on these benchmarks may fail under heterogeneous settings. Thus alternatives focus on avoiding the assumption of isomorphism are widely explored.

Some consider independent training of every agent so that heterogeneous agents don't interfere each other. IQL [39] is proposed to learn individual Q function directly from iterating bellman equation independently on every single agent. IPPO [49] applies PPO algorithm directly to multi-agent environments and shows better empirical results than QMix and IQL on several challenges in SMAC. MA2QL [37] applies a minimal modification on IQL to acquire the theoretical guarantee on converging to a Nash Equilibrium [26]. MABCQ [14] exploits value deviation and transition normalization to modify the transition probability to derive an offline decentralized multi-agent algorithm. Yet the assumption of independence may lead to divergence or rather low sample complexity [22], thus these algorithm may lose scalability to the number of agents.

Some applies modern techniques in reinforcement learning or deep learning to address the problem of heterogeneous agents. Graph neural networks [9,50,53] are introduced to describe the relationships between heterogeneous agents, and HMAGQ-Net [23] proposed a graphical description of multi-agent system. Communication [2,20,24] is introduced to stabilize the training of independent learners, which builds channels between agents to achieve better collaboration. DDDQN [5] introduces 3 techniques in reinforcement learning to solve a heterogeneous traffic light control problem. These methods bypass the problem of heterogeneous agents by more expressive neural structures or training

methods, but may introduce more computational cost or training difficulty that requires carefully fine tuning.

Some methods are inspired by population based training [6,10,16,21,40,52,54]. League training [44] is introduced to find weakness of policies. IQ-algorithm [29] introduces an imitation learner to solve a heterogeneous multi-agent problem. Role based learning [47,48] introduces roles to break isomorphism. These methods address directly to the problem of heterogeneous agents, but may requires prior knowledge of the environment. Others focuses on quantifying heterogeneity. Model-free conventions [17] are considered in heterogeneous settings to encourage exploration. FMQ [15] algorithm is proposed to learn to coordinate heterogeneous agents. And communication heterogeneity [3] is considered to provide an analysis tool to describe and quantify heterogeneity during communication.

3 Method

Here we describe an alternative to address the problem of heterogeneous agents by curriculum learning.

3.1 Preliminaries

Consider how a human student learn skills from class. She starts with learning basic concepts and practicing by solving simple problem. As she gets familiar to the newly learnt knowledge, she turns to practice with more difficult problems to become an expert. This is how Curriculum Learning [1] works. To formalize, the preferred curriculum learning is to search for a task selection function [28] $D : H \rightarrow T$ where H contains information about past interactions and T is the target task, the objective

$$Obj : \max_{D} \int_{T \sim T_{target}} P_T^N \, dT \tag{3}$$

indicates the outcome of curriculum learning, where T_{target} denotes the distribution of target tasks, N denotes the number of training steps and P_T^N denotes the fitness on task T after training N steps.

The way to choose a task selection function is one of the most central problem of curriculum learning. The task selection function can be viewed as a control of training trajectory, as shown in Fig. 3, on the skill potential landscape the two training trajectories A and B are of the same starting and target points. However, trajectory A tries to climb through the cliff, which means a rapid increase on training task. This will result in the agent is not prepared to solve the upcoming problem thus getting stuck at the valley. While trajectory B looks for a tortuous path but with slowly ascending difficulty. Therefore although trajectory B is geometrically longer than trajectory A, it is more training friendly. In particular, when the agent is trained directly from the target task, it means to jump vertically from the starting point to the target and is usually the most

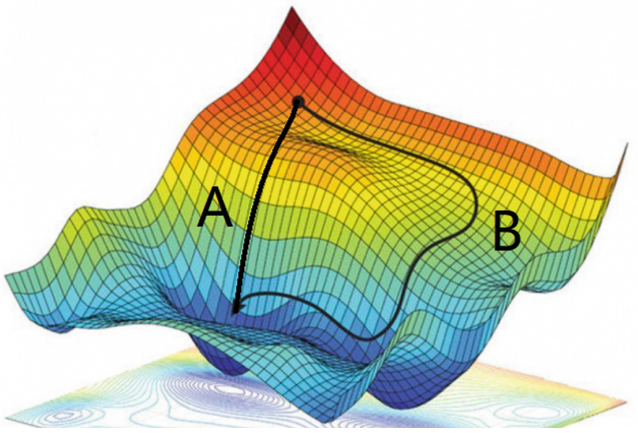

Fig. 3. Different training trajectory means different learning difficulty [13]

difficult trajectory. Therefore, there are tasks that are nearly impossible to learn even with long training time, but can be solved by curriculum learning.

Alternatives to select tasks along training process are broadly explored [28]. BARC [11] determines the initial states to control the difficulties. The reward function is also considered for exploration [4], guidance [8] or intrinsic goals [12]. Others may also consider changing the goal [18,30] during training, which is also known as multi-goal learning.

3.2 Curriculum Learning by Adjusting Opponents

In this work, a curriculum learning task selection by modifying the behaviour of opponents is used, since in heterogeneous force combat engagement problem there are naturally two competitive teams. The difficulty of the task of defeating the opponent can be slowly shifted by interfering the behaviour. Intuitively, consider a talented coach trying to train a teammate by sparring, the coach can conceal his skill in early days and make it all-out when the teammate is trained. Here we proposed two methods that can control the strength of the opponent.

The first method is to blur the observation of the opponent by a vanishing noise, as shown in Fig. 4. To be specific, let the observation of the opponent to be o and now is the j-th round of training, we feed observation

$$o' = o + \frac{N(0, \sigma)}{j} \tag{4}$$

as the input of the opponent. As the decision of the opponent is misled by the noise, it will no longer be a fatal threat that prevents the learner to learn even the basic rules of the game.

Theorem 1. *For a task with continuous action spaces, consider a perfectly trained linear controller G. Under the blurred observation, the distance between*

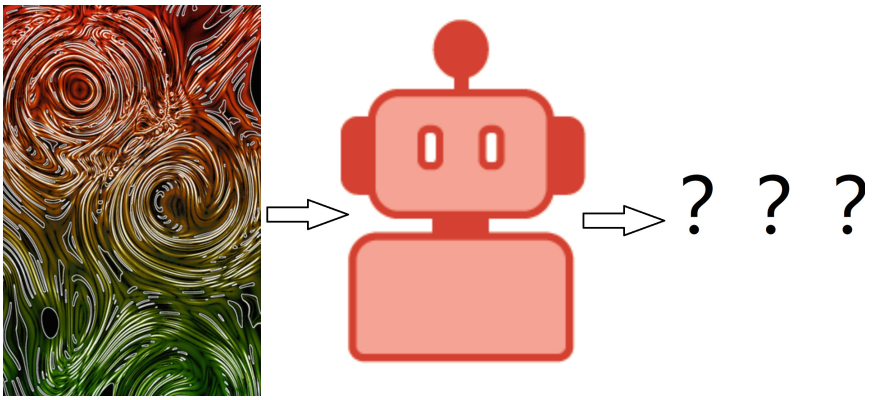

Fig. 4. Control opponent action by blurring observation

the actual behaviour and the best reaction is proportion to a Gaussian with mean 0 *and variance* $\frac{\sigma}{j^2}$.

Proof.

$$||G(o) - G(o')||_1 = ||G(o - o')||_1 \sim ||o - o'|| \sim N(0, \frac{\sigma}{j^2})$$

which indicates that we are able to control an opponent with ascending strength.

The second method is to randomly interfere the action commands of the opponent, which prevent the opponent from acting correctly. For a task with continuous action spaces, the output action a is added by a vanishing Gaussian noise, that is

$$a' = a + \frac{N(0, \sigma)}{j} \tag{5}$$

For a task with discrete action spaces, the action commands are randomly drop with refer to a probability inversely proportion to the number of rounds trained, that is

$$drop(a) = \frac{1}{j} \tag{6}$$

so we derives a method of interfering the behaviour of the opponent, with descending noise away from the best reaction.

4 Experiments

To evaluate the effectiveness of curriculum learning, we adopt the heterogeneous force combat engagements environment.

4.1 Environment and Settings

The task of heterogeneous force combat engagements is to defeat the opponent by destroying the command post. There are multiple agent types including radar, GBAD, destroyer, fighter plane, jammer, bomber and scout. As shown in Fig. 5, agents are highly heterogeneous and the observation is complex.

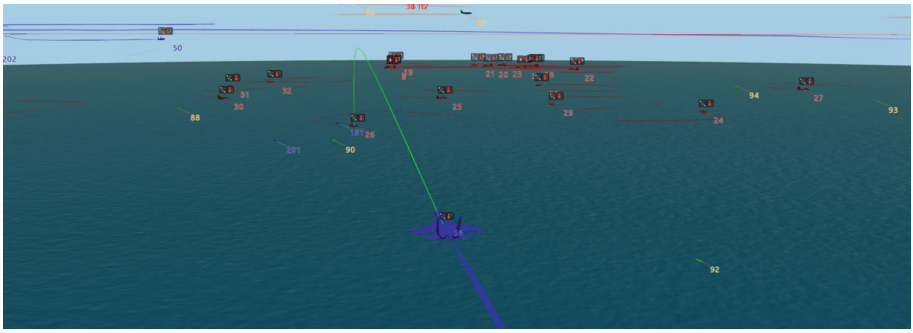

Fig. 5. A snapshot of the environment

Due to limited speed of the simulator, we utilize distributed training with a cluster of 8 GPUs, with each GPU we collect data from 8 parallel environments. And we compare the performance between the backbone multi-agent reinforcement learning algorithm with or without curriculum learning.

To examine the effectiveness of the two proposed methods of curriculum learning, two sets of contrast experiments are carried out. A hierarchical decision framework with high level instructions from the neural network controller and low level execution by a rule based controller is used. Two sets of rules of the executor are used respectively in two experiments to dispel the effects of rule-based controllers.

4.2 Evaluation

To evaluate the performance of the proposed methods, four metrics summarized from the training process are used.

- asymptotic expected rewards $V_{\pi^*}(S_0)$
- maximum expected rewards $\max_t V_{\pi_t}(s_0)$
- time to converge t_*
- time to reach a threshold of rewards λ, t_λ

concepts of the four metrics are visualized in Fig. 6.

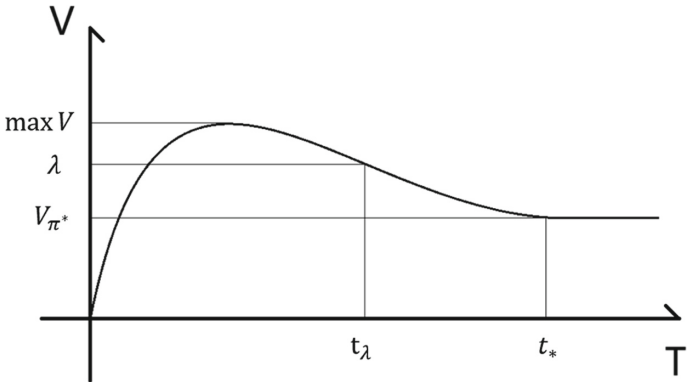

Fig. 6. The metrics used to evaluate an algorithm

4.3 Results

This work first did an experiment on blurring the observation of the opponent. Figure 7 shows the training curves of the baseline algorithm based on rule set A, where 7(a) shows the loss function of PPO and 7(b) shows the expected accumulative rewards.

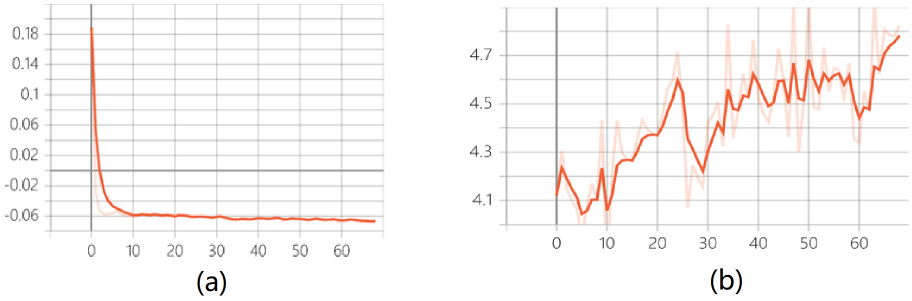

Fig. 7. The loss curve (a) and accumulative rewards (b) of baseline algorithm

As a contrast, Fig. 8 shows the training curves of the curriculum learning based on observation blurring, which also uses the rule set A for its low-level controller. According to the accumulative rewards, the four metrics are summarized in Table 1, where the threshold $\lambda = 4.7$.

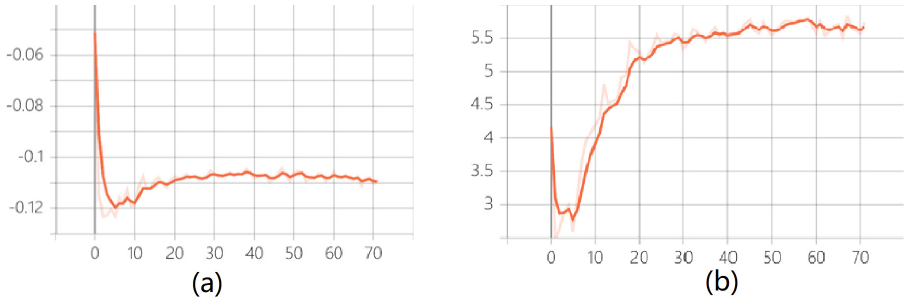

Fig. 8. The loss curve (a) and accumulative rewards (b) of observation blurring

Table 1. Comparison of baseline and observation blurring

Metrics	$V_{\pi*}$	$\max V_{\pi_t}$	t_*	t_λ
Baseline	4.8	4.9	68	62
OB	5.6	5.75	45	18

The second part of the experiment consists of evaluating curriculum learning based on action interference. Figure 9 shows the training curves of the baseline algorithm based on rule set B.

Fig. 9. The loss curve (a) and accumulative rewards (b) of baseline algorithm

While Fig. 10 shows the training curves of curriculum learning based on action interference, which also uses rule set B for low-level control. Table 2 summarizes the metrics of both training process, where $\lambda = 5$.

4.4 Discussion

The experiments reveal that both implementations of curriculum learning, observation blurring and action interference, outperforms the baseline algorithm in

Fig. 10. The loss curve (a) and accumulative rewards (b) of action interference

Table 2. Comparison of baseline and action interference

Metrics	V_{π^*}	$\max V_{\pi_t}$	t_*	t_λ
Baseline	4	7	2000	2000
AI	5	10	2000	1000

policy performance, convergence and sample efficiency. Firstly, as shown in the comparison of $\max V_{\pi_t}$, curriculum learning found policies that gains rewards 10% more than the baseline algorithm. This shows that curriculum learning is able to find better policy, as we are trying to exploit reinforcement learning to find an optimal controller for the problem. Secondly, the comparison of V_{π^*} and t_* also reveals that curriculum learning outperforms the baseline algorithm in asymptotic performance, no matter in convergent point or time to converge. Thirdly, the comparison of t_λ reveals that curriculum learning boosts the sample efficiency for at least 10%. This also in line with expectation, since curriculum learning leverage the training trajectory that is more suitable all along the training process. To sum up, the experiments show a 10% boosting on performance and in turn support the claims this work mentioned above.

5 Conclusion

In this work we proposed an curriculum learning method based on interfering the opponent to solve an heterogeneous force combat. According to the empirical results, the two proposed interfering alternative, observation blurring and action dropping, are both able to achieve a 10% boosting on evaluation metrics.

Although the empirical results show the effectiveness of this method, the theoretically understanding of applying curriculum learning to heterogeneous multi-agent learning problem is still unclear, which we will leave as a future work.

References

1. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: Proceedings of the 26th Annual International Conference On Machine Learning, pp. 41–48 (2009)
2. Bravo, M., Alvarado, M.: On the pragmatic similarity between agent communication protocols: modeling and measuring. In: On the Move to Meaningful Internet Systems: OTM, pp. 128–137 (2008)
3. Bravo, M., Reyes-Ortiz, J.A., Rodríguez, J., Silva-López, B.: Multi-agent communication heterogeneity. In: 2015 International Conference on Computational Science and Computational Intelligence (CSCI), pp. 583–588. IEEE (2015)
4. Burda, Y., Edwards, H., Storkey, A., Klimov, O.: Exploration by random network distillation. arXiv preprint arXiv:1810.12894 (2018)
5. Calvo, J.A., Dusparic, I.: Heterogeneous multi-agent deep reinforcement learning for traffic lights control. In: AICS, pp. 2–13 (2018)
6. Dorigo, M., Birattari, M., Stutzle, T.: Ant colony optimization. IEEE Comput. Intell. Mag. **1**(4), 28–39 (2006)
7. Foerster, J., Farquhar, G., Afouras, T., Nardelli, N., Whiteson, S.: Counterfactual multi-agent policy gradients. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
8. Fournier, P., Sigaud, O., Chetouani, M., Oudeyer, P.Y.: Accuracy-based curriculum learning in deep reinforcement learning. arXiv preprint arXiv:1806.09614 (2018)
9. Henaff, M., Bruna, J., LeCun, Y.: Deep convolutional networks on graph-structured data. arXiv preprint arXiv:1506.05163 (2015)
10. Hu, W., Tan, Y.: Prototype generation using multiobjective particle swarm optimization for nearest neighbor classification. IEEE Trans. Cybern. **46**(12), 2719–2731 (2015)
11. Ivanovic, B., Harrison, J., Sharma, A., Chen, M., Pavone, M.: BARC: backward reachability curriculum for robotic reinforcement learning. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 15–21. IEEE (2019)
12. Jabri, A., Hsu, K., Gupta, A., Eysenbach, B., Levine, S., Finn, C.: Unsupervised curricula for visual meta-reinforcement learning. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
13. Jain, P., Kar, P., et al.: Non-convex optimization for machine learning. Found. Trends Mach. Learn. **10**(3–4), 142–363 (2017)
14. Jiang, J., Lu, Z.: Offline decentralized multi-agent reinforcement learning. arXiv preprint arXiv:2108.01832 (2021)
15. Kapetanakis, S., Kudenko, D.: Reinforcement learning of coordination in heterogeneous cooperative multi-agent systems. In: Kudenko, D., Kazakov, D., Alonso, E. (eds.) AAMAS 2003-2004. LNCS (LNAI), vol. 3394, pp. 119–131. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-32274-0_8
16. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN 1995-International Conference on Neural Networks, vol. 4, pp. 1942–1948. IEEE (1995)
17. Köster, R.: Model-free conventions in multi-agent reinforcement learning with heterogeneous preferences. arXiv preprint arXiv:2010.09054 (2020)
18. Lair, N., Colas, C., Portelas, R., Dussoux, J.M., Dominey, P.F., Oudeyer, P.Y.: Language grounding through social interactions and curiosity-driven multi-goal learning. arXiv preprint arXiv:1911.03219 (2019)

19. Li, H., He, H.: Multi-agent trust region policy optimization. arXiv preprint arXiv:2010.07916 (2020)
20. Liu, C.L., Tian, Y.P.: Formation control of multi-agent systems with heterogeneous communication delays. Int. J. Syst. Sci. **40**(6), 627–636 (2009)
21. Liu, L., Zheng, S., Tan, Y.: S-metric based multi-objective fireworks algorithm. In: 2015 IEEE Congress on Evolutionary Computation (CEC), pp. 1257–1264. IEEE (2015)
22. Lowe, R., Wu, Y.I., Tamar, A., Harb, J., Pieter Abbeel, O., Mordatch, I.: Multi-agent actor-critic for mixed cooperative-competitive environments. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
23. Meneghetti, D.D.R., Bianchi, R.A.C.: Towards heterogeneous multi-agent reinforcement learning with graph neural networks. arXiv preprint arXiv:2009.13161 (2020)
24. Meneghetti, D.D.R., da Costa Bianchi, R.A.: Specializing inter-agent communication in heterogeneous multi-agent reinforcement learning using agent class information. arXiv:abs/2012.07617 (2020)
25. Minsky, M.: Steps toward artificial intelligence. Proc. IRE **49**(1), 8–30 (1961)
26. Nash, J.F., Jr.: Equilibrium points in n-person games. Proc. Natl. Acad. Sci. **36**(1), 48–49 (1950)
27. Oliehoek, F.A., Spaan, M.T., Vlassis, N.: Optimal and approximate q-value functions for decentralized Pomdps. J. Artif. Intell. Res. **32**, 289–353 (2008)
28. Portelas, R., Colas, C., Weng, L., Hofmann, K., Oudeyer, P.Y.: Automatic curriculum learning for deep RL: a short survey. arXiv preprint arXiv:2003.04664 (2020)
29. Price, B., Boutilier, C.: Reinforcement learning with imitation in heterogeneous multi-agent systems
30. Racaniere, S., Lampinen, A.K., Santoro, A., Reichert, D.P., Firoiu, V., Lillicrap, T.P.: Automated curricula through setter-solver interactions. arXiv preprint arXiv:1909.12892 (2019)
31. Rashid, T., Farquhar, G., Peng, B., Whiteson, S.: Weighted QMIX: expanding monotonic value function factorisation for deep multi-agent reinforcement learning. Adv. Neural. Inf. Process. Syst. **33**, 10199–10210 (2020)
32. Rashid, T., Samvelyan, M., Schroeder, C., Farquhar, G., Foerster, J., Whiteson, S.: QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. In: International Conference On Machine Learning, pp. 4295–4304. PMLR (2018)
33. Samvelyan, M., et al.: The StarCraft Multi-Agent Challenge. CoRR abs/1902.04043 (2019)
34. Schulman, J., Levine, S., Abbeel, P., Jordan, M., Moritz, P.: Trust region policy optimization. In: International Conference on Machine Learning, pp. 1889–1897. PMLR (2015)
35. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
36. Son, K., Kim, D., Kang, W.J., Hostallero, D.E., Yi, Y.: QTRAN: learning to factorize with transformation for cooperative multi-agent reinforcement learning. In: International Conference on Machine Learning, pp. 5887–5896. PMLR (2019)
37. Su, K., Zhou, S., Gan, C., Wang, X., Lu, Z.: Ma2QL: a minimalist approach to fully decentralized multi-agent reinforcement learning. arXiv preprint arXiv:2209.08244 (2022)
38. Sunehag, P., et al.: Value-decomposition networks for cooperative multi-agent learning. arXiv preprint arXiv:1706.05296 (2017)

39. Tan, M.: Multi-agent reinforcement learning: independent vs. cooperative agents. In: Proceedings of the Tenth International Conference on Machine Learning, pp. 330–337 (1993)
40. Tan, Y., Zhu, Y.: Fireworks algorithm for optimization. In: Tan, Y., Shi, Y., Tan, K.C. (eds.) ICSI 2010. LNCS, vol. 6145, pp. 355–364. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13495-1_44
41. Terry, J.K., Grammel, N., Hari, A., Santos, L.: Parameter sharing is surprisingly useful for multi-agent deep reinforcement learning (2020)
42. Terry, J.K., Grammel, N., Hari, A., Santos, L., Black, B.: Revisiting parameter sharing in multi-agent deep reinforcement learning. arXiv preprint arXiv:2005.13625 (2020)
43. Terry, J.K., Grammel, N., Son, S., Black, B.: Parameter sharing for heterogeneous agents in multi-agent reinforcement learning. arXiv:abs/2005.13625 (2020)
44. Vinyals, O., et al.: Grandmaster level in StarCraft ii using multi-agent reinforcement learning. Nature **575**(7782), 350–354 (2019)
45. Vinyals, O., et al.: StarCraft ii: A new challenge for reinforcement learning. arXiv preprint arXiv:1708.04782 (2017)
46. Wang, J., Ren, Z., Liu, T., Yu, Y., Zhang, C.: Qplex: duplex dueling multi-agent q-learning. arXiv preprint arXiv:2008.01062 (2020)
47. Wang, T., Dong, H., Lesser, V., Zhang, C.: Roma: Multi-agent reinforcement learning with emergent roles. arXiv preprint arXiv:2003.08039 (2020)
48. Wang, T., Gupta, T., Mahajan, A., Peng, B., Whiteson, S., Zhang, C.: Rode: learning roles to decompose multi-agent tasks. arXiv preprint arXiv:2010.01523 (2020)
49. de Witt, C.S., et al.: Is independent learning all you need in the StarCraft multi-agent challenge? arXiv preprint arXiv:2011.09533 (2020)
50. Yang, S., Yang, B., Kang, Z., Deng, L.: IHG-MA: Inductive heterogeneous graph multi-agent reinforcement learning for multi-intersection traffic signal control. Neural Netw. **139**, 265–277 (2021). https://doi.org/10.1016/j.neunet.2021.03.015
51. Yu, C., Velu, A., Vinitsky, E., Wang, Y., Bayen, A., Wu, Y.: The surprising effectiveness of ppo in cooperative, multi-agent games. arXiv preprint arXiv:2103.01955 (2021)
52. Zheng, Z., Tan, Y.: Group explosion strategy for searching multiple targets using swarm robotic. In: 2013 IEEE Congress on Evolutionary Computation, pp. 821–828. IEEE (2013)
53. Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., Sun, M.: Graph neural networks: a review of methods and applications. arXiv:abs/1812.08434 (2020)
54. Zhou, Y., Tan, Y.: GPU-based parallel multi-objective particle swarm optimization. Int. J. Artif. Intell. **7**(A11), 125–141 (2011)