Scalable and Sample-Efficient Multi-Agent Imitation Learning

Wonseok Jeon[1,2], **Paul Barde**[1,2], **Joelle Pineau**[1,2,3], **Derek Nowrouzezahrai**[1,2],
[1]Mila, Quebec AI Institute
[2]McGill University
[3]Facebook AI Research
jeonwons@mila.quebec

Abstract

Multi-agent generative adversarial imitation learning (MAGAIL) is a recent approach that extends single-agent GAIL to problems in multi-agent imitation learning. While MAGAIL shows promising results on cooperative and competitive tasks, it requires agent-environment interactions during training, which may reduce sample efficiency in practice. Moreover, MAGAIL was validated empirically on only a handful of agents, and its scalability to larger numbers of agents remains a question. We propose a multi-agent imitation learning algorithm that addresses these issues. Specifically, we apply multi-agent actor-critic (MAAC) and multi-agent attention-actor-critic (MAA2C) – off-policy multi-agent reinforcement learning (MARL) approaches – in the MARL imitation learning inner loop, as opposed to MACK – the on-policy MARL method used in MAGAIL. We then model centralized and decentralized discriminators to evaluate whether a given behavior results from agent or expert actions, defining reward functions for the MARL inner loop. We demonstrate that our method scales more effectively, and more sample-efficient, than MAGAIL. We also demonstrate that imitation learning with decentralized discriminators is robust, performing surprisingly well for a large number of agents compared to its centralized counterpart.

Introduction

Imitation learning is a sequential decision-making problem in which an agent attempts to mimic a predefined expert's behavior. Generative adversarial imitation learning (GAIL) (Ho and Ermon 2016) is a class of imitation learning algorithms that applies an adversarial training method inspired by Generative Adversarial Networks (GANs): GAIL trains a discriminator to classify the behaviors of agents and experts, using it to define a reward signal to allow an agent to recognize and reduce behavioral discrepancies. GAIL successfully mimics expert behavior on several control benchmarks where prior work – such as behavioral cloning (Ross, Gordon, and Bagnell 2011) or inverse reinforcement learning (Ziebart et al. 2008) – had otherwise failed.

GAIL, however, requires a large number of agent-environment interactions in order to converge, precluding its use in applications with costly environment interactions, e.g., robotics and autonomous vehicles. Recent works work to address this sample efficiency issue: generative moment-matching imitation learning (GMMIL) (Kim and Park 2018) applies a generative moment matching networks (GMMNs) (Li, Swersky, and Zemel 2015) instead of a traditional GAN, Bayesian GAIL (BGAIL) (Jeon, Seo, and Kim 2018) leverages a Bayesian discriminator to improve reward signals, and other methods exploit off-policy reinforcement learning (Sasaki, Yohira, and Kawaguchi 2019; Kostrikov et al. 2019).

Meanwhile, MAGAIL (Song et al. 2018) was recently proposed for multi-agent imitation learning problems in which each agent is allowed to use demonstrations of a corresponding expert for imitation. Similar to GAIL, MAGAIL trains discriminators to create reward signals for each agent and is shown to perform better than agent-wise behavioral cloning, agent-wise GAIL and GAIL applied to multi-agents' joint trajectories. Although MAGAIL shows promising performance in multi-agent imitation, it uses the multi-agent extension of ACKTR (Wu et al. 2017) (MACK), an on-policy reinforcement learning (RL) algorithm and possibly sample-inefficient compared to off-policy RL algorithms that reuse past experiences. Also, MACK uses centralized critics for each agent, which is widely adopted in the *centralized-training with decentralized execution* framework (Lowe et al. 2017; Foerster et al. 2018) but is not scalable with respect to the number of agents.

To address the above issues of MAGAIL, we propose a multi-agent imitation learning method that is highly sample-efficient and scalable compared to MAGAIL. Rather than using MACK, we use MAAC and MAA2C (Iqbal and Sha 2019), which are off-policy MARL algorithms, instead of using MACK in MAGAIL. We also consider centralized and decentralized discriminators both of which were proposed by MAGAIL and are used to define reward signals for the MARL inner loop of multi-agent imitation. As a result, we empirically verify that both imitation learning algorithms with MAAC an MAA2C perform better than MAGAIL regardless of the number of agents. Especially, we can find out that MAA2C performs the best among those algorithms and is highly scalable for a large number of agents. Finally, it

turns out that imitation learning with decentralized discriminators scales better than its centralized counterpart over a variety of our benchmarks.

Background

Markov Games and Notations

In a Markov Game (MG), it is assumed that there are multiple agents observing a shared environment states and taking actions. After taking an action each agent gets a reward and the environment transitions to a new state. Mathematically, a Markov Game is defined by $\mathcal{G} = \langle \mathcal{S}, \{\mathcal{A}_i\}_{i=1}^N, \{r_i\}_{i=1}^N, P_T, \nu, \gamma \rangle$, where N is the number of agents, $\mathcal{S}$ is the state space, $\mathcal{A}_i$ is the action space for agent i, $r_i(s, a_1, ..., a_N)$ is a reward function for agent i, $P_T(s'|s, a_1, ..., a_N)$ is a state transition distribution, $\nu(s)$ is an initial state distribution, γ is a discount factor (Littman 1994). When $N = 1$, a Markov Game is reduced to a Markov Decision Process (MDP) (Bellman 1957) that is used to model single-agent sequential decision making problems. In this work, we consider a partially observable Markov Game, where each agent can only measure the partial observation $o_i \in \mathcal{O}_i$ from the environment's state $s \in \mathcal{S}$. For each agent i, a policy $\pi_i(a_i|o_i)$ is the probability that the i-th agent chooses an action a_i after observing o_i. For succinct notations, we use bars to indicate joint quantities over the agents. For example, $\bar{\mathcal{A}} = \mathcal{A}_1 \times \cdots \mathcal{A}_N$ is a joint action space, $\bar{a} = (a_1, ..., a_N)$ is a joint action, $\bar{\pi} = (\pi_1, ..., \pi_N)$ is a joint policy, $\bar{r}(s, \bar{a}) = (r_1(s, \bar{a}), r_2(s, \bar{a}), ..., r_N(s, \bar{a}))$ is a joint reward.

Given a state s, the value function of the i-th agent with respect to $\bar{\pi}$ is defined by

$$V_i^{\bar{\pi}}(s) = \mathbb{E}^{\bar{\pi}} \left[\sum_{t=0}^{\infty} \gamma^t r_i(\boldsymbol{s}_t, \bar{\boldsymbol{a}}_t) | \boldsymbol{s}_0 = s \right],$$

where the bold letters indicate random variables and the superscript $\bar{\pi}$ on the expectation implies that states and joint actions are sampled from ν, P_T and $\bar{\pi}$. Similarly, one can define the state-action value function of the i-th agent with respect to $\bar{\pi}$ as

$$Q_i^{\bar{\pi}}(s, \bar{a}) = \mathbb{E}^{\bar{\pi}} \left[\sum_{t=0}^{\infty} \gamma^t r_i(\boldsymbol{s}_t, \bar{\boldsymbol{a}}_t) | \boldsymbol{s}_0 = s, \bar{\boldsymbol{a}}_0 = \bar{a} \right].$$

Additionally, the γ-discounted state occupancy measure $\rho^{\bar{\pi}}$ of the joint policy $\bar{\pi}$ is defined as $\rho^{\bar{\pi}}(s) = \mathbb{E}^{\bar{\pi}}[\sum_{t=0}^{\infty} \gamma^t \mathbb{I}\{\boldsymbol{s}_t = s\}]$, where $\mathbb{I}\{\cdot\}$ is an indicator function. We also define

$$\rho^{\bar{\pi}}(s, \bar{a}) = \rho^{\bar{\pi}}(s) \prod_{i=1}^N \pi_i(a_i|s)$$

as a state-action occupancy measure with a little abused notation. The goal of each agent in the MARL setting is to optimize its own policy so that it maximizes its individual expected cumulative reward.

Multi-Agent Generative Adversarial Imitation Learning

In MAGAIL (Song et al. 2018), it is assumed that there are N experts $\bar{\pi}^E = (\pi_1^E, ..., \pi_N^E)$ that N agents try to imitate respectively but can only use the limited amount of data from a limited number of experts' demonstrations. Similar to GAIL, MAGAIL enables agents to learn experts' policies by casting the imitation learning problem into the following joint state-action occupancy matching problem:

$$\underset{\pi_1, ..., \pi_N}{\operatorname{argmin}} \sum_{i=1}^N \left\{ \psi_i^*(\rho_{\bar{\pi}^E_{-i}(\pi_i)} - \rho_{\bar{\pi}^E}) - \beta H_i(\pi_i) \right\}. \quad (1)$$

Here, $\bar{\pi}_{-i}(\hat{\pi}) = (..., \pi_{i-1}, \hat{\pi}, \pi_{i+1}, ...)$ is a joint policy with a uni-literal change of policy from π_i to $\hat{\pi}$, $(\psi_1, ..., \psi_N)$ is a set of convex reward function regularizers, f^* is a convex conjugate function of f, $H_i(\pi_i) = \mathbb{E}_{\boldsymbol{s}, \bar{\boldsymbol{a}} \sim \rho_{\bar{\pi}^E_{-i}(\pi_i)}}[-\log \pi_i(\boldsymbol{a}_i|\boldsymbol{s})]$ is a causal entropy of the i-th agent's policy. Roughly, the objective (1) matches the experts' occupancy measure $\rho_{\bar{\pi}^E}(s, a)$ with another occupancy measure $\rho_{\bar{\pi}^E_{-i}(\pi_i)}(s, a)$ which assumes that the i-th agent's policy is used instead of the i-th expert's policy. By choosing $\psi_1, ..., \psi_N$ similar to GAIL and letting β equal to 0, it was shown in (Song et al. 2018) that the objective becomes solving the following mini-max problem:

$$\min_{\bar{\pi}} \max_{D_1, ..., D_N} \mathbb{E}_{\boldsymbol{s}, \bar{\boldsymbol{a}} \sim \rho_{\bar{\pi}}} \left[\sum_{i=1}^N \log D_i(\boldsymbol{s}, \boldsymbol{a}_i) \right] + \mathbb{E}_{\boldsymbol{s}, \bar{\boldsymbol{a}} \sim \rho_{\bar{\pi}^E}} \left[\sum_{i=1}^N \log(1 - D_i(\boldsymbol{s}, \boldsymbol{a}_i)) \right],$$

where $D_1, \ldots, D_N$ are multiple discriminators that classify whether state-action pairs comes from agents or experts. MAGAIL solves the above problem by first training discriminators and applying MARL for $\bar{\pi}$ with reward functions defined by the discriminators.

Off-Policy Single-Agent Generative Adversarial Imitation Learning

There have been some recent works that use off-policy RL algorithms to enhance the sample efficiency of imitation. In (Kostrikov et al. 2019), TD3 (Fujimoto, Hoof, and Meger 2018), a state-of-the-art off-policy reinforcement learning algorithm, was used with a discriminator. To stabilize their algorithm, it was proposed to learn terminal-state values, whereas conventional off-policy reinforcement learning algorithms implicitly consider zero terminal-state values and do not learn them. As a result, learning terminal-state value was shown to be effective in some tasks since a discriminator-based reward $\log D(s, a)$ $(-\log(1 - D(s, a)))$, which is always negative (positive), makes terminal-state values non-zero and can result in biased value function estimation. In (Sasaki, Yohira, and Kawaguchi 2019), another sample-efficient imitation learning algorithm was proposed. In contrast with the prior works of (Sasaki, Yohira, and Kawaguchi 2019), (Sasaki, Yohira, and Kawaguchi 2019) didn't use discriminators by considering the Bellman consistency of the imitation learning reward signal. Then, by using off-policy actor-critic (Off-PAC) (Degris, White, and Sutton 2012), it was shown that the proposed method is much more sample-efficient than GAIL.

Scalable Multi-Agent Learning

For a large number of agents, it has been regarded as a challenging problem for MARL to achieve cooperative and coordinated multi-agent behaviors. Although existing works

using centralized critics such as MADDPG (Lowe et al. 2017) and COMA (Foerster et al. 2018) make a handful of agents coordinated, they struggle when the number of agents to manage increases. This is mainly due to the exponential growth of the critic inputs with the number of agents, which possibly increases the input variance of training as well. MAA2C (Iqbal and Sha 2019) addressed such a scalability issue by using the attention mechanism in (Vaswani et al. 2017) and a shared critic network and outperformed existing algorithms for a large number of agents. Thanks to the attention mechanism, MAA2C is trained to focus only on part of joint observations and actions, which leads to rapid and efficient training. Also, due to the use of a shared critic network, the number of critic network parameters linearly increases as the number of agents increases, whereas it exponentially increases without sharing those parameters. Mean-field MARL (MFMARL) (Yang et al. 2018) is another approach to address the scalability issue of MARL, which is based on mean-field approximation for the centralized critics. However, its application is restricted to the situation where all agents are homogeneous, whereas MAA2C can be applied to much general scenarios in which non-homogeneous agents exist.

In the meantime, there were some multi-agent imitation learning algorithms applied to large-scale environments. (Le et al. 2017) proposed multi-agent imitation learning in the index-free control setting, where experts' demonstrations are available, but agents are not allowed to know the index of its expert. The proposed method trains a model that infers and assigns the role of each agent with rollout trajectories and given experts' demonstrations and exploits that model to make highly coordinated behavior. (Sanghvi, Yonetani, and Kitani 2019) uses multi-agent imitation learning to learn social group communication among agents with a single shared policy network among agents. However, they used multi-agent behavioral cloning and focused on the environments with homogeneous agents. In contrast with those works, our algorithm can deal with non-homogeneous agents as well. In addition, it has been reported in existing literature (Kostrikov et al. 2019; Sasaki, Yohira, and Kawaguchi 2019; Song et al. 2018) that behavioral cloning performs poorly when there are only a small number of experts' demonstration due to the co-variate shift (Ross, Gordon, and Bagnell 2011). For these reasons, we narrow down our scope to MAGAIL in this work.

Our approach

We introduce our multi-agent imitation learning method. As outlined in **Algorithm 1**, our method iteratively trains discriminators using experts' demonstrations and agents' rollout trajectories – using MARL and a reward signal modeled by the discriminators – similarly to GAIL.

MARL Algorithm

We adopt MAAC and MAA2C (Iqbal and Sha 2019) as our MARL algorithms. Especially, MAA2C is chosen due to its sample-efficiency as well as its scalability, and we summarize it as follows. Assuming discrete action spaces, let

$$\bar{\vec{Q}}_{\bar{\phi}}(\bar{o}, \bar{a}) = (\vec{Q}_{\phi_1}(\bar{o}, \bar{a}), ..., \vec{Q}_{\phi_N}(\bar{o}, \bar{a}))$$

denote action values, where each element of $\bar{\vec{Q}}_{\bar{\phi}}(\bar{o}, \bar{a})$ is a (vector-valued) action value of the corresponding agents and $\bar{\phi} = (\phi_1, ..., \phi_N)$ is a set of neural network parameters. In MAA2C (Iqbal and Sha 2019), the i-th agent's action value was modeled as a neural network

$$\vec{Q}_{\phi_i}(\bar{o}, \bar{a}) = \vec{f}_i(g_i^{\text{local}}(o_i), g_i^{\text{global}}(\bar{o}, \bar{a})),$$

where g_i^{local} is a network looking at the i-th agent's local observation and action, g_i^{global} is another network that considers *other* agents' observations and actions. Finally, $\vec{f}_i$ is a network that takes into account the extracted features from both of the previous neural networks. The main idea of MAA2C was to model g_i^{global} with an attention mechanism (Vaswani et al. 2017) and to share this network among agents. Specifically, for the i-th agent's embedding e_i,

$$g_i^{\text{global}}(\bar{o}, \bar{a}) = \sum_{j \neq i} P_{i \to j}(o_i, a_i, o_j, a_j) v_j(o_j, a_j),$$

Algorithm 1 Multi-Agent Imitation Learning with Actor-Attention-Critic

Inputs:
- a buffer $\mathcal{B}_A$ for agents' rollout trajectories,
- a buffer $\mathcal{B}_E$ of experts' demonstrations,
- policy networks $\bar{\pi}_{\bar{\theta}} = (\pi_{\theta_1}, \ldots, \pi_{\theta_N})$,
- a shared value network $\bar{Q}_{\bar{\phi}} = (Q_{\phi_1}, \ldots, Q_{\phi_N})$, and
- a centralized discriminator (or individual discriminators) $\bar{D}_{\bar{\psi}} = (D_{\psi_1}, \ldots, D_{\psi_N})$

for each iteration do

  Sample rollout trajectories with policy:

$$(\bar{o}, \bar{a}, \bar{o}') \sim \bar{\pi}_{\bar{\theta}}.$$

  Add sampled rollout trajectories to $\mathcal{B}_A$.

  for each training iteration do

    Sample from the buffers:

$$(\bar{o}^A, \bar{a}^A) \sim \mathcal{B}_A, (\bar{o}^E, \bar{a}^E) \sim \mathcal{B}_E.$$

    Calculate rewards $\bar{r}^A = (r_1^A, ..., r_N^A)$:

$$r_i^A(\bar{o}^A, \bar{a}^A) = \begin{cases} \log D_{\psi_i}(\bar{o}^A, \bar{a}^A) \\ \text{(a centralized discriminator),} \\ \log D_{\psi_i}(o_i^A, a_i^A) \\ \text{(decentralized discriminators).} \end{cases}$$

    Calculate $\nabla_{\bar{\theta}}, \nabla_{\bar{\phi}}$ with (2), (3) and $(\bar{o}^A, \bar{a}^A, \bar{r}^A)$.

    Calculate $\nabla_{\bar{\psi}}$ with (4) and $(\bar{o}^A, \bar{a}^A, \bar{o}^E, \bar{a}^E)$.

    Update $\bar{\theta}, \bar{\phi}$ and $\bar{\psi}$ with gradients.

  end for

end for

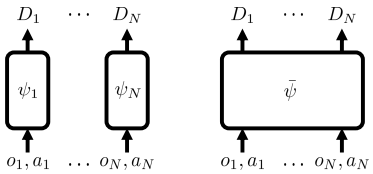

Figure 1: Decentralized (left) and centralized (right) discriminator models

where

$$P_{i\to j}(o_i, a_i, o_j, a_j) \propto \exp((W_K e_j(o_j, a_j))^T W_Q e_i(o_i, a_i)),$$
$$v_j(o_j, a_j) = \sigma(W_V e_j(o_j, a_j))$$

for element-wise non-linear activation σ and shared parameters W_K (Key), W_Q (Query) and W_V (Value).

Since the critic shares some of its parameter across agents, the objective of the critic training is to minimize the sum of temporal difference (TD) errors over all agents:

$$\underset{\bar{\phi}}{\operatorname{argmin}}\, \mathbb{E}_{\bar{o},\bar{a}\sim\rho_{\mathcal{B}}}\left[\sum_{i=1}^{N}(y_i(\bar{o},\bar{a},\bar{o}') - Q_{\phi_i}(\bar{o},\bar{a}))^2\right], \quad (2)$$
$$y_i(\bar{o},\bar{a},\bar{o}') = r_i(\bar{o},\bar{a}) + \gamma \mathbb{E}_{\bar{a}'\sim\bar{\pi}_{\bar{\theta}'}(\cdot|\bar{o}')} Q_{\phi_i'}(\bar{o}',\bar{a}'),$$

where $\bar{o}, \bar{a} \sim \rho_{\mathcal{B}}$ implies that $\bar{o}, \bar{a}$ are sampled from an experience replay buffer $\mathcal{B}$, Q_{ϕ_i} is the value for a_i in $\vec{Q}_{\phi_i}$, $\bar{\theta}'$ is a target policy parameter, and ϕ_i' is a target critic parameter. Additionally, r_i is the i-th agent's reward defined by the discriminator and will be introduced in the following section.

For each agent's policy update, we use the policy gradient

$$\mathbb{E}_{\bar{o}\sim\rho_{\mathcal{B}},\bar{a}\sim\bar{\pi}_{\bar{\theta}}(\cdot|\bar{o})} \nabla_{\theta_i} \log \pi_{\theta_i}(a_i|o_i) A_i(\bar{o},\bar{a}), \quad (3)$$

where

$$A_i(\bar{o},\bar{a}) = Q_{\phi_i}(\bar{o},\bar{a}) - \sum_{a_i'\in\mathcal{A}_i} \pi_i(a_i'|o_i) Q_{\phi_i}(\bar{o}, \bar{a}_{-i}(a_i'))$$

and $\bar{a}_{-i}(a_i')$ is the change of the i-th action in $\bar{a}$ to a_i'. It should be noted that MAAC in this work utilizes the same loss function but use fully-connected neural networks as critics instead of using an attention-based critic.

Discriminator

In MAGAIL, two kinds of discriminator models were proposed as described in Fig. 1. One is a centralized discriminator that takes all agents' observations and actions as its input and outputs multi-head classification for each agent. The other is a decentralized discriminator that takes each agent's local observations and actions and outputs a single-head classification. In the experiments of MAGAIL, it turns out that using a centralized discriminator leads to better performances than using decentralized discriminators when a handful of agents is considered. Let

$$\bar{D}_{\bar{\psi}}(\bar{o},\bar{a}) = (D_{\psi_1}(\bar{o},\bar{a}), ..., D_{\psi_N}(\bar{o},\bar{a}))$$

denote the vector-valued discriminator output. Note that this is a general expression for both types of discriminators since one can consider shared values among $\psi_1, ..., \psi_N$ for the centralized discriminator, whereas decentralized discriminators do not share values among parameters but ignore other agents' observations and actions, i.e., $D_{\psi_i}(o_i, a_i)$. For each training iteration, we train discriminator by using the following objective:

$$\underset{\bar{\psi}}{\operatorname{argmax}}\, \mathbb{E}_{\bar{o}^A,\bar{a}^A\sim\rho_{\bar{\pi}^A}}\left[\sum_{i=1}^{N}\log(1 - D_{\psi_i}(\bar{o}^A,\bar{a}^A))\right]$$
$$+ \mathbb{E}_{\bar{o}^E,\bar{a}^E\sim\rho_{\bar{\pi}^E}}\left[\sum_{i=1}^{N}\log D_{\psi_i}(\bar{o}^E,\bar{a}^E)\right]. \quad (4)$$

Therefore, discriminators are trained in a way that expert-like behavior gets higher values, whereas non-expert-like behavior results in lower values. For decentralized discriminators, each discriminator was trained by using an individual stochastic gradient optimizer, whereas a single optimizer was used for a centralized discriminator. It should be noted that the objective (4) implies the use of rollout samples from the policy $\bar{\pi}^A$ in the first expectation. In practice, however, we use the samples from the replay buffer (without off-policy correction) to enhance the sample-efficiency of discriminator training via sample reuse. Experimentally, ignoring off-policiness does not harm performance (Kostrikov et al. 2019).

Reward

In previous works on single-agent imitation learning (Kostrikov et al. 2019; Fu, Luo, and Levine 2018; Ho and Ermon 2016), three types of reward function was proposed;

- $r(o,a) = \log D(o,a)$ that has a negative value and is used for the tasks with per-step penalty.
- $r(o,a) = -\log(1 - D(o,a))$ that has a positive value and is used for the tasks with per-step bonus.
- $r(o,a) = \log D(o,a) - \log(1 - D(o,a))$ that can be used for both cases.

We experimentally checked all three cases and use $r_i(\bar{o},\bar{a}) = \log D_{\psi_i}(\bar{o},\bar{a})$ over all experiments, which leads to the much better performance and stabilized training.

Experiments

We consider the following questions in our experiments:

1. Is our algorithm sample-efficient in terms of the number of agent-environment interactions?
2. Is our algorithm sample-efficient in terms of the number of available expert demonstrations?
3. Is our algorithm scalable to a large number of agents?

To answer those questions, we consider two classes of environments where we respectively call small-scale and large-scale environments, both of which run on OpenAI Multi-agent Particle Environment (MPE)[1]. The small-scale environments (Lowe et al. 2017) are those used in MAGAIL and described as follows:

[1] https://github.com/openai/multiagent-particle-envs

- *Keep Away*− There are 2 agents ”reacher” and ”pusher”. This is a competitive game since reacher tries to reach the goal, while pusher tries push it away from the goal.
- *Cooperative Communication*−There are 2 agents, ”speaker” and ”listener”. One of three landmarks is randomly chosen as a target at each episode and its location can only be seen by speaker. However, speaker cannot move, whereas listener can observe speaker's message and move toward the target landmark. Thus, speaker and listener should cooperate with each other in order for listener to reach the landmark. Note that speaker and listener share their rewards.
- *Cooperative Navigation*− There are 3 agents and 3 landmarks, and the goal of agents is to cover as many landmarks as possible. All the agents are penalized with the minimum distance between all agents and landmarks as well as the collision among the agents. Note that all of the agents share their rewards.
- *Predator Prey*− There are 4 agents, one of them is ”prey” and the others are ”predators”. Prey is slightly faster than predators, and predators try to capture prey while prey try to run away from predators. Rewards are shared among predators, whereas prey gets the rewards different from those of predators.

We call the environments proposed in MAA2C (Iqbal and Sha 2019) as large-scale environments and use one of them to assess the scalability of multi-agent imitation:

- *Rover Tower*− There are an even number of agents, where half of them are ”rovers” and the others are ”towers”. At each episode, rovers and towers are randomly paired, and each tower has its own goal. Similar to *Cooperative Communication.*, towers cannot move but can communicate with rovers so that rovers can move toward corresponding goals. Each pair of rover and tower is negatively rewarded by the distance of the rover to its goal at each episode.

For all of the above environments, we consider three MARL algorithms:

- MACK (Song et al. 2018) is a multi-agent extension of ACKTR (Wu et al. 2017) and was first used in MAGAIL for multi-agent imitation learning. Each agent has its own centralized critic that takes every agent's observations and (preceding) actions as its inputs. These critics are used for generalized advantage estimation (GAE) (Schulman et al. 2016) to reduce the variance of policy gradients. Note that we implement MACK by refactoring PyTorch implementation of ACKTR[2]. For the fairness of our experiments, we improved MARL performance of MACK as much as we could. Although we do not report specific results in this work, we check that using heuristics such as advantage normalization[3], observation normalization with running mean statistics[4] and reward normalization for each agent (Iqbal and Sha 2019) leads to much stable and rapid training.

[2]https://github.com/ikostrikov/pytorch-a2c-ppo-acktr-gail

[3]https://github.com/openai/baselines/blob/master/baselines/trpo_mpi/trpo_mpi.py

[4]https://github.com/openai/baselines

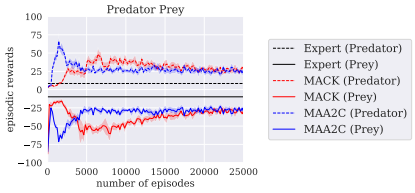

Figure 2: Episodic scores for each agent in *Predator Prey* environment. Note that 200 expert demonstration and decentralized discriminators are used.

- MAA2C is an MARL algorithm with attention critic. It uses a shared critic network among agents. We refactored the released implementation of MAA2C[5] and used it in this work.

- MAAC is a multi-agent extension of actor-critic methods. Each agent has its own critic that takes all agents' observations and actions. In contrast with MADDPG (Lowe et al. 2017) that uses the value gradient with Gumbel-Softmax relaxation (Jang, Gu, and Poole 2016), MAAC optimizes policies with the objective (3). Since MAAC does not share critic parameters among agents, the number of critic parameters exponentially increases as the number of agents increases. We check the performance of MAAC to verify the minimum cumulative reward that MAA2C is desired to achieve and assess the effectiveness of parameter sharing of MAA2C.

For experts' policies, we trained MAA2C agents over 50000 episodes for small-scale environments and 100000 episodes for large-scale environments, where the expected cumulative reward of all agents are shown to converge. Over all tasks, we generate 500 episodes from the trained MAA2C agents, where the actions in those episodes are always taken with the largest probability. For each task, we vary the number of available demonstrations among 50, 100 and 200.

To measure the multi-agent imitation performance, we use the *normalized score similarity* (NSS) defined as

$$\mathrm{NSS} = \frac{1}{N}\sum_{i=1}^{N}\frac{\mathrm{score}_{A_i} - \mathrm{score}_{R_i}}{\mathrm{score}_{E_i} - \mathrm{score}_{R_i}},$$

where score_{A_i} is the i-th agent's (episode) score during training, score_{E_i} is the i-th expert's *average* score for experts' demonstrations, and score_{R_i} is the *average* score of the i-th agent when uniformly random actions were taken by all agents. Intuitively, NSS gets close to 1 if every agent shows expert-like behavior since such behavior will lead to the experts' score. One advantage of NSS is that we can evaluate multi-agent imitation performance for both competitive and cooperative tasks. Our experiments show that it was effective measure for most of our target tasks. Additionally, we calculate both score_{E_i} and score_{R_i} from 500 trajectories for each i-th agent and report them in Table 1 and Table 2. Finally, we report NSS in Figure 4 and 5.

Number of Expert Trajectories	Agent	Discriminator Type	Keep Away (2)		Cooperative Communication (2)	Cooperative Navigation (3)	Predator Prey (4)	
			Pusher (1)	Reacher (1)			Predator (3)	Prey (1)
-	Random	-	114.401 ± 5.087	-25.910 ± 1.034	-57.196 ± 2.404	-178.575 ± 4.108	4.240 ± 0.811	-22.285 ± 2.144
-	Expert	-	44.449 ± 1.673	-9.688 ± 0.331	-15.290 ± 0.811	-77.129 ± 1.766	8.340 ± 1.178	-9.910 ± 1.217
200	MACK	Centralized	49.869 ± 1.776	-11.021 ± 0.363	-36.933 ± 1.629	-125.650 ± 2.229	27.800 ± 2.624	-47.190 ± 4.124
		Decentralized	$\mathbf{46.342 \pm 1.594}$	-10.165 ± 0.321	-29.936 ± 1.143	-95.823 ± 1.741	36.400 ± 3.234	-43.933 ± 3.017
	MAAC	Centralized	47.680 ± 1.597	-10.375 ± 0.324	$\mathbf{-18.602 \pm 0.687}$	-81.704 ± 1.875	25.360 ± 2.746	-29.672 ± 2.854
		Decentralized	46.621 ± 1.316	$\mathbf{-10.157 \pm 0.266}$	-18.711 ± 0.684	$\mathbf{-79.572 \pm 1.866}$	28.400 ± 2.218	-31.363 ± 2.242
	MAA2C	Centralized	47.283 ± 1.522	-10.292 ± 0.308	-18.613 ± 0.712	-82.570 ± 1.957	$\mathbf{24.680 \pm 2.104}$	$\mathbf{-28.702 \pm 2.251}$
		Decentralized	47.009 ± 1.519	-10.229 ± 0.306	-18.729 ± 0.691	-79.693 ± 1.777	27.160 ± 2.002	-30.352 ± 1.956
100	MACK	Centralized	50.118 ± 1.566	-11.066 ± 0.320	-36.971 ± 1.500	-129.617 ± 2.290	26.320 ± 2.532	-42.298 ± 2.780
		Decentralized	$\mathbf{46.262 \pm 1.604}$	-10.168 ± 0.324	-30.856 ± 1.175	-96.379 ± 1.724	35.780 ± 2.776	-44.671 ± 3.019
	MAAC	Centralized	48.034 ± 1.627	-10.431 ± 0.327	$\mathbf{-19.620 \pm 0.749}$	-85.711 ± 2.038	$\mathbf{23.340 \pm 1.950}$	$\mathbf{-28.135 \pm 1.776}$
		Decentralized	46.936 ± 1.608	-10.227 ± 0.323	-20.404 ± 0.699	$\mathbf{-81.819 \pm 1.896}$	25.680 ± 2.245	-29.899 ± 2.192
	MAA2C	Centralized	46.596 ± 1.511	$\mathbf{-10.147 \pm 0.302}$	-19.630 ± 0.769	-84.695 ± 1.982	23.740 ± 2.409	-28.245 ± 2.496
		Decentralized	46.932 ± 1.581	-10.220 ± 0.319	-20.458 ± 0.636	-82.764 ± 1.913	25.400 ± 2.243	-28.943 ± 2.213
50	MACK	Centralized	49.099 ± 1.546	-10.840 ± 0.320	-38.574 ± 1.539	-135.950 ± 2.494	27.700 ± 2.749	-43.418 ± 3.059
		Decentralized	$\mathbf{44.937 \pm 1.541}$	$\mathbf{-9.895 \pm 0.310}$	-34.363 ± 1.175	-98.657 ± 1.658	39.100 ± 3.053	-47.136 ± 3.104
	MAAC	Centralized	49.205 ± 1.655	-10.654 ± 0.334	$\mathbf{-21.536 \pm 0.776}$	-88.918 ± 2.102	24.560 ± 3.188	-37.536 ± 6.441
		Decentralized	47.206 ± 1.610	-10.270 ± 0.325	-23.670 ± 0.846	$\mathbf{-87.772 \pm 1.956}$	27.420 ± 2.489	$\mathbf{-31.382 \pm 2.516}$
	MAA2C	Centralized	48.630 ± 1.509	-10.545 ± 0.301	-21.684 ± 0.845	-88.150 ± 2.051	$\mathbf{23.500 \pm 2.571}$	-31.826 ± 3.859
		Decentralized	47.283 ± 1.609	-10.282 ± 0.327	-23.495 ± 0.956	-88.597 ± 1.957	26.260 ± 2.318	-33.849 ± 3.551

Table 1: 95% confidence intervals of episodic scores in the small-scale environments. The numbers in brackets is the number of corresponding agents. Random agents and experts are averaged over 500 episodes. For MACK, MAAC and MAA2C, the scores of the last 200 episodes over 10 different runs are considered. The agents are trained over 5000 episodes for *Keep Away*, *Cooperative Communication*, *Cooperative Navigation* and 10000 episodes for *Predator Prey*, respectively. In most cases except *Keep Away*, using MAAC and MAA2C results in the score closest to the experts' score (See bold numbers). In addition, both centralized and decentralized discriminators show the similar performances for MAAC and MAA2C in the small-scale environments.

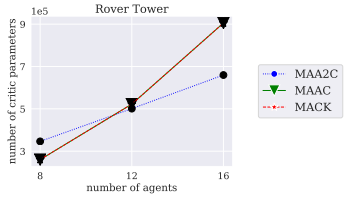

Figure 3: Number of critic parameters over all agents.

Small-Scale Environments

The results in the small-scale environments are summarized in Figure 4 and Table 1, where each column indicates the results for different environments, whereas each row indicates the use of 50, 100, 200 expert trajectories from bottom to top. Over all small-scale environments, we found that our method (blue curves) converges much faster than MAGAIL (red curves), which proves the sample-efficiency of our method. Specifically, within 1000 episodes for *Keep Away*, 3000 episodes for *Cooperative Communication* and *Cooperative Navigation*, our method converges, whereas MAGAIL fails to converge. In *Predator Prey*, however, we found that all algorithms failed to achieve NSS close to 1. To analyze this result, we plot the average episodic scores for *Predator Prey* for 200 expert demonstration in Figure 2. At the initial training phase (before episode 1000), predators' episodic score extremely increases while prey's score decreases. That is, the learned predators perform better whereas the learned preys are caught more often than in the demonstrations. After then, it converges to a different equilibrium than the experts' one. We think that this is due to the inherent multi-modality present in the demonstrated expert prey's trajectories (i.e. the prey can choose to escape from the left, the right, up or down). On the other hand, the behavior of the expert predators is more straightforward to capture as they always rush towards the prey. Therefore, training prey's discriminator is much more complicated and may result in a poor learning signal for the forward RL algorithm, eventually leading to an under-performing prey. In addition, the performance of MAAC is shown to be almost identical to that of MAA2C in the small-scale environments. This is because both MAA2C and MAAC performs well on the tasks where there are a handful of agents.

[5]https://github.com/shariqiqbal2810/MAAC

Large-Scale Environments

In the large-scale environments, we test the scalability of our method by varying the number of agents as shown in Figure 5 and Table 2. One interesting observation is that using a centralized critic leads to poor performances compared to using decentralized discriminators. This is in contrast with the results in the small-scale environments, where both types of discriminators lead to the similar performances. We think that this is due to the increased size of discriminator inputs, which makes the input variance increase and discriminators difficult to be optimized. Another interesting observation is that the convergence of MAGAIL, the imitation learning with MACK, requires much more agent-environment interactions for a larger number of agents even when decentralized discriminators are used, which shows the poor scalability of MAGAIL. We also find that MAA2C performs similar to MAAC for 8 agents and slightly more sample-efficient for 12 and 16 agents while the number of parameters used for MAA2C is less than that of MAAC for 12 and 16 agents (See Figure 3[6]). Note that both numbers of parameters for MACK and MAAC exponentially increase because of the increasing size of joint input spaces. This proves that the imitation learning with MAA2C is much more scalable to a large number of agents without losing the gain.

[6]Note that considering the number of critic parameters is sufficient since the number of policy parameters and decentralized discriminator parameters linearly increases.

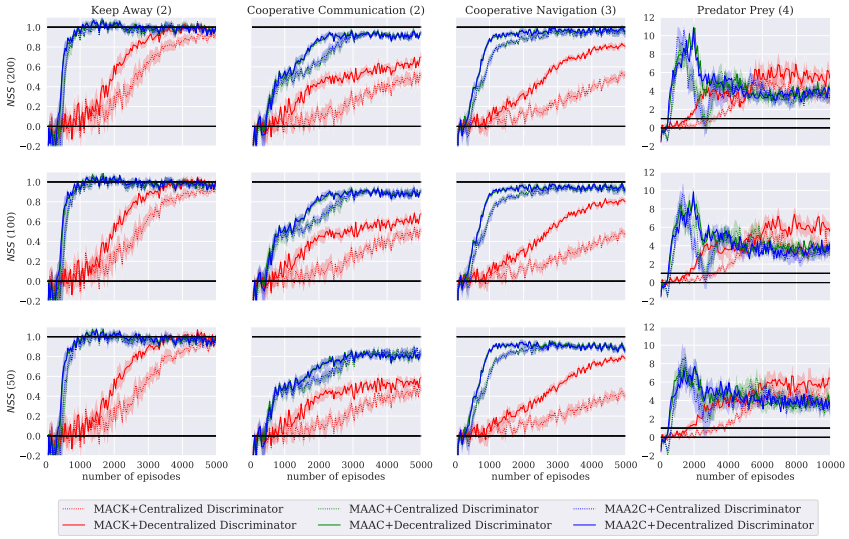

Figure 4: Imitation learning performance in small-scale environments. Each column indicates a different environment, and the columns are ordered by the number of agents (from left to right). Each row indicates the different number of experts' demonstration (200, 100, 50 from top to bottom). x-axis of each subplot is the number of episodes used for imitation learning, and y-axis is NSS. Note that the imitation learning with either MAAC or MAA2C converges much faster than that of MACK regardless of the discriminator type. In addition, the methods with decentralized discriminators converge faster than corresponding centralized counterparts.

Number of Expert Trajectories	Agent	Discriminator Type	Rover Tower (8)	Rover Tower (12)	Rover Tower (16)
-	Random	-	-8.350 ± 7.324	-7.691 ± 7.399	-7.016 ± 7.447
-	Expert	-	127.506 ± 7.136	126.258 ± 7.149	123.905 ± 7.349
200	MACK	Centralized	33.569 ± 3.812	7.464 ± 3.014	-1.463 ± 2.408
		Decentralized	$\mathbf{127.934 \pm 3.994}$	$\mathbf{115.466 \pm 3.374}$	106.340 ± 2.480
	MAAC	Centralized	54.347 ± 8.062	17.100 ± 2.947	5.405 ± 2.561
		Decentralized	125.312 ± 3.911	107.976 ± 3.872	97.247 ± 3.282
	MAA2C	Centralized	15.211 ± 6.080	11.076 ± 2.879	0.692 ± 2.280
		Decentralized	126.877 ± 3.973	114.423 ± 3.406	$\mathbf{109.347 \pm 2.728}$
100	MACK	Centralized	18.361 ± 3.343	3.537 ± 2.772	-2.923 ± 2.458
		Decentralized	116.389 ± 4.313	94.775 ± 3.914	74.764 ± 2.602
	MAAC	Centralized	46.637 ± 5.474	13.989 ± 3.340	-1.290 ± 2.409
		Decentralized	120.513 ± 4.091	91.911 ± 3.486	75.996 ± 2.594
	MAA2C	Centralized	15.055 ± 6.648	5.014 ± 2.451	-3.105 ± 2.440
		Decentralized	$\mathbf{122.531 \pm 3.988}$	$\mathbf{99.292 \pm 3.504}$	$\mathbf{83.044 \pm 3.517}$
50	MACK	Centralized	13.887 ± 3.789	0.153 ± 2.991	-3.463 ± 2.178
		Decentralized	84.555 ± 4.892	52.724 ± 3.591	32.278 ± 2.611
	MAAC	Centralized	31.300 ± 3.853	1.738 ± 2.470	-7.524 ± 2.238
		Decentralized	98.989 ± 4.378	60.110 ± 4.487	42.425 ± 2.773
	MAA2C	Centralized	22.700 ± 3.824	2.198 ± 2.802	-10.411 ± 3.475
		Decentralized	$\mathbf{102.173 \pm 4.441}$	$\mathbf{69.390 \pm 4.446}$	$\mathbf{50.971 \pm 3.411}$

Table 2: 95% confidence intervals of episodic scores in the large-scale environments. The numbers in brackets is the number of corresponding agents. Random agents and experts are averaged over 500 episodes. For MACK, MAAC and MAA2C, the scores of the last 200 episodes over 10 different runs are considered. The agents are trained over 50000 episodes and the average score over rovers is reported since it is approximately the same as that over towers due to the uniform pairing and the shared reward between rovers and towers. Note that the learning with the centralized discriminator performs poorly, whereas that with decentralized discriminators performs much better. In addition for 200 expert trajectories, there's a little performance gap between MACK and MAA2C with decentralized discriminators, but the gap becomes larger as the number of available expert trajectories decreases.

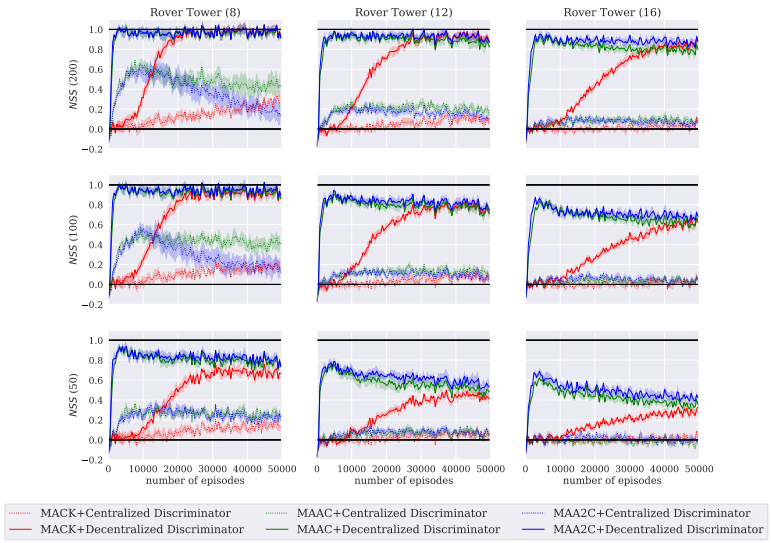

Figure 5: Imitation learning performance in large-scale environments. Each column indicates the different number of agents (8, 12, 16 from left to right). Each row indicates the different number of experts' demonstration (200, 100, 50 from top to bottom). x-axis of each subplot is the number of episodes used for imitation learning, and y-axis is NSS. The sample-efficiency of the imitation learning with MACK is severely affected by the number of agents, whereas that with either MAA2C or MAAC is much more robust to the number of agents. Note that the methods with centralized discriminators performs poorer than their decentralized counterparts in the large-scale environments. Also, as the number of agents increases and the number of expert trajectories decreases, there is a tendency that the final performance gap between the learning with MAA2C and that with *unshared* centralized critics (MAAC and MACK) increases.

Effect of number of experts' demonstration

For both small-scale and large-scale environments, we vary the number of available expert demonstration among 50, 100, 200 and check its effect on the imitation learning performance. In the small-scale environments, we find that there's almost no performance degradation, whereas it becomes significant for a large number of agents. We think this is due to the increased size of joint observation-action spaces, which makes discriminators easily overfitted to the experts' demonstrations. One way to address this issue is to use Bayesian discriminator (Jeon, Seo, and Kim 2018) so that the discriminator becomes robust to the small amount of expert data, but we leave it for our future work. In addition, it is worthwhile to note that the performance degradation of MACK and MAAC is much more severe than MAA2C. We guess this is due to the fact that MAA2C uses critics with shared parameters, which leads to a bit more robust training compared to MAAC and MACK.

Discussion and Conclusion

We propose a sample-efficient imitation learning algorithm that is much more scalable to the large-scale environments compared to the existing method. Our proposed methods are sample-efficient due to the off-policy MARL algorithms with experience replay. Also, we show that the best performance can be achieved by using the shared attention critic of which the number of parameters linearly increases with respect to the number of agents.

One interesting result is that imitation learning with decentralized discriminators performs extremely well over both small-scale and large-scale environments. Since decentralized discriminators only consider their own local observations and actions, however, there need to be some sources of coordination outside discriminators to make multi-agent imitation learning work. We think about two sources of coordination that leads to successful imitation. The first source is the use of centralized critics that take joint observations and actions as inputs as it was in many MARL algorithms. In MAGAIL (Song et al. 2018), it was shown that agent-wise GAIL based on independent RL failed to achieve a good performance, which shows the effectiveness of centralized critics. The second source, which we believe is critical, is due to sampling experts' joint observations and actions that happened in the same time step. Since experts' joint experience how the information to coordinate at the specific time step, decentralized discriminators can be trained toward the coordination of experts. Thanks to those sources of coordination, decentralized discriminators can focus on local experiences, which highly reduces the input variance and results in the better performance.

References

Bellman, R. 1957. A Markovian decision process. *Journal of mathematics and mechanics* 679–684.

Degris, T.; White, M.; and Sutton, R. S. 2012. Off-policy actor-critic. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 179–186.

Foerster, J. N.; Farquhar, G.; Afouras, T.; Nardelli, N.; and

Whiteson, S. 2018. Counterfactual multi-agent policy gradients. In *The 32nd AAAI Conference on Artificial Intelligence*.

Fu, J.; Luo, K.; and Levine, S. 2018. Learning robust rewards with adverserial inverse reinforcement learning. In *Proceedings of the 6th international conference on learning representations (ICLR)*.

Fujimoto, S.; Hoof, H.; and Meger, D. 2018. Addressing function approximation error in actor-critic methods. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 1582–1591.

Ho, J., and Ermon, S. 2016. Generative adversarial imitation learning. In *Advances in neural information processing systems (NeurIPS)*, 4565–4573.

Iqbal, S., and Sha, F. 2019. Actor-attention-critic for multi-agent reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2961–2970.

Jang, E.; Gu, S.; and Poole, B. 2016. Categorical reparameterization with Gumbel-softmax.

Jeon, W.; Seo, S.; and Kim, K.-E. 2018. A Bayesian approach to generative adversarial imitation learning. In *Advances in neural information processing systems (NeurIPS)*, 7429–7439.

Kim, K.-E., and Park, H. S. 2018. Imitation learning via kernel mean embedding. In *Proceedings of the 32nd AAAI conference on artificial intelligence*.

Kostrikov, I.; Agrawal, K. K.; Dwibedi, D.; Levine, S.; and Tompson, J. 2019. Discriminator-Actor-Critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. In *Proceedings of the 7th international conference on learning representations (ICLR)*.

Le, H. M.; Yue, Y.; Carr, P.; and Lucey, P. 2017. Coordinated multi-agent imitation learning. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 1995–2003.

Li, Y.; Swersky, K.; and Zemel, R. 2015. Generative moment matching networks. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 1718–1727.

Littman, M. L. 1994. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings*. Elsevier. 157–163.

Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; and Mordatch, I. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. 6379–6390.

Ross, S.; Gordon, G.; and Bagnell, D. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the 14th international conference on artificial intelligence and statistics (AISTATS)*, 627–635.

Sanghvi, N.; Yonetani, R.; and Kitani, K. 2019. Modeling social group communication with multi-agent imitation learning. *arXiv preprint arXiv:1903.01537*.

Sasaki, F.; Yohira, T.; and Kawaguchi, A. 2019. Sample efficient imitation learning for continuous control. In *Proceedings of the 7th international conference on learning representations (ICLR)*.

Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.; and Abbeel, P. 2016. High-dimensional continuous control using generalized advantage estimation.

Song, J.; Ren, H.; Sadigh, D.; and Ermon, S. 2018. Multi-agent generative adversarial imitation learning. In *Advances in neural information processing systems (NeurIPS)*, 7461–7472.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems (NeurIPS)*, 5998–6008.

Wu, Y.; Mansimov, E.; Grosse, R. B.; Liao, S.; and Ba, J. 2017. Scalable trust-region method for deep reinforcement learning using Kronecker-factored approximation. In *Advances in neural information processing systems (NeurIPS)*, 5279–5288.

Yang, Y.; Luo, R.; Li, M.; Zhou, M.; Zhang, W.; and Wang, J. 2018. Mean field multi-agent reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 5567–5576.

Ziebart, B. D.; Maas, A. L.; Bagnell, J. A.; and Dey, A. K. 2008. Maximum entropy inverse reinforcement learning. In *Proceedings of the 23rd AAAI conference on artificial intelligence*, volume 8, 1433–1438.