Decentralized Multi-Agent Reinforcement Learning via Distribution Matching

Caroline Wang*
The University of Texas at Austin
Austin, Texas, US
caroline.l.wang@utexas.edu

Ishan Durugkar*
The University of Texas at Austin
Austin, Texas, US
ishand@cs.utexas.edu

Elad Liebman*
Spark Cognition
Austin, Texas, US
eliebman@sparkcognition.com

Peter Stone
Sony AI
Austin, Texas, US
pstone@cs.utexas.edu

ABSTRACT

Current approaches to multi-agent cooperation rely heavily on centralized mechanisms or explicit communication protocols to ensure convergence. In this paper, we study the problem of decentralized multi-agent learning, without resorting to explicit coordination schemes. We propose the use of distribution matching to facilitate independent agents' coordination. Each individual agent will match a target distribution of concurrently sampled trajectories from a joint expert policy. Such distribution matching allows us to theoretically show that the agents will converge to a stationary joint distribution if the sampled trajectories include observations of the other agents' behaviors. Experimental validation on the StarCraft domain shows that combining the reward for distribution matching with the environment reward allows agents to outperform a fully distributed baseline and an uncoordinated imitation learning scheme.

KEYWORDS

Reinforcement Learning, Multi-agent RL, Distribution Matching

1 INTRODUCTION

Multi-agent reinforcement learning (MARL) [19] is a paradigm for learning agent policies that may interact with each other in cooperative or competitive settings. MARL algorithms can be applied to train agents to play soccer [36], two-player zero-sum games [33, 34], and ad-hoc teamwork tasks [3]. Training multiple agents at once can be challenging, since an agent updating its own strategy induces a nonstationary environment for other agents, potentially leading to training instabilities. To overcome these issues, agent policies can be set up as a monolith, such that the agents can be trained together but then deployed individually [7, 26], or coordinated through some form of communication among agents. [14, 20, 21].

Fully decentralized training of agent policies remains an open problem in MARL. Independent training is desirable in settings with a large number of agents, where agents are faced with changing environments [22], agents must team up in an ad hoc fashion [3], when agents learn in a lifelong manner [38], or when ensuring privacy is a concern [17].

This paper considers learning MARL policies in a decentralized manner without explicit communication or a central training mechanism, by using individual distribution matching against demonstrations to assist learning. In the proposed approach, the individual agents learn to match the state (or observation) visitation distribution of demonstrations from corresponding expert agents that have been trained together on the task of interest. A scenario in which such demonstrations would be realistic to expect is in the state-only imitation learning setting, where human experts could provide a rich source of demonstrations. For example, demonstrations of expert football/soccer players could be useful when training robot players [36]. Another natural example for matching demonstrations is that of human medical teams trained to accomplish difficult, specialized tasks.

In the theoretical analysis, the paper shows how each agent attempting to individually match the visitation distribution of its corresponding expert demonstrations will lead to them learning the joint expert policy, as long as the demonstrations were sampled from expert policies that are in an equilibrium with respect to some task.

The paper then proposes a practical algorithm that leverages the above convergence properties, and presents each agent with a mixed reward consisting of a cost function to encourage coordination through distribution matching and the environment reward. Experimental evaluation in the StarCraft domain shows that this approach accelerates learning compared to a distributed learning of the environment reward in multiple scenarios. The evaluation also shows that this benefit is obtained even when the demonstrations are from a set of experts that are only partially competent at the task to be accomplished. The ablations then tease apart the properties of the demonstrations needed to assist with the learning. These ablations show that the expert demonstrations given to each agent do not have to be from the same trajectories, i.e., they do not need to be recorded concurrently. It is sufficient for them to be from the policies that were trained concurrently. However, demonstrations from policies that were not trained together do not assist learning in a similar manner.

2 RELATED WORK

Cooperation in the Decentralized Setting. Many algorithms for multi-agent cooperation tasks require some degree of information sharing between agents. The information sharing can take many

*Equal contribution.

Proc. of the Adaptive and Learning Agents Workshop (ALA 2022), Cruz, Hayes, da Silva, Santos (eds.), May 9-10, 2022, Online, https://ala2022.github.io/. 2022.

forms. In some methods, agents directly share model components. For instance, centralized training decentralized execution (CTDE) methods use a single centralized critic that aggregates information during training, but is no longer required at execution time [7, 21, 26, 37, 43]. In practical implementations of CTDE methods, agent networks often share parameters during training as well, constituting another form of model sharing.

Assuming that agents share model components during training is not always practical. Another body of work studies the decentralized setting, where agents (and critics) are distinct models, and information is communicated between agents. There are various ways to accomplish such communication. For one, agents are allowed to directly communicate information to each other [14, 18]. In others, there is a central network that provides coordinating signals to all agents [12, 20]. The information communicated can be leveraged in different ways. Wen et al. [42] propose multi-agent trust region learning, where each agent has knowledge of the other agents' policies during training, and use this knowledge to ensure that the best response of each agent does not cause the joint policies to deviate too much. In contrast, this work studies the fully decentralized setting without communication during training: separate agent policies may observe each other, but no sharing of information via a shared critic or communication protocols is permitted during training or execution.

To our knowledge, relatively few works consider decentralized cooperation without communication. Early work analyzed simple cases where two agents with similar but distinct goals could cooperate for mutual benefit under a rationality assumption [9, 28]. More recently, Godoy et al. [10] propose the ALAN system for multi-agent navigation, in which agents learn via a multi-armed bandits method that does not require any communication. Jiang and Lu [15] study the decentralized multi-agent cooperation in the *offline* setting—in which each agent can only learn from its own data set of pre-collected behavior without communication—and propose a learning technique that relies on value and transition function error correction.

Distribution Matching in MARL. Ho and Ermon [13] originally proposed adversarial distribution matching as a way to perform imitation learning in the single agent setting (the GAIL algorithm). Song et al. [35] extend GAIL to the multi-agent setting in certain respects, by setting up imitation learning as searching for a Nash equilibrium, and assuming that a unique equilibrium exists. Their experiments focus on training the agent policies in the CTDE paradigm, rather than the fully distributed setting.

Wang et al. [41] study using copula functions to explicitly model the dependence between marginal agent policies for multi-agent imitation learning. Durugkar et al. [6] show that when faced with a cooperative task, balancing individual preferences with the shared task reward can accelerate progress on the shared task for some mixing schemes of the preference reward and the shared task. One of the preferences they utilized was to match the state-action visitation distribution of some strategies to solve the shared task. In contrast to the above works, the goal of this paper is not to study imitation learning, but rather to study how distribution matching by independent agents can enhance performance in cooperative tasks.

3 BACKGROUND

This section describes the problem setup for MARL, as well as the imitation learning and distribution matching problems.

3.1 Markov games

A Markov game [19] or a stochastic game [8] with K agents is defined as a tuple $\langle K, \mathcal{S}, \boldsymbol{\mathcal{A}}, \rho_0, \mathcal{T}, \boldsymbol{R}, \gamma \rangle$, where $\mathcal{S}$ is the set of states, and $\boldsymbol{\mathcal{A}} \equiv \mathcal{A}^K$ is the product of the set of actions $\mathcal{A}$ available to each agent. The initial state distribution is described by $\rho_0 : \Delta(\mathcal{S})$, where $\Delta(\cdot)$ indicates a distribution over the corresponding set. The transitions between states are controlled by the transition distribution $\mathcal{T} : \mathcal{S} \times \mathcal{A}_0 \times \mathcal{A}_1 \times \ldots \times \mathcal{A}_{K-1} \longmapsto \Delta(\mathcal{S})$. Each agent i acts according to a policy $\pi_i : \mathcal{S} \longmapsto \Delta(\mathcal{A}_i)$, and the joint policy $\boldsymbol{\pi}$ is the product of the individual agent policies. Note that each agent observes the full state. We use subscript $-i$ to refer to all agents except i. For example, π_{-i} is used to refer to the agent policies, $\{\pi_0, \ldots, \pi_{i-1}, \pi_{i+1}, \ldots, \pi_{K-1}\}$.

Each agent i is also associated with a reward function $R_i : \mathcal{S} \times \mathcal{A}_0 \times \ldots \times \mathcal{A}_{K-1} \longmapsto \mathbb{R}$. The agent aims to maximize its return $\mathbb{E}_{\boldsymbol{\pi}}[\sum_{t=0}^{\infty} \gamma^t r_{i,t}]$, where $r_{i,t}$ is the reward received by agent i at time step t, and the discount factor $\gamma \in [0, 1)$ specifies how much to discount future rewards. In the cooperative tasks considered by this paper, the rewards are identical across agents.

In Markov games, the optimal policy of an agent depends on the policies of the other agents. The *best response* policy is the best policy an agent can adopt given the other agent's policies $\pi_i^* = \text{argmax}_{\pi_i} \mathbb{E}_{\pi_i, \pi_{-i}}[\sum_{t=0}^{\infty} \gamma^t r_{i,t}]$. If no agent can unilaterally change its policy without reducing their return, then the policies are considered to be in a *Nash equilibrium*. That is, $\forall i \in [0, K-1], \forall \hat{\pi}_i \neq \pi_i, \mathbb{E}_{\pi_i, \pi_{-i}}[\sum_{t=0}^{\infty} \gamma^t r_{i,t}] \geq \mathbb{E}_{\hat{\pi}_i, \pi_{-i}}[\sum_{t=0}^{\infty} \gamma^t r_{i,t}]$.

The theory presented in Section 4 deals with the above fully observable setting. However, the experiments are conducted in partially observable MDPs (POMDPs), which can be formalized as Dec-POMDPs in the multi-agent setting [24]. Dec-POMDPs include two additional elements: the set of observations Ω and each agent's observation function $O_i : \mathcal{S} \longmapsto \Delta(\Omega)$.

3.2 Imitation Learning and Distribution Matching

Imitation learning [2, 29, 31] is the problem setting where an agent tries to mimic trajectories $\{\xi_0, \xi_1, \ldots\}$ where each ξ is a trajectory $\{(s_0, a_0), (s_1, a_1), \ldots\}$ demonstrated by an expert policy π_E.

Various methods have been proposed to address the imitation learning problem. Behavior cloning [1] treats the expert's trajectories as labeled data and applies supervised learning to recover the maximum likelihood policy. Another approach instead relies on reinforcement learning to learn the underlying expert policy, where the required reward function is recovered using inverse reinforcement learning (IRL) [23]. $\texttt{IRL}(\pi_E)$ aims to recover a reward function under which the trajectories demontrated by π_E are optimal.

For agent i,

$$\rho_{\pi_i, \pi_{-i}}(s, a) := (1 - \gamma)\pi_i(a|s) \sum_{t=0}^{\infty} \gamma^t p(s_t = s | \pi_i, \pi_{-i})$$

refers to the marginal state-action visitation distribution of agent i's policy π_i, given the other agents' policies π_{-i}. For a single agent,

$\rho_{\pi_i,\pi_{-i}}(s,a)$ is dependent only on that agent's policy and the environment transition function. Ho and Ermon [13] show that in the single agent setting, a policy that minimizes the mismatch of its state-action visitation distribution to the one induced by the expert's trajectories and maximizes its causal entropy $H(\pi)$ is a solution to the $\texttt{RL} \circ \texttt{IRL}(\pi_E)$ problem. The causal entropy $H(\pi)$ is defined as:

$$\begin{aligned} H(\pi) &:= \mathbb{E}_{\pi}\left[-\log \pi(a|s)\right] \\ &= \mathbb{E}_{s_t,a_t\sim\pi}\left[-\sum_{t=0}^{\infty}\gamma^t \log \pi(a_t|s_t)\right]. \end{aligned}$$

In the multi-agent setting, imitation learning has the added complexity that the expert trajectories are generated by the interaction of multiple expert policies $\pi_{E_0},\ldots,\pi_{E_K}$. Successful imitation in this setting thus involves the coordination of all K agents' policies.

Song et al. [35] show that if there is a unique Nash equilibrium, then the solution to a similar $\texttt{MARL} \circ \texttt{MAIRL}(\boldsymbol{\pi}_E)$ formulation is the expert policies.

When comparing distributions, a metric of interest is the Wasserstein distance [25, 40] which is a measure of the amount of work needed to convert one distribution to another optimally, where the work is defined in terms of a ground metric d in the metric space on which these distributions are defined. More concretely, suppose we have a metric space $(\mathcal{M}, d)$ where $\mathcal{M}$ is a set and d is a metric on $\mathcal{M}$. For two distributions μ and ν with finite moments on the set $\mathcal{M}$, the Wasserstein-p distance is denoted by:

$$W_p(\mu,\nu) := \inf_{\zeta\in Z(\mu,\nu)} \mathbb{E}_{(X,Y)\sim\zeta}\left[d(X,Y)^p\right]^{1/p} \tag{1}$$

where Z is the space of all possible couplings between μ and ν. In other words, Z is the space of all possible distributions $\zeta \in \Delta(\mathcal{M}\times\mathcal{M})$ whose marginals are μ and ν respectively. Finding this optimal coupling tells us what is the least amount of work, as measured by d, that needs to be done to convert μ to ν. We use $W(\mu,\nu)$ to denote the Wasserstein-1 distance between distributions μ and ν hereafter.

4 THEORETICAL FOUNDATION

This section provides theoretical grounding for the core proposition of this paper. First, it shows that if N agents are minimizing the distribution mismatch to demonstrations generated by experts that are trained together to perform a task [*], then for each agent, minimizing the distribution mismatch to its respective demonstrations will result in all agent policies converging to a Nash equilibrium with respect to each distribution matching reward. Second, it shows that if the agents are learning to maximize the mixture of an extrinsic task reward and a distribution mismatch cost—computed with respect to demonstrations by expert policies that do successfully maximize the task reward—then the agent policies will converge to a Nash equilibrium with respect to the joint reward.

Note that there can be an inherent tension between multi-agent learning and achieving single-agent imitation objectives. As a motivating example, let us consider a simple four tile grid world, where only one agent is allowed on a tile at a time:

$$\begin{bmatrix} A_{11}, A_{12} \\ A_{21}, A_{22} \end{bmatrix}.$$

Suppose there are two agents. Each agent i attempts to match a simple joint state-action distribution, consisting of the ith agent occupying tile A_{11}, and the other agent occupying one of the three remaining tiles. It is impossible for both agents to fully match their desired distributions. What they will end up doing instead is largely dependent on the learning scheme. For example, one possible policy the agents could jointly execute is to take turns on the A_{11} tile.

The example above illustrates that for each agent to completely match its desired distribution, the state-action distributions for all agents must be compatible in some way. We formalize this notion of compatibility in terms of the state-action visitation distributions of each of the expert policies.

Let the state-action visitation distribution of a joint policy $\boldsymbol{\pi} = \langle \pi_1, \ldots \pi_N \rangle$ be:

$$\rho_{\boldsymbol{\pi}}(s,\boldsymbol{a}) := (1-\gamma)\prod_{i=1}^{N}\pi_i(a_i|s)\sum_{t=0}^{\infty}\gamma^t P(s_t = s|(\boldsymbol{\pi})). \tag{2}$$

Definition 1. *State-action visitation distributions $\rho_{\pi_i,\hat{\pi}_{-i}}$ from a collection of N policies $\{\pi_i\}_{i=1}^N$ (where $\hat{\pi}_{-i}$ are the other agent policies executed with π_i to get the state-action visitation distribution $\rho_{\pi_i,\hat{\pi}_{-i}}$) are* compatible *if there exists a joint policy $\boldsymbol{\pi}' = \langle \pi'_1, \ldots, \pi'_N \rangle$ with the joint state-action visitation distribution $\rho_{\boldsymbol{\pi}}(s,\boldsymbol{a})$ (Equation 2) such that the marginal state-action visitation distribution for agent i*

$$\begin{aligned} \rho_{\pi'_i,\pi'_{-i}}(s,a) &:= (1-\gamma)\pi'_i(a|s)\sum_{t=0}^{\infty}\gamma^t P(s_t = s|\pi'_i,\pi'_{-i}) \\ &= \rho_{\pi_i,\hat{\pi}_{-i}}(s,a). \end{aligned}$$

for all i and for all $s \in \mathcal{S}, a \in \mathcal{A}$.

Observation 1. *N expert policies that are trained in the same environment to perform a task induce* compatible *individual state-action visitation distributions.*

Observation 1 provides a practical way to obtain compatible demonstrations. Note that the N expert policies do not have to successfully maximize their task rewards. If demonstrations are sampled by executing the N co-trained experts together, it is immediate that the individual state-action visitation distributions represented by the demonstrations are compatible.

Next we consider the use of reinforcement learning to minimize the distribution mismatch of an agent's state-action visitation distribution to a target distribution. To do so, we define a reward function that can be used to minimize this distribution mismatch.

Definition 2. *A* distribution matching reward *is a reward function which, if used to train an RL agent, leads to a minimization of the distribution mismatch between the agent's state-action visitation distribution and a target state-action visitation distribution.*

An example of such a reward is the reward used in GAIL [13] $R_{gail}(s,a) = -\log(D(s,a))$, where D is the discriminator trained to distinguish between the agent's state-action visitation distribution and the target distribution. We denote the distribution mismatch between the agent's state-action visitation distribution $\rho_{\pi_i,\pi_{-i}}$ and

[*] For this claim, it is not assumed that the experts have successfully maximized the task reward.

a given target state-action visitation distribution $\rho_{\pi_{E_i},\pi_{-E_i}}$ using the Wasserstein distance $W(\rho_{\pi_i,\pi_{-i}}, \rho_{\pi_{E_i},\pi_{-E_i}})$.

Claim 1. *Consider the problem setting where N agents are each attempting to minimize the distribution mismatch to demonstrations from compatible state-action visitation distributions. The individual demonstration policies that generated the demonstrations are a Nash equilibrium for the imitating agents with respect to the distribution matching reward.*

Proof. Let $\rho_{\pi_{E_i},\pi_{-E_i}}$ be the state-action visitation distribution of the demonstration policy π_{E_i} used to provide demonstrations for agent i, which is compatible with the other demonstration distributions according to Definition 1. Let the the corresponding joint policy be $\boldsymbol{\pi}_E$. Consider the distribution matching error for agent i, for example the Wasserstein distance $W(\rho_{\pi_i,\pi_{-i}}, \rho_{\pi_{E_i},\pi_{-E_i}})$.

We know there exists a joint imitation policy for which the cumulative distribution matching error $\sum_i (W(\rho_{\pi_i,\pi_{-i}}, \rho_{\pi_{E_i},\pi_{-E_i}})) = 0$, which is $\boldsymbol{\pi}_E$. Assuming the other agents follow policies π_{-E_i}, $W(\rho_{\pi_i,\pi_{-E_i}}, \rho_{\pi_{E_i},\pi_{-E_i}}) \geq 0$ for all $\pi_i \neq \pi_{E_i}$. Therefore, the joint policy $\boldsymbol{\pi}_E$ is a Nash equilibrium with respect to distribution matching reward for agent i. We can show the same for all $i \in [1, N]$. □

Next, we highlight the key observation that because the state-action visitation distributions represented by the demonstrations are compatible, steps each agent takes to reduce its individual distribution mismatch cost push the joint expected distribution mismatch down for all agents, meaning that learning converges to the desired joint policy.

Claim 2. *Let each agent simultaneously and independently minimize the distribution mismatch to the state-action visitation distribution represented by the corresponding demonstrations. Then each agent's policy will converge to the corresponding demonstration policy, and the set of demonstration policies will constitute a Nash equilibrium for the agent policies.*

Proof. Let us define $\boldsymbol{\pi}^t$ the resultant joint policy of N agents performing distribution matching for t steps, $\boldsymbol{\pi}^t(s) = \langle \pi_1^t, \ldots, \pi_N^t \rangle$. Based on Ratliff et al. [27], Ross et al. [29], we know that at each learning step, all agents get closer in expectation to their corresponding visitation distributions, i.e.,

$$W(\rho_{\pi_i^t,\pi_{-i}^t}, \rho_{\pi_{E_i},\pi_{-E_i}}) \geq W(\rho_{\pi_i^{t+1},\pi_{-i}^{t+1}}, \rho_{\pi_{E_i},\pi_{-E_i}}).$$

Because of this expected reduction in distribution mismatch with each learning step, the distribution matching error gets strictly smaller in expectation as the agents learn, implying they will eventually converge to the joint expert policy $\boldsymbol{\pi}_E$ as desired. $\boldsymbol{\pi}_E$ is a Nash equilibrium for the distribution matching problems, as indicated by Claim 1. □

As stated earlier, one way to obtain compatible demonstrations is to sample them from demonstrators that have been trained together to perform some task T — where the demonstrators do not necessarily maximize the task reward R_T. In imitation learning, it is typically not necessary for the agents to know what the demonstrators' task reward is. However, suppose that the agents have access to both R_T and demonstrations from experts at task R_T, meaning their policies maximize the return for that task.

Let $R_{I,i}$ be the distribution matching reward, such that an agent maximizing $R_{I,i}$ will minimize the distribution mismatch to expert i's state-action visitation distribution, where the expert policies maximize R_T. Note that expert policies that maximize R_T are in a Nash equilibrium with respect to R_T. Claim 3 states that if the agents are trained to maximize a reward function that is a linear combination of the task reward R_T and $R_{I,i}$, then the converged agent policies should also be in a Nash equilibrium with respect to R_T. The proof of Claim 3 relies on Claim 2, which shows that the expert policies constitute a Nash equilibrium with respect to the distribution matching reward.

Claim 3. *Let R_T be the reward function used to train the expert policies $\boldsymbol{\pi}_E$, and let the expert policies have converged with respect to R_T (i.e., they are in a Nash equilibrium with respect to reward R_T). Let $R_{I,i} = -W(\rho_{\pi_i,\pi_{-i}}, \rho_{\pi_{E_i},\pi_{-E_i}})$. Then $\boldsymbol{\pi}_E$ are a Nash equilibrium for reward functions of the form, $\alpha R_T + \beta R_{I,i}$, for any $\alpha, \beta > 0$.*

Proof. Let $R_{c,i} = \alpha R_T + \beta R_{I,i}$. The following reasoning is on a per-agent basis, so we drop the i from $R_{c,i}$ and $R_{I,i}$ for convenience. For π_{E_i} to not be a Nash equilibrium with respect to R_c there needs to exist a policy $\tilde{\pi}_i$ such that

$$\mathbb{E}[R_c(\tilde{\pi}_i(s)|\pi_{E_{-i}})] > \mathbb{E}[R_c(\pi_{E_i}(s)|\pi_{E_{-i}})].$$

That implies

$$\begin{aligned}&\alpha\mathbb{E}[R_T(\tilde{\pi}_i(S)|\pi_{E_{-i}})] + \beta\mathbb{E}[R_I(\tilde{\pi}_i(S)|\pi_{E_{-i}})] \\ &> \alpha\mathbb{E}[R_T(\pi_{E_i}(s)|\pi_{E_{-i}})] + \beta\mathbb{E}[R_I(\pi_{E_i}(s)|\pi_{E_{-i}})].\end{aligned}$$

But by definition, for all $\pi_{E_i}(s)$,

$$\mathbb{E}[R_T(\pi_{E_i}(s)|\pi_{E_{-i}})] \geq \mathbb{E}[R_T(\tilde{\pi}_i(S)|\pi_{E_{-i}})]$$

and

$$\mathbb{E}[R_I(\pi_{E_i}(s)|\pi_{E_{-i}})] \geq \mathbb{E}[R_I(\tilde{\pi}_i(S)|\pi_{E_{-i}})],$$

which is a contradiction. □

In this section we focus on the process of N agents jointly imitating their respective demonstrations. However, it is important to note that this imitation is ultimately meant to help these agents achieve some other goal.

In Claims 1 and 2 we do not assume the provided demonstrations are generated by demonstrators that maximize the reward of some task. However, Claim 3 implies that *if* they *are* experts maximizing the reward of a desired task, then not only is the imitation process going to converge to the desired policies, but also the task reward and distribution matching reward can be combined to optimize the same task — as we do when we empirically evaluate our approach. This possibility is particularly useful in cases in which compatible demonstrations are available, but not the policies which generated them (as, for example, in cases requiring expert demonstrations from teams of humans).

5 BALANCING DISTRIBUTION MATCHING WITH THE TASK REWARD

The algorithm we propose is inspired by the theoretical analysis in Section 4, and balances the individual objective of distribution matching with the shared task. To do so, the agents are provided a mixed reward: part cost function for minimizing individual distribution mismatch, part environment reward. This approach has

been shown to be effective in balancing individual preferences with shared objectives in multi-agent RL [5, 6]. The individual agent policies are learned by independently updating each agent's policy using an on-policy RL algorithm of choice.

The demonstrations used as targets for the distribution matching are the state-only trajectories generated by agents trained on the same task of interest. Using the state-only demonstrations has been shown to be effective when imitating based on observations alone [39], and the experiments also show its effectiveness in this setting. These "expert" policies can show an intermediate competency in the task at hand and the sampled demonstrations do not need to be the same for all the agents, which is verified in Section 6.

This learning scheme for training individual agents is summarized by Algorithm 1 and Figure 1.

6 EXPERIMENTAL EVALUATION

This section performs two main experiments. The first experiment evaluates whether our method may improve coordination — and therefore learning efficiency— over a decentralized MARL baseline. A comparison against a CTDE algorithm is also performed. The second experiment is an ablation study on the demonstrations that are provided to our algorithm, to investigate the sense in which the expert demonstrations should be coordinated.

6.1 Environments

Experiments were conducted on the StarCraft Multi-Agent Challenge domain. StarCraft features cooperative tasks where a team of controllable "allied" agents must defeat a team of enemy agents. The enemy agents are controlled by a fixed AI. The battle is won and the episode terminates if the allies can defeat all enemy agents. The allies receive a team reward every time an enemy agent is killed, and when the battle is won. In all experiments, each allied agent directly receives the team reward. StarCraft is a partially observable domain, where an allied agent can observe features about itself, as well as allies and enemies within a fixed radius. The code is provided at https://github.com/xxxxxx/adaptive_marl.

The specific StarCraft tasks used here are described below.

- 5m vs 6m (5v6): The allied team and enemy team consist of 5 Marines and 6 Marines respectively.
- 3s vs 4z (3sv4z): The allied team and enemy team consist of 3 Stalkers and 4 Zealots respectively.

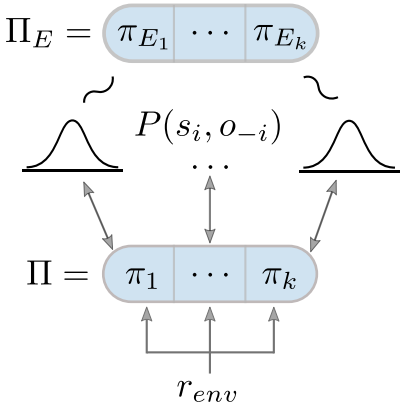

Figure 1: Demonstrations are sampled from a joint expert policy. Agents individually match the visitation distributions implied by the corresponding demonstration. The demonstrations consist of the expert agent's own state and observations of other expert agents. Additionally, agents are also provided with the shared task reward.

Algorithm 1: Distributed MARL with distribution matching

Input: Number of agents K,
expert demonstrations $\mathcal{D}_0, \ldots, \mathcal{D}_K$,
environment *env*, number of epochs N,
number of time-steps per epoch M,
reward mixture coefficient c

for $k = 0, \ldots, K-1$ **do**
  Initialize discriminator parameters ϕ_k;
  Initialize policy parameters θ_k;
end
for $n = 0, 1, \ldots, N-1$ **do**
  Gather $m = 1, \ldots, M$ steps of data $(s^m, \boldsymbol{a}^m, r^m_{env})$ from *env*;
  for $k = 0, \ldots, K-1$ **do**
    Sample M states from demonstration $\mathcal{D}_k$;
    Update discriminator D^k_ϕ;
    Get GAIL reward $r^m_{k,GAIL} = D_{k,\phi}(s^m)$ for $m = 1, \ldots, M$;
    set agent reward $r^m_{k,mix} = r^m_{env} + r^m_{k,GAIL} * c$;
    Update agent policy π^k_θ with data $(s_m, \boldsymbol{a}_m, r^m_{k,mix})$ for $m = 1, \ldots, M$;
  end
end
Output: K agent policies $\boldsymbol{\pi}_\theta$

6.2 Baselines

Our method is compared against a naive decentralized MARL algorithm, independent PPO [32] (IPPO), where individual PPO agents directly receive the team environment reward. Although agents trained under the IPPO scheme cannot share information and see only local observations, prior work has shown that IPPO can be surprisingly competitive with CTDE methods [43]. We also compare against a widely used CTDE method, QMIX [26]. Since agents trained with QMIX have the advantage of a shared critic network that receives the global state during training, the performance of QMIX is expected to be better than that of decentralized methods with no communication.

6.3 Setup

Our algorithm uses the same IPPO implementation as the baseline, with the addition of a GAIL discriminator for each independent agent i to generate an imitation reward signal, $r_{i,GAIL}$. The scaled

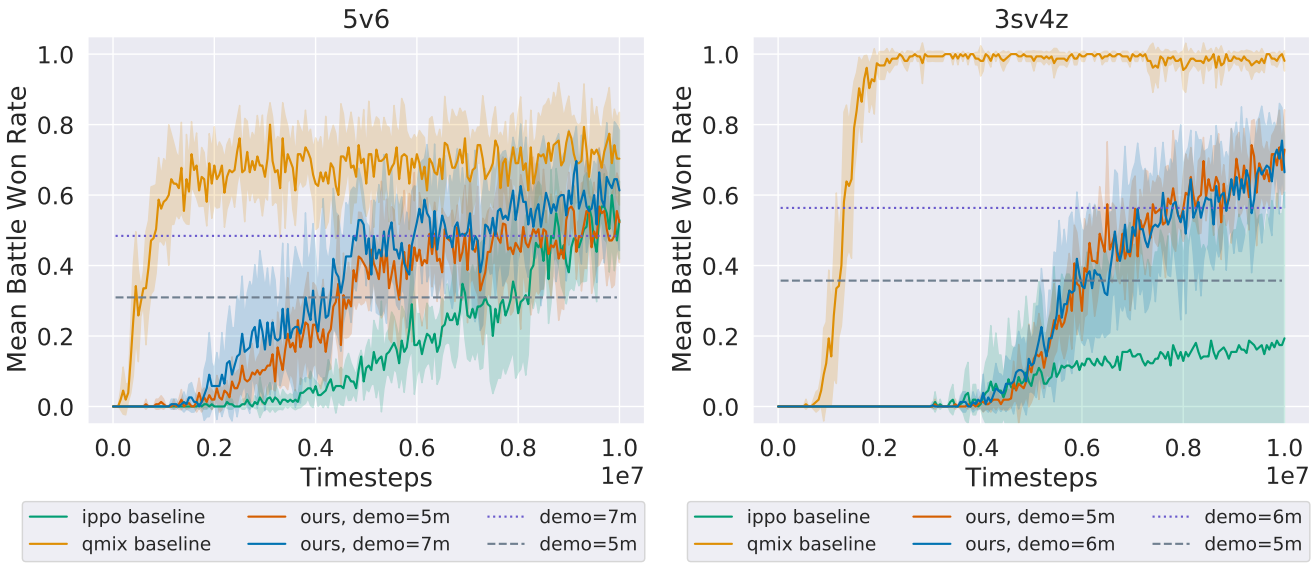

Figure 2: Learning curves of our algorithm, trained with two demonstration qualities, compared to IPPO and QMIX baselines on the 5v6 task (left) and the 3sv5z task (right). Each curve is the mean of 5 runs with independent seeds. The shaded area indicates the standard error for each curve. The win rates achieved by the demonstration policies are plotted as horizontal lines.

GAIL reward is added to the environment reward r_{env}, with scaling coefficient $c \in \mathbb{R}$:

$$r_{i,mix} = r_{env} + r_{i,GAIL} * c \tag{3}$$

The data for the GAIL discriminator consists of 1000 joint observation-only trajectories (no actions). The data is sampled from checkpoints during training runs of baseline IPPO with the environment reward. In runs of our algorithm, each agent imitates the marginal observations of the corresponding agent from the dataset (i.e., agent i will imitate agent i's observations from the dataset). Since the allied agent teams in our experimental domains are homogeneous [†], the precise mapping of agents to demonstration trajectories does not matter—there simply needs to be a mapping and it should remain fixed during training. For each task, we train our method with demonstrations sampled from two joint expert policies that achieve approximately 30% and 50% win rates respectively. The win rates achieved by the demonstration policies are plotted on the graphs.

6.4 Main Results

We compare our method, which learns with a mixed imitation/task reward, to baselines that are trained on the task reward only. All algorithms are evaluated for 32 test episodes at regular intervals during training, and trained for 10 million time steps. The evaluation metric is the mean rate of battles won against enemy teams during test episodes. To evaluate our proposed method's sensitivity to demonstration quality, the method was trained with two sets of demonstrations that have differing win rates.

[†]Homogeneous in the sense that all allied agents have the same state and action space.

Figure 2 shows that in both 5v6 and 3sv4z, our method significantly improves learning speed over IPPO (the decentralized baseline). QMIX (the CTDE baseline) learns faster than our method and IPPO on both tasks, illustrating the challenging nature of the decentralized cooperation problem. However, on 5v6, all three methods converge to a similar win rate at the end point of training. It is possible that given enough training time, our method and IPPO could converge to the QMIX win rate on 3sv4z as well. For both demonstration qualities, our method surpasses the win rate of the expert joint policies. Despite a win rate difference between the demonstrations of approximately 20% in both tasks, our method performs similarly. This relative invariance to demonstration quality suggests that the demonstrations provide a useful cooperative signal that enable the agents to coordinate and thus discover behaviors that aggregate more rewards than portrayed in the demonstrations themselves.

6.5 Ablation Study

In the main results section, we provide our method with demonstrations that satisfy two coordination conditions: that they are co-trained, and that the demonstrations are collected concurrently. In this section, we perform an ablation study on these two coordination conditions to investigate which contributes more to the performance of our method.

First, the demonstrations were sampled from co-trained expert policies—this decision was motivated by the theoretical arguments in Section 4. Second, the demonstrations for each agent were concurrently sampled. As the experiments are in the partially observable setting, agent states contain observations of the other agents in the

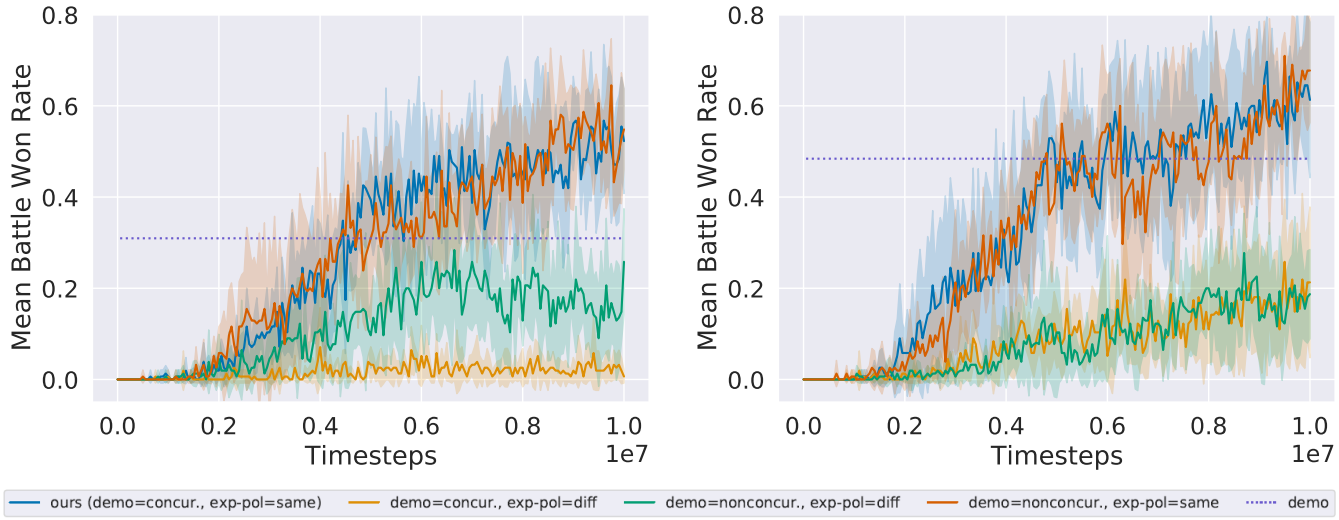

Figure 3: Ablations for IPPO trained with r_{mix} on the 5v6 task. The case where the demonstrations are concurrently sampled from co-trained expert policies corresponds to our method. Each curve is the mean of 5 runs executed with independent random seeds. The shaded portions are standard error. The win rates achieved by demonstrations are plotted as horizontal lines. Left: Experiments performed with the lower quality demonstration. Right: Experiments performed with the higher quality demonstration.

environment. This implies that the state distribution matching reward will consider how well agents can match their observations of the other agents in addition to matching their own state. Therefore, it might be beneficial for coordination if the expert demonstrations were concurrently sampled by executing the expert policies in the same environment at the same time.

We empirically test these hypotheses by applying our method to demonstrations that vary in two dimensions: (1) whether the demonstrations are sampled from co-trained expert agents, and (2) whether the demonstrations were concurrently sampled. This leads to four possible styles of demonstrations. For co-trained agents with demonstrations sampled non-concurrently, the demonstrations may be sampled from co-trained expert policies, but each agent's demonstrations originate from disjoint episodes. However, for agents that were not trained together but whose demonstrations are sampled concurrently, demonstrations could be obtained from expert policies that were each trained in separate teams[‡], but executed together in the same environment.

The study is performed on the 5v6 task, with the same hyperparameters used in the experiments of the previous section. Figure 3 shows the learning curves of the four combinations. The axis that appears to make the greatest difference in learning is whether the demonstrations originate from expert policies that were co-trained. However, whether the agent demonstrations were concurrently sampled does not appear to significantly impact learning. A possible explanation for this phenomenon is that GAIL matches the state distribution of the expert demonstrations. Although the non-concurrently sampled demonstrations do not reflect the same underlying joint trajectories, they do reflect the same distributions. We observe similar trends when our method is trained with the lower quality demonstration (Figure 3, left).

Thus, the study validates the hypothesis that demonstrations from co-trained experts are necessary for the learning benefits observed by our method over baseline IPPO.

7 DISCUSSION

This paper presents an avenue for distributed multi-agent training without communication or explicit coordination mechanisms. Fully distributed MARL is difficult, since simultaneous updates to different agents' policies can cause them to diverge. This paper studies a possible way to enable distributed MARL for cooperative tasks, by having each agent attempt to match a target state visitation distribution, in addition to maximizing the return on their shared task.

In our theoretical analysis, we show that if the target distributions are of demonstrations from expert policies trained together, then the agents should converge to the expert policies even if they are learning independently. Our experiments verify that mixing the rewards for distribution matching with the task reward does indeed accelerate cooperative task learning, compared to learning without the distribution matching objective. The ablation experiments further show that expert demonstrations should be from policies that were trained together, but do not have to be concurrently sampled.

This work is a meaningful step towards fully distributed multi-agent learning via distribution matching. However, there is much that remains to be studied to achieve this goal in full. Future work should, for instance, consider whether demonstrations sampled from expert policies with other properties, such as those trained

[‡]To ensure that each expert policy is of similar quality — despite not being trained together — the joint expert policies are trained with different seeds of the same algorithm.

with reward signals corresponding to different tasks, could be beneficial for distributed learning. It is also necessary to thoroughly analyze the relative robustness of our proposed approach and how sensitive it is to things such as demonstration quality, levels of expert compatibility, and the presence of non-imitating agents. Finally, the method proposed in this paper could be leveraged to combine human demonstrations with a task reward for applications of MARL ranging from expert decision making (similar to that done by [11] in the context of medical recommendation) or in the context of complex multi-agent traffic navigation [4]. Another potential path forward would be considering human in the loop settings such as the TAMER architecture [16], but in a fully distributed multi-agent setting.

REFERENCES

[1] Michael Bain and Claude Sammut. 1995. A Framework for Behavioural Cloning.. In *Machine Intelligence 15*. 103–129.

[2] Paul Bakker and Yasuo Kuniyoshi. 1996. Robot see, robot do: An overview of robot imitation. In *AISB96 Workshop on Learning in Robots and Animals*. 3–11.

[3] Samuel Barrett and Peter Stone. 2012. An analysis framework for ad hoc teamwork tasks. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. 357–364.

[4] Feryal Behbahani, Kyriacos Shiarlis, Xi Chen, Vitaly Kurin, Sudhanshu Kasewa, Ciprian Stirbu, Joao Gomes, Supratik Paul, Frans A Oliehoek, Joao Messias, et al. 2019. Learning from demonstration in the wild. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 775–781.

[5] Jiaxun Cui, William Macke, Harel Yedidsion, Aastha Goyal, Daniel Urielli, and Peter Stone. 2021. Scalable Multiagent Driving Policies For Reducing Traffic Congestion. *arXiv preprint arXiv:2103.00058* (2021).

[6] Ishan Durugkar, Elad Liebman, and Peter Stone. 2020. Balancing individual preferences and shared objectives in multiagent reinforcement learning. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI 2020)*. International Joint Conference on Artificial Intelligence.

[7] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2018. Counterfactual Multi-Agent Policy Gradients. *Proceedings of the AAAI Conference on Artificial Intelligence* 32 (April 2018). https://ojs.aaai.org/index.php/AAAI/article/view/11794

[8] Roy Gardner and Guillermo Owen. 1983. Game Theory (2nd Ed.). *J. Amer. Statist. Assoc.* 78 (1983), 502.

[9] Michael R. Genesereth, Matthew L. Ginsberg, and Jeffrey S. Rosenschein. 1986. Cooperation without Communication. In *AAAI*.

[10] Julio Godoy, Tiannan Chen, Stephen J. Guy, Ioannis Karamouzas, and Maria L. Gini. 2018. ALAN: adaptive learning for multi-agent navigation. *Autonomous Robots* 42 (2018), 1543–1562.

[11] Matthew Gombolay, Xi Jessie Yang, Bradley Hayes, Nicole Seo, Zixi Liu, Samir Wadhwania, Tania Yu, Neel Shah, Toni Golen, and Julie Shah. 2018. Robotic assistance in the coordination of patient care. *The International Journal of Robotics Research* 37, 10 (2018), 1300–1316.

[12] Xu He, Bo An, Yanghua Li, Haikai Chen, R. Wang, Xinrun Wang, Runsheng Yu, Xin Li, and Zhirong Wang. 2020. Learning to Collaborate in Multi-Module Recommendation via Multi-Agent Reinforcement Learning without Communication. *Fourteenth ACM Conference on Recommender Systems* (2020).

[13] Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. *Advances in neural information processing systems* 29 (2016), 4565–4573.

[14] Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro Ortega, Dj Strouse, Joel Z. Leibo, and Nando De Freitas. 2019. Social Influence as Intrinsic Motivation for Multi-Agent Deep Reinforcement Learning. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR. https://proceedings.mlr.press/v97/jaques19a.html

[15] Jiechuan Jiang and Zongqing Lu. 2021. Offline Decentralized Multi-Agent Reinforcement Learning. *ArXiv* abs/2108.01832 (2021).

[16] W Bradley Knox and Peter Stone. 2009. Interactively shaping agents via human reinforcement: The TAMER framework. In *Proceedings of the fifth international conference on Knowledge capture*. 9–16.

[17] Thomas Léauté and Boi Faltings. 2013. Protecting privacy through distributed computation in multi-agent decision making. *Journal of Artificial Intelligence Research* 47 (2013), 649–695.

[18] Hepeng Li and Haibo He. 2020. Multi-Agent Trust Region Policy Optimization. *CoRR* abs/2010.07916 (2020). https://arxiv.org/abs/2010.07916

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

[20] Bo Liu, Qiang Liu, Peter Stone, Animesh Garg, Yuke Zhu, and Animashree Anandkumar. 2021. Coach-Player Multi-Agent Reinforcement Learning for Dynamic Team Composition. In *International Conference on Machine Learning*.

[21] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, P. Abbeel, and Igor Mordatch. 2017. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In *NeurIPS*.

[22] Andrei Marinescu, Ivana Dusparic, and Siobhán Clarke. 2017. Prediction-based multi-agent reinforcement learning in inherently non-stationary environments. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 12, 2 (2017), 1–23.

[23] Andrew Y Ng, Stuart J Russell, et al. 2000. Algorithms for inverse reinforcement learning.. In *Icml*, Vol. 1. 663–670.

[24] Frans A. Oliehoek. 2012. *Decentralized POMDPs*. Springer Berlin Heidelberg, Berlin, Heidelberg, 471–503. https://doi.org/10.1007/978-3-642-27645-3_15

[25] Gabriel Peyré and Marco Cuturi. 2019. Computational Optimal Transport. *Foundations and Trends in Machine Learning* 11, 5-6 (2019), 355–607. http://arxiv.org/abs/1803.00567

[26] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2018. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*. PMLR.

[27] Nathan D Ratliff, David Silver, and J Andrew Bagnell. 2009. Learning to search: Functional gradient techniques for imitation learning. *Autonomous Robots* 27, 1 (2009), 25–53.

[28] Jeffrey S. Rosenschein and John S. Breese. 1989. Communication-Free Interactions among Rational Agents: A Probabilistic Approach. In *Distributed Artificial Intelligence*.

[29] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. 627–635.

[30] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philiph H. S. Torr, Jakob Foerster, and Shimon Whiteson. 2019. The StarCraft Multi-Agent Challenge. *CoRR* abs/1902.04043.

[31] Stefan Schaal. 1997. Learning from demonstration. In *Advances in neural information processing systems*. 1040–1046.

[32] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[33] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. 2018. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science* 362, 6419 (2018), 1140–1144.

[34] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. 2017. Mastering the game of go without human knowledge. *nature* 550, 7676 (2017), 354–359.

[35] Jiaming Song, Hongyu Ren, Dorsa Sadigh, and Stefano Ermon. 2018. Multi-Agent Generative Adversarial Imitation Learning. In *Advances in Neural Information Processing Systems*, Vol. 31.

[36] Peter Stone, Richard S Sutton, and Gregory Kuhlmann. 2005. Reinforcement learning for robocup soccer keepaway. *Adaptive Behavior* 13, 3 (2005), 165–188.

[37] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. 2018. Value-Decomposition Networks For Cooperative Multi-Agent Learning Based On Team Reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems* (Stockholm, Sweden) *(AAMAS '18)*. International Foundation for Autonomous Agents and Multiagent Systems.

[38] Sebastian Thrun. 1998. Lifelong learning algorithms. In *Learning to learn*. Springer, 181–209.

[39] Faraz Torabi, Garrett Warnell, and Peter Stone. 2019. Generative Adversarial Imitation from Observation. *arXiv:1807.06158 [cs, stat]* (June 2019). http://arxiv.org/abs/1807.06158 arXiv: 1807.06158.

[40] Cédric Villani. 2008. *Optimal transport: old and new*. Vol. 338. Springer Science & Business Media.

[41] Hongwei Wang, Lantao Yu, Zhangjie Cao, and Stefano Ermon. 2021. Multi-agent Imitation Learning with Copulas. In *Machine Learning and Knowledge Discovery in Databases. Research Track*. 139–156.

[42] Ying Wen, Hui Chen, Yaodong Yang, Zheng Tian, Minne Li, Xu Chen, and Jun Wang. 2021. Multi-Agent Trust Region Learning. https://openreview.net/forum?id=eHG7asK_v-k

[43] Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre Bayen, and Yi Wu. 2021. The Surprising Effectiveness of MAPPO in Cooperative Multi-Agent Games. arXiv:2103.01955 [cs.LG]

8 EXPERIMENT DETAILS

The experimental code is based on the multi-agent PPO implementation provided by [43] and the PyMARL code base [30]. All MARL implementations in this paper have fully separate policy/critic networks and optimizers per agent.

For all IPPO agents, the policy architecture is two fully connected layers, followed by an RNN (GRU) layer. Each layer has 64 neurons with ReLU activation units. For QMIX agents, the policy architecture is the same except there is only a single fully connected layer before the RNN layer [§]. We attempted running QMIX with the the IPPO agent architecture, but found that the performance of QMIX significantly suffered (Figure 4 on 5v6. Thus, for the QMIX experiments in the main body of the paper, the better-performing policy architecture was applied.

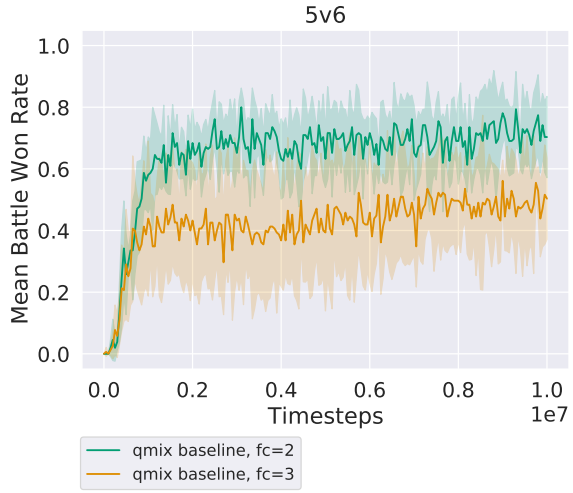

Figure 4: QMIX is sensitive to the agent policy architecture. Performance on the 5v6 task suffers significantly when an extra fully connected layer is added.

The critic architecture is the same as the policy architecture. The discriminator architecture consists of two fully connected layers with tanh activation functions.

9 HYPERPARAMETERS

For QMIX, the default parameters specified in Rashid et al. [26] are used for both tasks. For IPPO, and the IPPO component of our method, mostly default parameters (as specified in [26, 43]) were used. The hyperparameters that varied between tasks or were tuned are provided in Table 1. The remaining hyperparameters may be viewed at the GitHub repository.

We conducted a hyperparameter search over the following GAIL parameters: the GAIL reward coefficient, the number of epochs that the discriminator was trained for each IPPO update, the buffer size, and the batch size. The final selected values are given in Table 2.

[§]This is the architecture used in Rashid et al. [26]

	5v6	**3sv4z**
epochs	10	15
buffer size	1024	1024
gain	0.01	0.01
clip	0.05	0.2

Table 1: IPPO Hyperparameters.

	5v6	**3sv4z**
gail rew coef	0.3	0.05
discr epochs	120	120
buffer size	1024	1024
batch size	64	64
n exp eps	1000	1000

Table 2: GAIL Hyperparameters.