Multi-Agent Advisor Q-Learning (Extended Abstract)[*]

Sriram Ganapathi Subramanian[1,3], **Matthew E. Taylor**[2,4], **Kate Larson**[3] and **Mark Crowley**[3]
[1]Vector Institute
[2]University of Alberta
[3]University of Waterloo
[4]Alberta Machine Intelligence Institute (Amii)
sriram.subramanian@vectorinstitute.ai, matthew.e.taylor@ualberta.ca, kate.larson@uwaterloo.ca, mcrowley@uwaterloo.ca

Abstract

In the last decade, there have been significant advances in multi-agent reinforcement learning (MARL) but there are still numerous challenges, such as high sample complexity and slow convergence to stable policies, that need to be overcome before wide-spread deployment is possible. However, many real-world environments already, in practice, deploy sub-optimal or heuristic approaches for generating policies. An interesting question that arises is how to best use such approaches as *advisors* to help improve reinforcement learning in multi-agent domains. We provide a principled framework for incorporating action recommendations from online sub-optimal advisors in multi-agent settings. We describe the problem of *ADvising Multiple Intelligent Reinforcement Agents* (ADMIRAL) in nonrestrictive *general-sum stochastic game* environments and present two novel Q-learning-based algorithms: **ADMIRAL - Decision Making (ADMIRAL-DM)** and **ADMIRAL - Advisor Evaluation (ADMIRAL-AE)**, which allow us to improve learning by appropriately incorporating advice from an advisor (ADMIRAL-DM), and evaluate the effectiveness of an advisor (ADMIRAL-AE). We analyze the algorithms theoretically and provide fixed point guarantees regarding their learning in general-sum stochastic games. Furthermore, extensive experiments illustrate that these algorithms: can be used in a variety of environments, have performances that compare favourably to other related baselines, can scale to large state-action spaces, and are robust to poor advice from advisors.

1 Introduction

Reinforcement learning (RL) research is growing and expanding rapidly; however, this approach still finds only limited applications in practical real-world settings [Dulac-Arnold *et al.*, 2021]. One major reason for this is that RL algorithms typically have high sample complexity and can learn effective policies only after experiencing millions of data samples in simulation [Kakade, 2003]. Multi-agent reinforcement learning (MARL) extends RL to domains where more than one agent learns simultaneously in the environment [Shoham and Leyton-Brown, 2008]. Moving from single-agent to multi-agent settings introduces new challenges, including nonstationary environments and the curse of dimensionality [Hernandez-Leal *et al.*, 2019], while concerns from single-agent RL such as exploration-exploitation trade-offs and sample efficiency remain [Yogeswaran and Ponnambalam, 2012]. In MARL environments, it has been reported that learning complex tasks from scratch is even impractical due to its poor sample complexity [Silva and Costa, 2019]. In this regard, it becomes necessary for agents to obtain guidance from an external source to have any possibility of scaling up to real-world domains. Furthermore, during the early stages of learning, agents' policies may be quite random and dangerous, which makes it almost impossible to use them in real-world environments. Thus, it is hard to improve upon these policies by only using direct interactions with the environment. In this paper, we tackle the problem of improving sample efficiency in MARL through the use of other sources of knowledge, particularly during the early stages of training.

In single agent RL, the use of external knowledge sources such as *advisors* to drive exploration has been successful in a variety of domains. The advisors provide actions to the agent at different states to bootstrap learning by targeted exploration [Nair *et al.*, 2018]. However, the biases of sub-optimal advisors pose a challenge to successful learning [Gao *et al.*, 2018]. Further, many of these approaches do not directly extend to MARL due to the additional complications present in the multi-agent environments. Although learning from external sources of knowledge has been explored in MARL, many previous works assume the presence of fully optimal experts [Natarajan *et al.*, 2010; Hadfield-Menell *et al.*, 2016; Yu *et al.*, 2019]. Generally, they entail additional assumptions such as having simplified environments with only two agents [Lin *et al.*, 2019] and consider restrictive environments such as competitive zero-sum [Wang and Klabjan, 2018] or fully cooperative settings where all agents share a common goal [Natarajan *et al.*, 2010; Le *et al.*, 2017; Peng *et al.*, 2020]. Additionally, some approaches, such as [Lin *et al.*, 2019] restrict themselves to simple multi-agent

[*] This is an extended abstract of our JAIR article [Subramanian *et al.*, 2022]

environments with discrete state and action spaces. The use of sub-optimal advisors in multi-agent general-sum settings with an arbitrary number of agents has been less explored, and to the best of our knowledge, there has been no comprehensive analysis of this approach, especially from a theoretical perspective.

We introduce a principled framework for studying the problem of **ADvising Multiple Intelligent Reinforcement Agents** (ADMIRAL). We propose two Q-learning-based algorithms [Watkins and Dayan, 1992]. The first algorithm, **ADvising Multiple Intelligent Reinforcement Agents - Decision Making** (ADMIRAL-DM), learns to act in the environment using advisor-guidance, while the second, **ADvising Multiple Intelligent Reinforcement Agents - Advisor Evaluation** (ADMIRAL-AE), provides a principled method to evaluate the usefulness of the advisor in the current MARL context. To the best of our knowledge, we are the first to propose a method to evaluate a knowledge source before using it for learning in MARL. We empirically study the performance of our algorithms in suitable test-beds, along with a comparison to related baselines. Theoretically, we establish conditions under which we can provide fixed point guarantees regarding the learning of our ADMIRAL algorithms in general-sum stochastic game environments [Shapley, 1953].

2 Advisor Q-Learning

First, we introduce the problem of ADvising Multiple Intelligent Reinforcement Agents (ADMIRAL). We have a set of agents that can either take an action using their own policy or consult an advisor that provides action recommendations, given the current state, at each time step. Each agent has access to at most one advisor. An advisor can be any external source of knowledge, such as a rule-based agent, a pre-trained policy, or any other system that continues to learn during gameplay. The advisor is assumed to be available online with the possibility of providing instantaneous action recommendations to an agent. Furthermore, we consider a centralized training setting where agents can observe the state, the local actions, and the rewards of all other agents. We also assume that the advisor and agent communication is free, while the agents cannot communicate among themselves. There is no communication amongst the agents themselves, since establishing reliable communication protocols amongst every single agent may be prohibitively expensive in large multi-agent environments. Having specified this setting, we subsequently show that many of these restrictions can be relaxed.

We study two challenges that arise when learning from advisors in MARL and provide algorithms for each problem. The first challenge is learning a policy with the help of an advisor. We introduce an algorithm for this challenge, which we call ADvising Multiple Intelligent Reinforcement Agents - Decision Making (ADMIRAL-DM). In this setting, each agent aims to learn a suitable policy that provides the best responses to the opponent(s) and performs effectively in the given multi-agent environment. An agent has access to a (possibly sub-optimal) advisor that could be leveraged to improve the speed of learning. A schematic of this setting is provided in Figure 1. The second challenge is the evaluation of the advisor itself. Before using an advisor, it is beneficial to evaluate it to determine whether the advisor will provide effective advice. Hence, we propose a 'pre-learning' phase (i.e., a distinct phase before the beginning of training of ADMIRAL-DM) and provide an algorithm called ADvising Multiple Intelligent Reinforcement Agents - Advisor Evaluation (ADMIRAL-AE) which has the goal of getting a good understanding of the capabilities of the advisor in the current environment. We assume that a single advisor exists in the system, and this advisor could be evaluated by one or more agents. A schematic of this setting is provided in Figure 2.

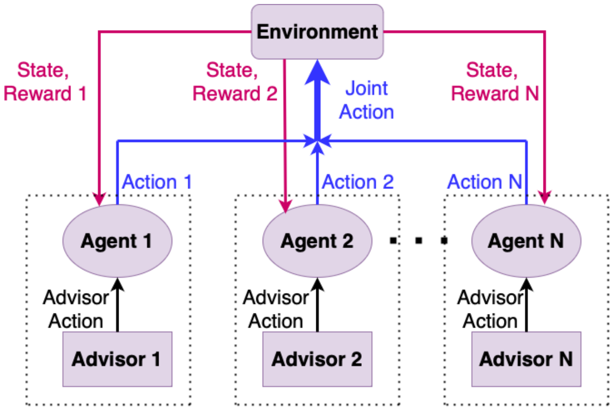


Figure 1: Architecture of the ADMIRAL-DM algorithm.

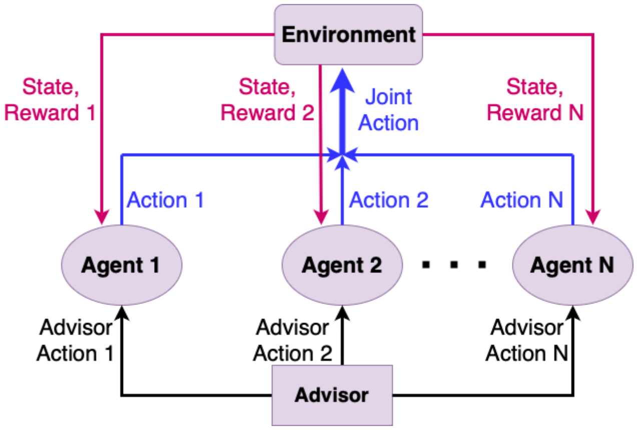


Figure 2: Architecture of the ADMIRAL-AE algorithm.

First we describe ADMIRAL-DM. As in [Hu and Wellman, 2003], ADMIRAL-DM allows all agents maintain a copy of the Q-updates of the other agents. This is possible since, during training, agents are in a centralized setting and can observe the local actions and rewards of all other agents at each time step. This helps in predicting the actions of opponents needed for providing the best responses. A learning agent (represented by j) starts with an arbitrary initialization of its Q-value $Q_0^j(s, \boldsymbol{a})$. Here $\boldsymbol{a} = (a^1, \ldots, a^n)$ represents the joint action of all agents. Recall, in this setting, each agent has access to an online advisor that it could query during learning. Whenever the agent needs to choose an action, it does so based on its current Q-value, the advisor's recommendation, or simply a random action, as the case may be. The dependence on the advisor's recommendation and random exploration is captured by two hyperparameters, ϵ'_t and ϵ_t, respectively. This action is subsequently executed and the actions and rewards of the other agents are observed, including the next state s'. During training, at each time step, the agent picks the possible next actions of other agents using its copy of the other agents' Q values. Then, the agent j picks its next action based on ADMIRAL-DM algorithm's policy which chooses a random action and an advisor action with diminishing probabilities, and a greedy action with increasing probabilities, such that it becomes greedy in the limit with infinite exploration (GLIE). Thus, the agent is guaranteed to train without any further advisor influence after some finite time t in the training process. Accordingly, the dependence on the advisor's recommendation is decayed linearly. In this process, the dependence of an agent is more on the advisor during the earlier stages of learning, when its own policy is quite bad. This dependence gradually reduces as its own policy improves. The Q-values are updated as,

$$\begin{aligned} &Q^j_{t+1}(s, \boldsymbol{a}) \\ &= (1-\alpha_t)Q^j_t(s, \boldsymbol{a}) + \alpha_t[r^j_t + \beta Q^j_t(s', \boldsymbol{a'})] \end{aligned} \tag{1}$$

where $\boldsymbol{a} = (a^1, \dots, a^n)$ denotes the actions for all agents at state s and $\boldsymbol{a'} = (a^{1'}, \dots, a^{n'})$ denotes the actions for all the agents at state s'. β denotes the discount factor, and $\alpha_t \in (0, 1)$ is the learning rate. The algorithm's steps are repeated continuously until either the Q-values fully converge or come within a small threshold of convergence, as is commonly done in practice [Sutton and Barto, 1998].

We extend the ADMIRAL-DM it to an actor-critic method — **ADvising Multiple Intelligent Reinforcement Agents - Decision Making (Actor-Critic)** abbreviated as ADMIRAL-DM(AC). This algorithm uses the Q-function as the critic and the policy derived from Q as the actor. The algorithm follows a *Centralized Training and Decentralized Execution* (CTDE) scheme [Lowe *et al.*, 2017], where the critic uses the information associated with other agents during the training time and the actors can act independently without access to other agent information during execution. This allows our methods to be applicable in environments where global information (i.e., information associated with other agents) is available during training but not during execution, such as autonomous driving [Zhou *et al.*, 2020]. The CTDE scheme extends our algorithm to partially observable environments, where the actor can just use the local observations of the agent for action selection (during both training and execution), while the critic can use the joint observation of all agents during training.

Thus, ADMIRAL-DM uses an advisor if one exists. Furthermore, our second algorithm ADMIRAL-AE (which is related to our second challenge, i.e., the evaluation of the advisor) evaluates a potential advisor and helps guide the configuration of ADMIRAL-DM by setting the initial value of ϵ'. The objective is to make an agent following ADMIRAL-DM listen more to good advisors and listen less (or not at all) to bad advisors. In ADMIRAL-AE, at each time t, the agent j observes the current state s, and takes a local action a^j and observes the action of all agents (including itself), the reward it obtains and the new state s'. The agent then obtains a recommendation from the advisor for the next state s'. Subsequently, each agent j updates its Q-value as follows:

$$\begin{aligned} &Q^j_{t+1}(s, \boldsymbol{a}) \\ &= (1-\alpha_t)Q^j_t(s, \boldsymbol{a}) + \alpha_t[r^j_t + \beta AdvisorQ^j_t(s')]. \end{aligned} \tag{2}$$

The term $AdvisorQ^j_t(s')$, is the total value (payoff) that the agent j will obtain at the state s' when all agents (including itself) play the advisor solution. This is calculated as $AdvisorQ^j_t(s') = \sigma^1_t(s') \cdots \sigma^n_t(s') \cdot Q^j_t(s')$, where $(\sigma^1_t(s'), \dots, \sigma^n_t(s'))$ denotes the advisor recommendations at state s' and time t. This can be seen as a solution to the stage game $(Q^1_t(s'), \dots, Q^n_t(s'))$. Subsequently, these Q values are used to choose the initial value of ϵ' in ADMIRAL-DM (more details in [Subramanian *et al.*, 2022]). All our algorithms can be extended to large state-action environments using function approximations [Mnih *et al.*, 2015].

3 Theoretical Results

We have two important theoretical guarantees for our algorithms. First, we show that the Q-updates following ADMIRAL-AE converge to an ϵ-equilibrium in the stochastic game. Second, we prove that the Q-updates following ADMIRAL-DM converges to the Nash Q-value, thus finding the Nash equilibrium of the stochastic game.

The primary convergence result for algorithms based on Q learning in a general sum stochastic game was provided by [Hu and Wellman, 2003]. However, this result relies on a very restrictive assumption that states that every stage game of the stochastic game contains a Nash equilibrium that is either a global optimum or a saddle point. Additionally, an agent must use the payoff at this equilibrium to update its Q value in every stage game of the stochastic game. As shown by [Bowling, 2000], this assumption implies that every stage game should use the same kind of equilibrium, it cannot oscillate between being a global optimum or saddle point between stage games. There is almost no game that satisfies this condition in practice [Hu and Wellman, 2003]. The convergence results in our setting can be provided under a set of assumptions weaker than that used by [Hu and Wellman, 2003].

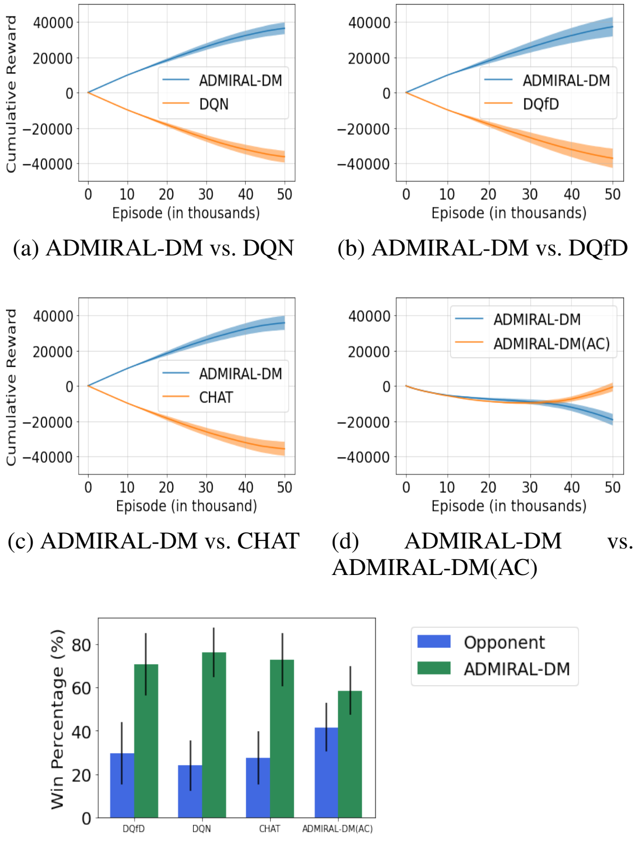
Theorem 1. ***(Informal)*** *Under a set of assumptions, the Q-functions updated by ADMIRAL-AE converge to a bounded distance from the Nash Q-function $Q_* = (Q^1_*, \dots, Q^n_*)$, in the time limit ($t \to \infty$).*

Theorem 2. ***(Informal)*** *Under a set of assumptions, the Q-functions updated by ADMIRAL-DM converge to the Nash Q-function, in the limit ($t \to \infty$).*

4 Experimental Results

We experimentally validate our algorithms, showing their effectiveness in a variety of situations using different testbeds. We also demonstrate superior performance to common baselines. The results from one setting is presented here, while more elaborate experimental results and associated discussions are in [Subramanian *et al.*, 2022].

The setting we consider is Domain OneVsOne of Pommerman introduced in [Resnick *et al.*, 2018]. Our baselines are DQfD [Hester *et al.*, 2018], CHAT [Wang and Taylor, 2017], and DQN [Mnih *et al.*, 2015]. We perform 50,000 episodes of training, where the algorithms train against specific opponents. Each episode is a full Pommerman game (lasting a maximum of 800 steps). All the algorithms relying on demonstrations (DQfD, CHAT, ADMIRAL-DM, and ADMIRAL-DM(AC)) use rule-based agents as advisors. These rule-based agents have been provided by [Resnick *et al.*, 2018], and are known to have a high standard of performance [Perez-Liebana *et al.*, 2019]. The probability of using the advisor action (ϵ'_t in ADMIRAL-DM) starts from a high value (determined using the ADMIRAL-AE method, more details in [Subramanian *et al.*, 2022]) and linearly decays to be close to zero at the end of training for both ADMIRAL-DM and ADMIRAL-DM(AC) (using the PPR technique). To provide data for offline pretraining in the case of DQfD, two instances of the rule-based agents are used to play many Pommerman games that generate the required data. The DQfD is pretrained with all of this data, before entering the training phase of our experiments.



(a) ADMIRAL-DM vs. DQN

(b) ADMIRAL-DM vs. DQfD

(c) ADMIRAL-DM vs. CHAT

(d) ADMIRAL-DM vs. ADMIRAL-DM(AC)

(e) Faceoff against ADMIRAL-DM

Figure 3: Pommerman competition against ADMIRAL-DM.

After the training phase, the trained algorithms enter a face-off competition of 10,000 games where there is no more training, no further exploration and additionally ADMIRAL-DM and ADMIRAL-DM(AC) play without any advisor influence. ADMIRAL-DM(AC) is a CTDE technique, which only performs decentralized execution in face-off using the trained actor-network. We plot the cumulative rewards in the training phase (Figure 3 (a), (b), (c), (d)), from which it can be seen that ADMIRAL-DM's performance is better than the baselines (DQN, DQfD, and CHAT). The face-off plots in Figure 3(e) show that ADMIRAL-DM wins more games on average against all the other baselines, showing its dominance. DQfD relies on pretraining, which is harder in MARL, as the nature of opponents that an agent will face during competition is impossible to determine upfront. The algorithms that use online advisors to give real-time feedback (capturing the changing nature of the opponent) tend to do better. DQfD has also been previously reported to have over-fitting issues [Gao *et al.*, 2018], which is likely to hurt its performance more in multi-agent environments compared to single-agent environments. In multi-agent environments, it is more important to be able to generalize to unseen dynamic opponent behaviour, which is different from that seen in pre-collected demonstration data. CHAT maintains a confidence measure on the advisor, which depends on the advisor's consistency in action recommendations at different states. In MARL, this measure is not completely reliable, since even good advisors may need to formulate stochastic action recommendations as responses to the opponent. DQN, on the other hand, learns directly from interaction experiences and cannot learn from advisor inputs. This is a disadvantage in environments where external sources of knowledge, such as advisors, are available to be leveraged. Furthermore, since our baselines are independent algorithms (that consider opponents to be part of the state), they lose out to ADMIRAL-DM, which explicitly tracks opponent action. ADMIRAL-DM loses to ADMIRAL-DM(AC) during training (Figure 3 (d)). Though ADMIRAL-DM(AC) shows slower learning overall (as it is training both actor and critic), it ultimately learns a higher performing policy. One important reason is that the actor-critic method trains a stochastic policy that can explore naturally, whereas the Q-learning method needs a hyperparameter to conduct forced exploration (ϵ-greedy). Another reason could be that ADMIRAL-DM(AC) learns from each recent experience, while ADMIRAL-DM has delayed learning using the replay buffer. However, in the face-off, ADMIRAL-DM has an edge over ADMIRAL-DM(AC) (Figure 3(e)), probably due to being centralized. Further, we perform a Fischer's exact test to check statistical significance ($p < 0.03$).

5 Conclusion

In this paper, we provide a principled framework for MARL algorithms to accelerate training using external advisors. Using Q-learning-based methods, we proposed two MARL algorithms for this problem. We conducted theoretical analyses of these algorithms, establishing conditions under which fixed point guarantees can be provided regarding their learning in general-sum stochastic games. Empirically, we showed that our algorithms can be scaled to domains with large state-action spaces using traditional function approximators like neural networks. Our empirical results further established the superiority of our algorithms compared to standard baselines.

Acknowledgements

Kate Larson is an affiliate of the Vector Institute, Toronto. The authors thank Seyed Majid Zahedi at the University of Waterloo for his comments on the paper draft. Resources used in preparing this research at the University of Waterloo were provided by the province of Ontario and the government of Canada through CIFAR, NRC, NSERC and companies sponsoring the Vector Institute. 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

[Bowling, 2000] Michael H. Bowling. Convergence Problems of General-Sum Multiagent Reinforcement Learning. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*, pages 89–94. Morgan Kaufmann, 2000.

[Dulac-Arnold *et al.*, 2021] Gabriel Dulac-Arnold, Nir Levine, Daniel J Mankowitz, Jerry Li, Cosmin Paduraru, Sven Gowal, and Todd Hester. Challenges of real-world reinforcement learning: Definitions, Benchmarks and Analysis. *Machine Learning*, 1:1–50, 2021.

[Gao *et al.*, 2018] Yang Gao, Huazhe Xu, Ji Lin, Fisher Yu, Sergey Levine, and Trevor Darrell. Reinforcement Learning from Imperfect Demonstrations. In *6th International Conference on Learning Representations, (ICLR 2018), Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*. OpenReview.net, 2018.

[Hadfield-Menell *et al.*, 2016] Dylan Hadfield-Menell, Stuart J. Russell, Pieter Abbeel, and Anca D. Dragan. Cooperative Inverse Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS 2016), Barcelona, Spain, December 5-10, 2016*, pages 3909–3917, 2016.

[Hernandez-Leal *et al.*, 2019] Pablo Hernandez-Leal, Bilal Kartal, and Matthew E Taylor. A survey and critique of multiagent deep reinforcement learning. *Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS)*, 33(6):750–797, 2019.

[Hester *et al.*, 2018] Todd Hester, Matej Vecerík, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, Gabriel Dulac-Arnold, John P. Agapiou, Joel Z. Leibo, and Audrunas Gruslys. Deep Q-learning From Demonstrations. In *Thirty-Second Conference of Association for the Advancement of Artificial Intelligence (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 3223–3230. AAAI Press, 2018.

[Hu and Wellman, 2003] Junling Hu and Michael P Wellman. Nash Q-learning for general-sum stochastic games. *Journal of Machine Learning Research (JMLR)*, 4(Nov):1039–1069, 2003.

[Kakade, 2003] Sham Machandranath Kakade. *On the sample complexity of reinforcement learning*. PhD thesis, UCL (University College London), 2003.

[Le *et al.*, 2017] Hoang Minh Le, Yisong Yue, Peter Carr, and Patrick Lucey. Coordinated Multi-Agent Imitation Learning. In *Proceedings of the 34th International Conference on Machine Learning, (ICML 2017), Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1995–2003. PMLR, 2017.

[Lin *et al.*, 2019] Xiaomin Lin, Stephen C Adams, and Peter A Beling. Multi-agent inverse reinforcement learning for certain general-sum stochastic games. *Journal of Artificial Intelligence Research (JAIR)*, 66:473–502, 2019.

[Lowe *et al.*, 2017] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In *Advances in Neural Information Processing Systems (NeurIPS 2017), Long Beach, CA, USA, December 4-9, 2017*, pages 6379–6390, 2017.

[Mnih *et al.*, 2015] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[Nair *et al.*, 2018] Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Overcoming Exploration in Reinforcement Learning with Demonstrations. In *2018 IEEE International Conference on Robotics and Automation, (ICRA 2018) Brisbane, Australia, May 21-25, 2018*, pages 6292–6299. IEEE, 2018.

[Natarajan *et al.*, 2010] Sriraam Natarajan, Gautam Kunapuli, Kshitij Judah, Prasad Tadepalli, Kristian Kersting, and Jude W. Shavlik. Multi-Agent Inverse Reinforcement Learning. In *The Ninth International Conference on Machine Learning and Applications, (ICMLA 2010), Washington, DC, USA, 12-14 December 2010*, pages 395–400. IEEE Computer Society, 2010.

[Peng *et al.*, 2020] Peixi Peng, Junliang Xing, and Lili Cao. Hybrid Learning for Multi-agent Cooperation with Sub-optimal Demonstrations. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, (IJCAI-2020)*, pages 3037–3043. International Joint Conferences on Artificial Intelligence Organization, 7 2020. Main track.

[Perez-Liebana *et al.*, 2019] Diego Perez-Liebana, Raluca D Gaina, Olve Drageset, Ercüment Ilhan, Martin Balla, and Simon M Lucas. Analysis of statistical forward planning methods in pommerman. In *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 15, pages 66–72, 2019.

[Resnick *et al.*, 2018] Cinjon Resnick, Wes Eldridge, David Ha, Denny Britz, Jakob Foerster, Julian Togelius, Kyunghyun Cho, and Joan Bruna. Pommerman: A Multi-

Agent Playground. In *arXiv preprint arXiv:1809.07124*, 2018.

[Shapley, 1953] Lloyd S Shapley. Stochastic games. *Proceedings of the national academy of sciences*, 39(10):1095–1100, 1953.

[Shoham and Leyton-Brown, 2008] Yoav Shoham and Kevin Leyton-Brown. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.

[Silva and Costa, 2019] Felipe Leno Da Silva and Anna Helena Reali Costa. A Survey on Transfer Learning for Multiagent Reinforcement Learning Systems. *Journal of Artificial Intelligence Research (JAIR)*, 64:645–703, 2019.

[Subramanian *et al.*, 2022] Sriram Ganapathi Subramanian, Matthew E. Taylor, Kate Larson, and Mark Crowley. Multi-Agent Advisor Q-Learning. *Journal of Artificial Intelligence Research*, 74:1–74, 2022.

[Sutton and Barto, 1998] Richard S Sutton and Andrew G Barto. *Introduction to Reinforcement Learning*, volume 135. MIT Press Cambridge, 1998.

[Wang and Klabjan, 2018] Xingyu Wang and Diego Klabjan. Competitive Multi-agent Inverse Reinforcement Learning with Sub-optimal Demonstrations. In *Proceedings of the Thirty-fifth International Conference on Machine Learning (ICML 2018), Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 5130–5138. PMLR, 2018.

[Wang and Taylor, 2017] Zhaodong Wang and Matthew E. Taylor. Improving Reinforcement Learning with Confidence-Based Demonstrations. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI 2017), Melbourne, Australia, August 19-25, 2017*, pages 3027–3033. ijcai.org, 2017.

[Watkins and Dayan, 1992] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8(3-4):279–292, 1992.

[Yogeswaran and Ponnambalam, 2012] Mohan Yogeswaran and SG Ponnambalam. Reinforcement learning: Exploration–exploitation dilemma in multi-agent foraging task. *Opsearch*, 49(3):223–236, 2012.

[Yu *et al.*, 2019] Lantao Yu, Jiaming Song, and Stefano Ermon. Multi-Agent Adversarial Inverse Reinforcement Learning. In *Proceedings of the Thirty-Sixth International Conference on Machine Learning, (ICML 2019), Long Beach, California, USA, 9-15 June 2019*, volume 97 of *Proceedings of Machine Learning Research*, pages 7194–7201. PMLR, 2019.

[Zhou *et al.*, 2020] Ming Zhou, Jun Luo, Julian Villela, Yaodong Yang, David Rusu, Jiayu Miao, Weinan Zhang, Montgomery Alban, Iman Fadakar, Zheng Chen, Aurora Chongxi Huang, Ying Wen, Kimia Hassanzadeh, Daniel Graves, Dong Chen, Zhengbang Zhu, Nhat M. Nguyen, Mohamed Elsayed, Kun Shao, Sanjeevan Ahilan, Baokuan Zhang, Jiannan Wu, Zhengang Fu, Kasra Rezaee, Peyman Yadmellat, Mohsen Rohani, Nicolas Perez Nieves, Yihan Ni, Seyedershad Banijamali, Alexander Imani Cowen-Rivers, Zheng Tian, Daniel Palenicek, Haitham Bou-Ammar, Hongbo Zhang, Wulong Liu, Jianye Hao, and Jun Wang. SMARTS: Scalable Multi-agent Reinforcement Learning Training School for Autonomous Driving. *CoRR*, abs/2010.09776, 2020.