DM^2: Distributed Multi-Agent Reinforcement Learning via Distribution Matching

Caroline Wang [*]
Department of Computer Science
The University of Texas at Austin
caroline.l.wang@utexas.edu

Ishan Durugkar [*]
Department of Computer Science
The University of Texas at Austin
ishand@cs.utexas.edu

Elad Liebman [*]
SparkCognition Research
eliebman@sparkcognition.com

Peter Stone
Department of Computer Science
The University of Texas at Austin
and Sony AI
pstone@cs.utexas.edu

Abstract

Current approaches to multi-agent cooperation rely heavily on centralized mechanisms or explicit communication protocols to ensure convergence. This paper studies the problem of distributed multi-agent learning without resorting to explicit coordination schemes. The proposed algorithm (DM^2) leverages distribution matching to facilitate independent agents' coordination. Each individual agent matches a target distribution of concurrently sampled trajectories from a joint expert policy. The theoretical analysis shows that under some conditions, if each agent optimizes their individual distribution matching objective, the agents increase a lower bound on the objective of matching the joint expert policy, allowing convergence to the joint expert policy. Further, if the distribution matching objective is aligned with a joint task, a combination of environment reward and distribution matching reward leads to the same equilibrium. 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. Additional experiments probe the conditions under which expert demonstrations need to be sampled in order to outperform the fully distributed baseline.

1 Introduction

Multi-agent reinforcement learning (MARL) [20] 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 [37], two-player games [34, 35], and ad-hoc teamwork tasks [3]. Training multiple agents at once is 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, be trained together but then deployed individually [27, 7], or be coordinated through some form of communication. [22, 15, 21].

Fully distributed training of agent policies remains an open problem in MARL. Distributed, or decentralized training is desirable in various settings, such as those with a large number of agents, where parallelism, robustness, or scalability is needed, where agents are faced with changing environments [23], where agents learn in a lifelong manner [39], or where ensuring privacy is a concern [18].

[*] Equal contribution.

Preprint. Under review.

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. For example, demonstrations of expert football/soccer players could be useful when training robot players [37]. Another natural example for matching demonstrations is that of human medical teams trained to accomplish difficult, specialized tasks.

The theoretical analysis shows that agents independently, turn-by-turn updating their policy to improve the visitation distribution matching to that of their corresponding expert demonstrations leads to convergence to the joint expert policy under conditions on the demonstration policies and each agent's improvement of its individual imitation learning objective. The result is based on showing that each agent improving its private objective increases a lower bound on the objective of matching the joint expert policy. The analysis further shows that when expert policies are aligned with a given task, distribution matching can be combined with the task reward without perturbing the Nash equilibrium.

The paper then proposes the DM^2 algorithm (**d**istributed **m**ulti-agent RL via **d**istribution **m**atching), which 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 a task reward. Experimental evaluation in the StarCraft domain shows that this approach accelerates learning compared to 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 together. However, demonstrations from policies that were not trained together do not assist learning in a similar manner.

The contributions of this paper are as follows:

- It proposes the use of distribution matching for coordination in fully decentralized MARL.
- It shows the convergence of decentralized distribution matching to the joint expert policy.
- It empirically validates that if demonstrations are aligned with the shared objective, then combining the task reward with an imitation reward leads to improved performance over distributed learning with the task reward only.

2 Related Work

Cooperation in the Decentralized Setting. Many algorithms for multi-agent cooperation tasks require some degree of information sharing between agents. Centralized training decentralized execution (CTDE) methods use a single centralized critic that aggregates information during training, but is no longer required at execution time [22, 38, 27, 7, 43]. In practical implementations of CTDE methods, agent networks often share parameters during training as well.

Rather than sharing model components, methods may also explicitly communicate information between agents. Agents may be allowed to directly communicate information to each other [15, 19]. There might also be a central network that provides coordinating signals to all agents [13, 21]. Knowledge of other agents' policies during training may also be assumed to limit the deviation of the joint policy [42].

This work studies the fully decentralized setting without communication or shared model components. To our knowledge, relatively few works consider this setting. Early work analyzed simple cases where two agents with similar but distinct goals could cooperate for mutual benefit under a rationality assumption [28, 9]. More recently, in the ALAN system for multi-agent navigation [10], agents learn via a multi-armed bandits method that does not require any communication. Jiang and Lu [16] 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 [14] originally proposed adversarial distribution matching as a way to perform imitation learning in the single agent setting (the GAIL algorithm). Song et al. [36] extend GAIL to the multi-agent setting in certain respects. Their analysis sets up independent imitation learning as searching for a Nash equilibrium, and assumes that a unique equilibrium exists. Their experiments focus on training the agent policies in the CTDE paradigm, rather than the fully distributed setting. This work instead leverages recent single-agent GAIL convergence theory [12] to demonstrate convergence to the joint expert policy, and performs experiments with distributed learning. 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] and Radke et al. [26] show that balancing individual preferences (such as matching the state-action visitation distribution of some strategies) with the shared task reward can accelerate progress on the shared task. In contrast to these 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, imitation learning, and distribution matching.

Markov games. A Markov game [20] 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 $\rho_0 : \mathcal{S} \mapsto \Delta(\mathcal{S})$, 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 parameterized policy $\pi_{\theta_i} : \mathcal{S} \mapsto \Delta(\mathcal{A}_i)$, and the joint policy $\boldsymbol{\pi}_\theta = [\pi_{\theta_1}, \cdots, \pi_{\theta_K}]$ is the product of the individual agent policies. Occasionally, the policy parameters θ are omitted for convenience. Note that each agent observes the full state. We use subscript $i-$ to refer to all agents except i, i.e., π_{i-} refers 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 expected 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 task 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 $\pi_i^* = \operatorname{argmax}_{\pi_i} \mathbb{E}_{\pi_i, \pi_{i-}}[\sum_{t=0}^{\infty} \gamma^t r_{i,t}]$. 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, and assumes a discrete and finite state and action space. However, the experiments are conducted in partially observable MDPs (POMDPs) with continuous states, which can be formalized as Dec-POMDPs in the multi-agent setting [25]. Dec-POMDPs include two additional elements: the set of observations Ω and each agent's observation function $O_i : \mathcal{S} \longmapsto \Delta(\Omega)$.

Imitation learning and distribution matching. Imitation learning [2, 29, 31] is a problem setting where an agent tries to mimic trajectories $\{\xi_0, \xi_1, \ldots\}$ where each trajectory $\xi = \{(s_0, a_0), (s_1, a_1), \ldots\}$ is demonstrated by an expert policy π_E. Various methods have been proposed to address the imitation learning problem. Behavior cloning [1] applies supervised learning to expert demonstrations to recover the maximum likelihood policy. Inverse reinforcement learning (IRL) [24] recovers a reward function which can then be used to learn the expert policy using reinforcement learning. To do so, $\texttt{IRL}(\pi_E)$ aims to recover a reward function under which the trajectories demonstrated by π_E are optimal.

Ho and Ermon [14] formulate imitation learning as a distribution matching problem and propose the GAIL algorithm. Let the state-action visitation distribution of a joint policy $\boldsymbol{\pi} = \langle \pi_1, \ldots \pi_K \rangle$ be:

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

In a multi-agent setting, 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-}. Ho and Ermon [14] 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 $\mathtt{RL} \circ \mathtt{IRL}(\pi_E)$ problem.

Guan et al. [12] showed that in the single-agent case, the GAIL algorithm converges to the expert policy under a variety of policy gradient techniques, including TRPO [32]. Let r_ϕ be a reward function (discriminator) parameterized by ϕ, and let $\psi(\phi)$ be a convex regularizer. Guan et al. [12] formulate the GAIL problem as the following min-max problem:

$$\begin{aligned}&\min_{\theta}\max_{\phi}\mathcal{L}(\theta,\phi) \\ &\qquad\text{s.t. } \mathcal{L}(\theta,\phi) := V(\pi_E, r_\phi) - V(\pi_\theta, r_\phi) - \psi(\phi)\end{aligned} \tag{1}$$

where $V(\pi, r) = \mathbb{E}_{s_0\sim\rho_0}\mathbb{E}_{\pi}[\sum_{t=0}^{\infty}\gamma^t r_{i,t}]$ is the expected return from some start state when following policy π and using reward function r.

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

4 Theoretical Analysis

This section provides theoretical grounding for the core proposition of this paper. The analysis shows that if K agents independently and turn-by-turn minimize the distribution mismatch to their respective demonstrations, then under the conditions stated below, agent policies will converge to the joint expert policy. Next, it establishes that if the agents are learning to maximize the mixture of an extrinsic task reward and a distribution matching reward, then the agent policies will converge to a Nash equilibrium with respect to the joint reward.

The setting considered assumes that there exists a set of demonstrations generated by a joint expert policy that has been trained to perform a task [2] Further, for every policy considered, there is a minimal probability of visiting each state, and each agent learns via a single-agent imitation learning algorithm such that it improves its distribution matching reward at each step.

4.1 Convergence of Independent GAIL Learners

This analysis considers the setting where each agent i performs independent learning updates according to the GAIL algorithm, in order to imitate the ith expert. The analysis proposes a condition on individual agents' GAIL objective improvement, and shows that if this condition is satisfied, a lower bound on a joint imitation learning objective is improved. Further, the lower bound objective converges, demonstrating the convergence of independent GAIL.

Let the parameterized reward (discriminator) of the ith agent be $r_{\phi_i}: \mathcal{S}\times\mathcal{A}_i \to \mathbb{R}$, for $\phi_i \in \Phi$. At each agent's update, all the other agent policies are held fixed, and the corresponding discriminator has converged to $r^{opt}_{\phi_i}$, where $\phi^{opt}_i \in \operatorname{argmax}_{\phi_i\in\Phi}\mathcal{L}(\theta_i,\phi_i|\theta_{i-})$. Guan et al. [12] showed that the learning process of a single agent repeatedly updating ϕ_i converges to ϕ^{opt}_i. The update scheme we consider for theoretical purposes is specified in Algorithm 2 (located in the Appendix).

Define the per-agent GAIL loss as follows:

$$\begin{aligned}\mathcal{L}(\theta_i,\phi_i|\theta_{i-}) &:= V(\pi_{E_i}, r^{opt}_{\phi_i}|\pi_{E_{i-}}) - V(\pi_{\theta_i}, r^{opt}_{\phi_i}|\pi_{\theta_{i-}}) - \psi(\phi_i) \\ \text{s.t. } V(\pi_i, r^{opt}_{\phi_i}|\pi_{i-}) &:= \frac{1}{1-\gamma}\mathop{\mathbb{E}}_{s\sim\rho_{\boldsymbol{\pi}},a_i\sim\pi_i}\left[r^{opt}_{\phi_i}(s,a_i)\right],\end{aligned}$$

[2] The practical realization of this assumption is discussed in Section 5.

where $\rho_{\boldsymbol{\pi}}(s) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t P(s_t = s|\boldsymbol{\pi})$ is the discounted *state* visitation distribution.

Consider the random variable that is the indicator function $\mathbb{1}_{a_j = a_j^E}(s)$ for the event that at state s, agent j would take an action that matched expert j's action. Note that the expectation of this indicator is the probability of matching the expert's action. [3] Define the **joint action-matching objective** as the probability that agent actions match their corresponding experts (plus a constant), weighted by the probability of visiting states:

$$J(\boldsymbol{\pi}) = \sum_{s\in S}\rho_{\pi}(s)\left[(K-1) + \mathop{\mathbb{E}}_{\boldsymbol{a}\sim\boldsymbol{\pi}}[\mathbb{1}_{\bigcap_i a_i = a_i^E}(s)]\right]. \tag{2}$$

where $\mathbb{1}_{\bigcap_i a_i = a_i^E}(s)$ indicates the event that *all* the agents would take actions that match their corresponding experts. Maximizing $J(\boldsymbol{\pi})$ precisely corresponds to solving the multi-agent imitation learning problem because the joint expert policy π_E is the unique maximizer of $J(\boldsymbol{\pi})$ (see Lemma 4 in the Appendix).

Assumption 1. *Let $c \in \mathbb{R}^+$. Suppose for all agents i, the learned discriminator reward at each state is proportional to an action-matching indicator function:*

$$r_{\phi_i}(s,a) = c\mathbb{1}_{a_i = a_i^E}(s) = c(1 - \mathbb{1}_{a_i \neq a_i^E}(s)).$$

Theorem 1 (action-matching objective)**.** *The joint action-matching objective $J(\boldsymbol{\pi})$ is lower bounded by the following sum over individual action rewards r_{ϕ_i}:*

$$L(\boldsymbol{\pi}) := \sum_{s\in S}\rho(s)\left[\sum_{i=1}^{K}\frac{1}{c}\mathop{\mathbb{E}}_{a\sim\pi_i}[r_{\phi_i}(s,a)]\right]. \tag{3}$$

When an agent updates its policy to optimize its component of $L(\boldsymbol{\pi})$, the state visitation distribution ρ might change such that the expected action rewards for other agents decrease. The next corollary introduces a lower bound on $L(\boldsymbol{\pi})$ that is independent of the state visitation distribution ρ.

Corollary 1 (lower bound)**.** *Let ϵ be the minimum probability of visiting any state. For all ρ, $L(\boldsymbol{\pi})$ is lower bounded by $L_\epsilon(\boldsymbol{\pi})$:*

$$L(\boldsymbol{\pi}) > \sum_{s\in S}\sum_{i=1}^{K}\epsilon\left[\frac{1}{c}\mathop{\mathbb{E}}_{a\sim\pi_i}[r_{\phi_i}(s,a)]\right] =: L_\epsilon(\boldsymbol{\pi}). \tag{4}$$

In the Appendix, we lay out the condition that ensures each agent updating its policy leads to improvement in $L_\epsilon(\boldsymbol{\pi})$. This $L_\epsilon(\boldsymbol{\pi})$ is a lower bound on the actual objective of interest $J(\boldsymbol{\pi})$ — by Theorem 1 and Corollary 1. Further, $L_\epsilon(\boldsymbol{\pi})$ has a unique global maximizer, which is $\boldsymbol{\pi} = \boldsymbol{\pi}_E$ (Lemma 6). Thus, while the action reward for the other agents r_{ϕ_j} might decrease in the short term, the joint action matching objective across all agents will increase as the learning process continues. Since $J(\boldsymbol{\pi})$ is bounded from above (Lemma 3), this process of improving the lower bound will converge to the optimal policy for this objective, which is the joint expert policy.

4.2 Multi-agent Learning with Mixed Task and Imitation Reward

Lemma 4 in the Appendix shows that the joint expert policy globally maximizes the joint imitation learning objective. Let ϕ_i^{opt,E_i} be the optimal discriminator parameters for the ith expert, π_{E_i}. From a game theoretic perspective, this lemma implies that these expert policies are a Nash equilibrium for the imitating agents with respect to r_{ϕ_i,E_i}^{opt}.

Next, note that in imitation learning, it is typically not necessary for the agents to know what the demonstration actor's task reward is. However, suppose that the agents have access to both demonstrations from policies optimal at task T, and the corresponding reward function R_T.

Let $R_{I,i} = r_{\phi_i,E_i}^{opt}$, where the expert policies maximize R_T. The expert policies that maximize R_T are in a Nash equilibrium with respect to R_T. Theorem 2 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 are also in a Nash equilibrium with respect to R_T.

[3] For the purpose of exposition, we will assume that the expert policy is deterministic. The theory in this section can be extended to the case where $\boldsymbol{\pi}_E$ is stochastic by taking an expectation over expert actions.

Theorem 2. *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). Then $\boldsymbol{\pi}_E$ are a Nash equilibrium for reward functions of the form, $\alpha R_T + \beta R_{I,i}$, for any $\alpha, \beta > 0$.*

In Theorem 1, we do not assume the provided demonstrations are generated by demonstrators that maximize the reward of some task. However, Theorem 2 implies that if they *are* experts maximizing the reward of a desired task, then the task reward and distribution matching reward can be combined to optimize the same task — as we do when we empirically evaluate our approach. The mixed reward is particularly useful in cases where demonstrations are available, but not the policies that generated them (e.g. when demonstrations originate from teams of humans).

5 Methods

In this section we discuss practical considerations of fully distributed multi-agent imitation learning, and propose DM^2, an algorithm whose performance we analyze in Section 6.

5.1 Generating Expert Demonstrations

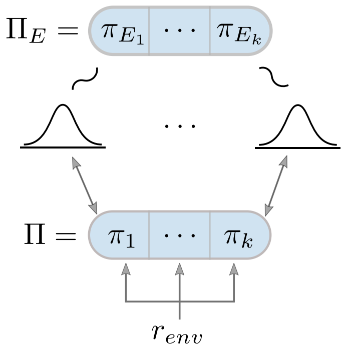

Figure 1: Demonstrations are sampled from a joint expert policy. Agents individually match the visitation distributions implied by the corresponding demonstration. Additionally, agents are also provided with the shared task reward.

Section 4 shows that agents individually following demonstrations induced from an existing joint policy can converge to said joint policy without centralized training or communication. In practice, demonstrations for individual agents can be obtained such that their visitation distributions imply an *achievable* joint expert policy. 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. Suppose there are two agents. Each agent i attempts to match a simple joint state-action distribution, consisting of the ith agent occupying a specific tile, and the other agent occupying one of the three remaining tiles. With these demonstrations, it is impossible for both agents to fully match their desired distributions.

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. This notion of compatibility is defined below.

Definition 1 (Compatible demonstrations)**.** *State-action visitation distributions $\rho_{\pi_i, \hat{\pi}_{i-}}$ from a collection of K policies $\{\pi_i\}_{i=1}^K$ (where $\hat{\pi}_{i-}$ are the other agent policies executed with π_i to obtain the state-action visitation distribution $\rho_{\pi_i, \hat{\pi}_{i-}}$) are* compatible *if for all i, $s \in \mathcal{S}, a \in \mathcal{A}$, there exists a joint policy $\boldsymbol{\pi}' = \langle \pi_1', \ldots, \pi_K' \rangle$ with the joint state-action visitation distribution $\rho_{\boldsymbol{\pi}}(s, \boldsymbol{a})$ (Equation 3) such that the marginal state-action visitation distribution for agent i is*

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

Observe that K expert policies that are trained in the same environment to perform a task induce compatible individual state-action visitation distributions, providing a practical method to obtain compatible demonstrations.

5.2 Practical Multi-Agent Distribution Matching

DM^2 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

```
Algorithm 1: DM$^2$ (Distributed MARL with distribution matching)
Input: Number of agents $K$, expert demonstrations $\mathcal{D}_0, \dots, \mathcal{D}_K$, environment $env$, number of
       epochs $N$, number of time-steps per epoch $M$, reward mixture coefficient $c$
for $k = 0, \dots, K-1$ do
    Initialize discriminator parameters $\phi_k$;
    Initialize policy parameters $\theta_k$;
end
for $n = 0, 1, \dots, N-1$ do
    Gather $m = 1, \dots, M$ steps of data $(s^m, \boldsymbol{a}^m, r^m_{env})$ from $env$;
    for $k = 0, \dots, K-1$ do
        Sample $M$ states from demonstration $\mathcal{D}_k$;
        Update discriminator $D^k_\phi$;
        Get GAIL reward $r^m_{k,GAIL} = D_{k,\phi}(s^m)$ for $m = 1, \dots, M$;
        Set agent reward $r^m_{k,mix} = r^m_{env} + r^m_{k,GAIL} * c$;
        Update agent policy $\pi^k_\theta$ with data $(s_m, \boldsymbol{a}_m, r^m_{k,mix})$ for $m = 1, \dots, M$;
    end
end
Output: $K$ agent policies $\boldsymbol{\pi}_\theta$
```

in multi-agent RL [6, 5]. The individual agent policies are learned by independently updating each agent's policy using an on-policy RL algorithm of choice.

In the experiments, 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 state-only demonstrations has been shown to be effective when imitating based on observations alone [40], and the experiments also show its effectiveness in this setting. These "expert" policies may possess intermediate competency in the task at hand and the demonstrations do not need to be jointly sampled for all the agents, which is shown in Section 6. The proposed 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 DM^2 may improve coordination—and therefore efficiency of learning—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.

Environments. Experiments were conducted on the StarCraft Multi-Agent Challenge domain [30]. StarCraft features cooperative tasks where a team of controllable "allied" agents must defeat a team of enemy agents, where 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 specific StarCraft tasks used here are:

- 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.

Baselines. DM^2 is compared against a naive decentralized MARL algorithm, independent PPO [33] (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 [27]. Since agents trained with QMIX have the

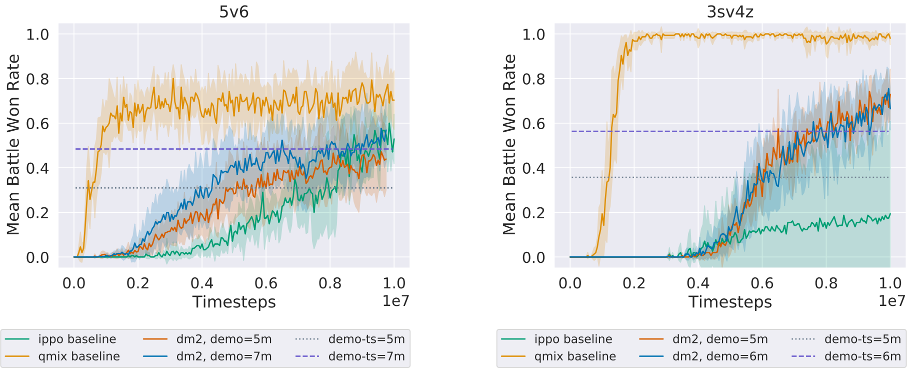

Figure 2: Learning curves of DM^2 (ours), trained with two demonstration qualities, compared to IPPO and QMIX baselines on the 5v6 task (left) and the 3sv5z task (right). The win rates achieved by the demonstration policies are plotted as horizontal lines.

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.

Setup. DM^2 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 GAIL reward is added to the environment reward r_{env}, with scaling coefficient $c \in \mathbb{R}$: $r_{i,mix} = r_{env} + r_{i,GAIL} * c$. Learning curves of all algorithms are the mean of 5 runs executed with independent random seeds, where each run is evaluated for 32 test episodes at regular intervals during training. The shaded regions on the plots show the standard error. The evaluation metric is the mean rate of battles won against enemy teams during test episodes.

The data for the GAIL discriminator consists of 1000 joint state-only trajectories (no actions). The data is sampled from checkpoints during training runs of baseline IPPO with the environment reward. In runs of DM^2, 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) [4]. For each task, demonstrations are 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.

6.1 Main Results

Figure 2 shows that in both 5v6 and 3sv4z, DM^2 significantly improves learning speed over IPPO (the decentralized baseline). QMIX (the CTDE baseline) learns faster than DM^2 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. We observe similar robustness to demonstration quality even with 10 different demonstration qualities (Figure 4 in the Appendix). The 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.

6.2 Ablation Study

In this section, we perform an ablation study on two coordination conditions satisfied by the demonstrations used in the main results section, to investigate their relative importance. First, the demonstrations were sampled from *co-trained expert policies*—this decision was motivated by the arguments in

[4] Since the allied agent teams in our experimental domains have the same state and action spaces, 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

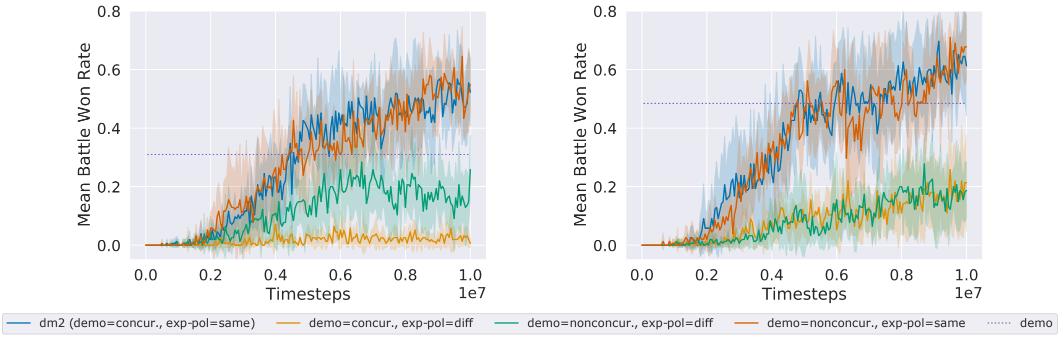

Figure 3: Ablations for IPPO trained with r_{mix} on the 5v6 task. The case where the demonstrations are concurrently sampled from co-trained expert policies corresponds to DM^2. The win rates achieved by demonstrations are plotted as horizontal lines. Left: Experiments performed with the lower quality demonstration. Right: Experiments performed with the higher quality demonstration.

Section 5. 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 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.

The experiments apply DM^2 to four possible demonstration styles that vary in the aforementioned two dimensions. A detailed explanation of how these four demonstration styles were constructed is provided in the Appendix. 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. Whether the agent demonstrations were concurrently sampled does not appear to significantly impact learning. A possible explanation 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. Similar trends are observed when DM^2 is trained with the lower quality demonstration (Figure 3, left).

7 Discussion and Future Work

This paper studies a way to enable distributed MARL for cooperative tasks without communication or explicit coordination mechanisms. Fully distributed MARL is challenging, since simultaneous updates to different agents' policies can cause them to diverge. The benefits of distributed MARL are abundant, as highlighted in the introduction. From a societal perspective, decentralized training could make agents more robust to the presence of agents they were not trained with (such as humans) and also allow more cooperation between agents that were not trained together. Decentralized training could also enable coordination while preserving the privacy of each agent. Potential ethical concerns could occur if the demonstration data isn't sufficiently anonymized. Various data sources based on human interaction could leak personal information and lead to privacy risks.

In the theoretical analysis, this paper shows that individual agents updating their policies turn-by-turn to reduce their distribution mismatch to corresponding expert distributions leads to an improvement on a lower bound to the objective of the joint agent policy matching the joint expert policy. Fully maximizing the lower bound corresponds to recovering the joint expert policy. The experiments verify that mixing the rewards for state 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 takes 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. For instance, the current theoretical analysis is limited to the tabular case. Empirical success in continuous, partially observable settings indicates that future work may be able to extend the analysis to the continuous case. Future work should also consider whether demonstrations sampled from expert policies with other properties, such as those trained with reward signals corresponding to different tasks, could be beneficial for distributed learning. The method proposed in this paper could also 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 [17], but in a fully distributed multi-agent setting.

References

[1] M. Bain and C. Sammut. A framework for behavioural cloning. In *Machine Intelligence 15*, pages 103–129, 1995.

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

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

[4] F. Behbahani, K. Shiarlis, X. Chen, V. Kurin, S. Kasewa, C. Stirbu, J. Gomes, S. Paul, F. A. Oliehoek, J. Messias, et al. Learning from demonstration in the wild. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 775–781. IEEE, 2019.

[5] J. Cui, W. Macke, H. Yedidsion, A. Goyal, D. Urielli, and P. Stone. Scalable multiagent driving policies for reducing traffic congestion. *arXiv preprint arXiv:2103.00058*, 2021.

[6] I. Durugkar, E. Liebman, and P. Stone. 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, 2020.

[7] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson. Counterfactual multi-agent policy gradients. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32, April 2018. URL `https://ojs.aaai.org/index.php/AAAI/article/view/11794`.

[8] R. Gardner and G. Owen. Game theory (2nd ed.). *Journal of the American Statistical Association*, 78:502, 1983.

[9] M. R. Genesereth, M. L. Ginsberg, and J. S. Rosenschein. Cooperation without communication. In *AAAI*, 1986.

[10] J. Godoy, T. Chen, S. J. Guy, I. Karamouzas, and M. L. Gini. Alan: adaptive learning for multi-agent navigation. *Autonomous Robots*, 42:1543–1562, 2018.

[11] M. Gombolay, X. J. Yang, B. Hayes, N. Seo, Z. Liu, S. Wadhwania, T. Yu, N. Shah, T. Golen, and J. Shah. Robotic assistance in the coordination of patient care. *The International Journal of Robotics Research*, 37(10):1300–1316, 2018.

[12] Z. Guan, T. Xu, and Y. Liang. When will generative adversarial imitation learning algorithms attain global convergence. In *AISTATS*, 2021.

[13] X. He, B. An, Y. Li, H. Chen, R. Wang, X. Wang, R. Yu, X. Li, and Z. Wang. Learning to collaborate in multi-module recommendation via multi-agent reinforcement learning without communication. *Fourteenth ACM Conference on Recommender Systems*, 2020.

[14] J. Ho and S. Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29:4565–4573, 2016.

[15] N. Jaques, A. Lazaridou, E. Hughes, C. Gulcehre, P. Ortega, D. Strouse, J. Z. Leibo, and N. De Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*. PMLR, 2019. URL `https://proceedings.mlr.press/v97/jaques19a.html`.

[16] J. Jiang and Z. Lu. Offline decentralized multi-agent reinforcement learning. *ArXiv*, abs/2108.01832, 2021.

[17] W. B. Knox and P. Stone. Interactively shaping agents via human reinforcement: The tamer framework. In *Proceedings of the fifth international conference on Knowledge capture*, pages 9–16, 2009.

[18] T. Léauté and B. Faltings. Protecting privacy through distributed computation in multi-agent decision making. *Journal of Artificial Intelligence Research*, 47:649–695, 2013.

[19] H. Li and H. He. Multi-agent trust region policy optimization. *CoRR*, abs/2010.07916, 2020. URL `https://arxiv.org/abs/2010.07916`.

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

[21] B. Liu, Q. Liu, P. Stone, A. Garg, Y. Zhu, and A. Anandkumar. Coach-player multi-agent reinforcement learning for dynamic team composition. In *International Conference on Machine Learning*, 2021.

[22] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *NeurIPS*, 2017.

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

[24] A. Y. Ng, S. J. Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 663–670, 2000.

[25] F. A. Oliehoek. *Decentralized POMDPs*, pages 471–503. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-27645-3. doi: 10.1007/978-3-642-27645-3_15. URL `https://doi.org/10.1007/978-3-642-27645-3_15`.

[26] D. Radke, K. Larson, and T. B. Brecht. Exploring the benefits of teams in multiagent learning. *ArXiv*, abs/2205.02328, 2022.

[27] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*. PMLR, 2018.

[28] J. S. Rosenschein and J. S. Breese. Communication-free interactions among rational agents: A probabilistic approach. In *Distributed Artificial Intelligence*, 1989.

[29] S. Ross, G. Gordon, and D. Bagnell. 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*, pages 627–635, 2011.

[30] M. Samvelyan, T. Rashid, C. S. de Witt, G. Farquhar, N. Nardelli, T. G. J. Rudner, C.-M. Hung, P. H. S. Torr, J. Foerster, and S. Whiteson. The StarCraft Multi-Agent Challenge. *CoRR*, abs/1902.04043, 2019.

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

[32] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

[33] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[34] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550 (7676):354–359, 2017.

[35] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

[36] J. Song, H. Ren, D. Sadigh, and S. Ermon. Multi-agent generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[37] P. Stone, R. S. Sutton, and G. Kuhlmann. Reinforcement learning for robocup soccer keepaway. *Adaptive Behavior*, 13(3):165–188, 2005.

[38] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, and T. Graepel. 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*, AAMAS '18. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

[39] S. Thrun. Lifelong learning algorithms. In *Learning to learn*, pages 181–209. Springer, 1998.

[40] F. Torabi, G. Warnell, and P. Stone. Generative Adversarial Imitation from Observation. *arXiv:1807.06158 [cs, stat]*, June 2019. URL `http://arxiv.org/abs/1807.06158`. arXiv: 1807.06158.

[41] H. Wang, L. Yu, Z. Cao, and S. Ermon. Multi-agent imitation learning with copulas. In *Machine Learning and Knowledge Discovery in Databases. Research Track*, pages 139–156, 2021.

[42] Y. Wen, H. Chen, Y. Yang, Z. Tian, M. Li, X. Chen, and J. Wang. Multi-agent trust region learning, 2021. URL `https://openreview.net/forum?id=eHG7asK_v-k`.

[43] C. Yu, A. Velu, E. Vinitsky, Y. Wang, A. Bayen, and Y. Wu. The surprising effectiveness of mappo in cooperative multi-agent games, 2021.

A Appendix

A.1 Convergence Proof Details

In Section 4, we lay out some of the conditions for our theoretical analysis. One of these conditions, that for every policy considered there is a minimal probability of visiting each state, is formalized below.

Condition 1. *Let $0 < \epsilon < 1$, and let ρ be the state visitation distribution induced by any joint policy during training. For all agents i and for all s, suppose that $\rho(s) \geq \epsilon$.*

```
Algorithm 2: Distributed MARL with distribution matching
Input: Number of agents $K$, expert demonstrations $\mathcal{D}_0, \dots, \mathcal{D}_K$, environment $env$, number of
       time-steps per epoch $M$
for $k = 0, \dots, K-1$ do
    Initialize discriminator parameters $\phi_k$;
    Initialize policy parameters $\theta_k$;
end
while any($\pi_{\theta_k}$) not converged do
    for $k = 0, \dots, K-1$ do
        Gather $m = 1, \dots, M$ steps of data $(s^m, \boldsymbol{a}^m, r^m_{env})$ from $env$;
        Update agent discriminator $r_{\phi_k}$ to maximize Equation 1 until convergence to $r^{opt}_{\phi_k}$;
        Update agent policy $\pi_{\theta_k}$ using TRPO to minimize Equation 1
    end
end
Output: $K$ agent policies $\boldsymbol{\pi}_\theta$
```

Recall that the **joint action-matching objective** is defined over the expected state visitation as the probability that all agent actions match their corresponding experts (plus a constant):

$$J(\boldsymbol{\pi}) = \sum_{s \in S} \rho_\pi(s) \left[(K-1) + \mathop{\mathbb{E}}_{\boldsymbol{a} \sim \boldsymbol{\pi}} [\mathbb{1}_{\bigcap_i a_i = a_i^E}(s)]\right].$$

where $\mathbb{1}_{\bigcap_i a_i = a_i^E}(s)$ indicates the event that *all* the agents took actions that matched their corresponding experts.

We first prove some properties of $J(\boldsymbol{\pi})$.

Lemma 3. *The objective $J(\boldsymbol{\pi})$ is bounded by*

$$(K-1) \leq J(\boldsymbol{\pi}) \leq K.$$

Proof. As shorthand, define $f(\boldsymbol{\pi}) = \sum_{s \in S} \rho_\pi(s) \mathbb{E}_{\boldsymbol{a} \sim \boldsymbol{\pi}}[\mathbb{1}_{\bigcap_i a_i = a_i^E}(s)]$. Then,

$$J(\boldsymbol{\pi}) = (K-1) + f(\boldsymbol{\pi}).$$

First, note that $f(\boldsymbol{\pi}) \geq 0$ because it is a weighted sum of expectations over indicator functions, where all weights are non-negative, and it is precisely 0 if for all states, the joint agent policy $\boldsymbol{\pi}$ does not match the correct expert actions. Thus, $J(\boldsymbol{\pi})$ is lower bounded by $(K-1)$. For the upper bound, notice that the outer summation is equivalent to the expectation under state visitation distribution ρ_π. Inside, each expectation over the indicators can be at most 1, implying that $f(\boldsymbol{\pi})$ is at most 1. Thus, $J(\boldsymbol{\pi}) \leq K$. □

Lemma 4. *Suppose the joint expert policy $\boldsymbol{\pi}_E$ is deterministic. Then $\boldsymbol{\pi}_E$ is the unique maximizer of $J(\boldsymbol{\pi})$.*

Proof. (Maximization) Since $\boldsymbol{\pi}_E$ is deterministic, by definition, each term

$$\mathop{\mathbb{E}}_{\boldsymbol{a} \sim \boldsymbol{\pi}_E} [\mathbb{1}_{\bigcap_i a_i = a_i^E}(s)] = 1.$$

Thus, $J(\boldsymbol{\pi}_E) = K$, which means that $\boldsymbol{\pi}_E$ achieves the upper bound of J.

(Uniqueness) Suppose there exists another policy $\boldsymbol{\pi} \neq \boldsymbol{\pi}_E$ that also achieves the upper bound, i.e. $J(\boldsymbol{\pi}) = K$. Let $a_i^E := \pi_{E_i}(s)$. Then there must be an agent i such that with positive probability, $a_i \sim \pi_i(s)$ such that $a_i \neq a_i^E$. Then it is immediate that at state s, $\mathbb{E}_{\boldsymbol{a} \sim \boldsymbol{\pi}}[\mathbb{1}_{\bigcap_i a_i = a_i^E}(s)] < 1$. Combined with the non-zero probability of visiting every state (Condition 1), this inequality then implies $J(\boldsymbol{\pi}_E) < K$. By contradiction, $\boldsymbol{\pi}_E$ is the unique optimizer for J. □

Next, we establish that individual agents performing GAIL updates maximizes a lower bound on $J(\boldsymbol{\pi})$. We leverage the single-agent GAIL convergence result by [12] to show this result.

Lemma 5. *Let t be the time step at which agent i's policy is updated. For all agents j, denote the optimal discriminators of Equation 1 as $r_{\phi_j}^{opt}$. Suppose agent i updates its policy parameters from θ_i^t to θ_i^{t+1} such that $\mathcal{L}(\theta_i^{t+1}, \phi_i^{opt} | \theta_{i-}^t) < \mathcal{L}(\theta_i^t, \phi_i^{opt} | \theta_{i-}^t)$. This decrease in loss is equivalent to increasing the agent i's expected discriminator reward.*

Proof. First, note that updating a single agent policy while keeping all discriminators fixed does not alter the expert value term $V(\pi_{E_k}, r_{\phi_k^{opt}})$ or the regularizer term $\psi(\phi_k^{opt})$ in the loss definition $\mathcal{L}$. Thus, the condition that agent i's loss has decreased is equivalent to the value of agent i increasing:

$$V(\theta_i^{t+1}, \phi_i^{opt} | \theta_{i-}^t) > V(\theta_i^t, \phi_i^{opt} | \theta_{i-}^t). \tag{5}$$

For convenience of notation, agent i's policy at time t will be written as π_i^t, and the ith discriminator $r_{\phi_i}^{opt}$ implicitly indicated by the action subscript, $r(s, a_i)$. Similarly, we will write the **state** visitation distribution induced by the policies π_i^{t+1}, π_{i-}^t by ρ^{t+1}, and the distribution induced by π_i^t, π_{i-}^t as ρ^t. Rewriting Equation 5 in terms of visitation distributions:

$$\frac{1}{1-\gamma} \mathop{\mathbb{E}}_{s \sim \rho^{t+1}, a_i \sim \pi_i^{t+1}} [r_{\phi_i}(s, a_i)] > \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{s \sim \rho^t, a_i \sim \pi_i^t} [r_{\phi_i}(s, a_i)]$$

$$\mathop{\mathbb{E}}_{s \sim \rho^{t+1}, a_i \sim \pi_i^{t+1}} [r_{\phi_i}(s, a_i)] > \mathop{\mathbb{E}}_{s \sim \rho^t, a_i \sim \pi_i^t} [r_{\phi_i}(s, a_i)]. \tag{6}$$

□

Assumption 1. *Let $c \in \mathbb{R}^+$. Suppose for all agents i, the learned discriminator reward at each state is proportional to an action-matching indicator function:*

$$r_{\phi_i}(s, a) = c\mathbb{1}_{a_i = a_i^E}(s) = c(1 - \mathbb{1}_{a_i \neq a_i^E}(s)).$$

We next show that $J(\boldsymbol{\pi})$ is lower bounded by the sum of the individual action rewards r_{ϕ_i} for all agents i, over all states.

Theorem 1 (action-matching objective)**.** *The joint action-matching objective $J(\boldsymbol{\pi})$ is lower bounded by the following sum over individual action rewards r_{ϕ_i}:*

$$L(\boldsymbol{\pi}) := \sum_{s \in S} \rho(s) \left[\sum_{i=1}^{K} \frac{1}{c} \mathop{\mathbb{E}}_{a \sim \pi_i} [r_{\phi_i}(s, a)] \right]. \tag{3}$$

Proof. Let us begin by rewriting $J(\boldsymbol{\pi})$ in terms of action *mismatches*.

$$\begin{aligned} J(\boldsymbol{\pi}) &= \sum_{s \in S} \rho(s) \left[(K-1) + \mathop{\mathbb{E}}_{\boldsymbol{a} \sim \boldsymbol{\pi}} [1 - \mathbb{1}_{\bigcup_i a_i \neq a_i^E}(s)] \right] \\ &= \sum_{s \in S} \rho(s) \left[(K-1) + 1 + \mathop{\mathbb{E}}_{\boldsymbol{a} \sim \boldsymbol{\pi}} [-\mathbb{1}_{\bigcup_i a_i \neq a_i^E}(s)] \right]. \end{aligned}$$

This formulation allows us to apply the Union Bound:

$$\begin{aligned} J(\boldsymbol{\pi}) &\geq \sum_{s \in S} \rho(s) \left[K + \sum_{i=1}^{K} \mathop{\mathbb{E}}_{a \sim \pi_i} [-\mathbb{1}_{a_i \neq a_i^E}(s)] \right] \\ &= \sum_{s \in S} \rho(s) \left[\sum_{i=1}^{K} \mathop{\mathbb{E}}_{a \sim \pi_i} [1 - \mathbb{1}_{a_i \neq a_i^E}(s)] \right] \\ &= \sum_{s \in S} \rho(s) \left[\sum_{i=1}^{K} \frac{1}{c} \mathop{\mathbb{E}}_{a \sim \pi_i} [r_{\phi_i}(s, a)] \right] \\ &= L(\boldsymbol{\pi}). \end{aligned}$$

□

Theorem 1 makes clear the relationship between the multi-agent imitation learning objective and the single-agent imitation learning objective. This is important because each term of $L(\boldsymbol{\pi})$ corresponds to an individual agent's loss in our independent GAIL learning algorithm. Note also that the additive form of $L(\boldsymbol{\pi})$ is similar to the value factorization assumptions made by algorithms like VDN [38].

Lemma 5 states that a single agent i updating its policy to improve its own GAIL loss is equivalent to increasing the expected action reward $r_{\phi_i}(s, a)$, where the expectation over states is with respect to some updated state visitation distribution ρ^{t+1} (Equation 6).

It is difficult to say anything directly about the expected reward of agent $j \neq i$, $\mathbb{E}_{s \sim \rho^{t+1}}[r(s, a_j)]$, as ρ^{t+1} may be a state distribution over which agent j makes more mistakes (i.e. taking actions that don't match the expert's). While in general agent j's expected reward may decrease due to agent i's update, we show that under the following conditions, agent i's update increases a lower bound to $L(\boldsymbol{\pi})$ that is independent of the state distribution.

Corollary 1 (lower bound)**.** *Let ϵ be the minimum probability of visiting any state. For all ρ, $L(\boldsymbol{\pi})$ is lower bounded by $L_\epsilon(\boldsymbol{\pi})$:*

$$L(\boldsymbol{\pi}) > \sum_{s \in S} \sum_{i=1}^{K} \epsilon \left[\frac{1}{c} \mathop{\mathbb{E}}_{a \sim \pi_i} [r_{\phi_i}(s, a)] \right] =: L_\epsilon(\boldsymbol{\pi}). \tag{4}$$

Proof. The proof follows from the definition of $L(\boldsymbol{\pi})$ and that for all $s \in S$, and all ρ encountered in training, $\rho(s) > \epsilon$. □

Observation 1. *$L_\epsilon(\boldsymbol{\pi})$ is bounded by*

$$0 \leq L_\epsilon(\boldsymbol{\pi}) \leq \epsilon |\mathcal{S}| K.$$

Lemma 6. *Suppose the joint expert policy $\boldsymbol{\pi}_E$ is deterministic. Then $\boldsymbol{\pi}_E$ is the unique maximizer of $L_\epsilon(\boldsymbol{\pi})$.*

Proof. Proof for this Lemma follows closely the proof for Lemma 4.

(Maximization) Since $\boldsymbol{\pi}_E$ is deterministic, by definition, for each agent i, each term

$$\mathop{\mathbb{E}}_{a_i \sim \pi_{E_i}} [\mathbb{1}_{a_i \neq a_i^E}(s)] = 0$$
$$\mathop{\mathbb{E}}_{a_i \sim \pi_{E_i}} [1 - \mathbb{1}_{a_i \neq a_i^E}(s)] = 1.$$

Summing over all agents and taking the sum weighted by ϵ over all states, $L_\epsilon(\boldsymbol{\pi}_E) = \epsilon |\mathcal{S}| K$, which means that $\boldsymbol{\pi}_E$ achieves the upper bound of L_ϵ.

(Uniqueness) Suppose there is at least one agent policy $\pi_i \neq \pi_{E_i}$ such that the joint policy $\boldsymbol{\pi}$ also achieves the upper bound, i.e. $L_\epsilon(\boldsymbol{\pi}) = \epsilon |\mathcal{S}| K$. Then there must be a state s such that with non-zero probability, $a_i \sim \pi_i(s)$ such that $a_i \neq a_i^E$. It follows that $\mathbb{E}_{a_i \sim \pi_{E_i}}[\mathbb{1}_{a_i \neq a_i^E}(s)] > 0$, meaning $L_\epsilon(\boldsymbol{\pi}_E) < \epsilon |\mathcal{S}| K$. By contradiction, $\boldsymbol{\pi}_E$ is the unique optimizer of L_ϵ. □

Condition 2. *Let $V(\pi)$ denote the value of an agent following a single-agent imitation learning algorithm. $|V(\pi_t) - V(\pi^E)|$ is then the optimality gap at update t of the agent. Suppose that $|V(\pi_t) - V(\pi^E)| \leq \eta(t)$, where as $t \to \infty$, $\eta(t) \to 0$.*

This condition says that the single-agent imitation learning process should converge to the optimal (expert) policy with convergence rate dictated by $\eta(t)$. For our setting, Guan et al. [12] shows that the single-agent GAIL algorithm converges (Theorem 3 and 4).

The next corollary shows that convergence of the single-agent imitation learning process is sufficient to guarantee the convergence of the multi-agent imitation learning scheme discussed in the main paper.

Corollary 2. *There exists $H \in \mathbb{N}^+$, $H < \infty$ such that within H updates, agent i is able to improve its policy such that it increases the probability of matching the expert's action, summed over all states:*

$$\sum_{s \in S} \epsilon \left[\frac{1}{c} \mathop{\mathbb{E}}_{a \sim \pi_i^{t+H}} [r_{\phi_i}(s,a)] \right] > \sum_{s \in S} \epsilon \left[\frac{1}{c} \mathop{\mathbb{E}}_{a \sim \pi_i^{t}} [r_{\phi_i}(s,a)] \right].$$

Proof. For a single agent i, define $V(\pi) = \sum_{s \in S} \rho_\pi(s) \left[\frac{1}{c} \mathbb{E}_{a \sim \pi_i}[r_{\phi_i}(s,a)] \right]$. Rewrite as follows:

$$\begin{aligned} V(\pi) &= \sum_{s \in S} \rho_\pi(s) \mathop{\mathbb{E}}_{a \sim \pi_i} [r_{\phi_i}(s,a)] \\ &= \frac{1}{c} \sum_{s \in S} (\rho_\pi(s) - \epsilon) \mathop{\mathbb{E}}_{a \sim \pi_i} [r_{\phi_i}(s,a)] + \frac{1}{c} \sum_{s \in S} \epsilon \mathop{\mathbb{E}}_{a \sim \pi_i} [r_{\phi_i}(s,a)]. \end{aligned}$$

Define the first term as $a(\pi) := \frac{1}{c} \sum_{s \in S} (\rho_\pi(s) - \epsilon) \mathbb{E}_{a \sim \pi_i}[r_{\phi_i}(s,a)]$, and the second term as $b(\pi) := \frac{1}{c} \sum_{s \in S} \epsilon \mathbb{E}_{a \sim \pi_i}[r_{\phi_i}(s,a)]$. Note that $b(\pi)$ is the quantity of interest in the corollary.

The properties of the max operation directly imply that

$$\begin{aligned} \max_\pi V(\pi) &= \max_\pi [a(\pi) + b(\pi)] \\ &\leq \max_\pi a(\pi) + \max_\pi b(\pi). \end{aligned}$$

The expert policy π_E should maximize $V(\pi)$ for any single-agent imitation learning algorithm. Note also that π_E maximizes $a(\pi)$ and $b(\pi)$. Thus in our setting, the inequality in the above set of equations is actually an equality for the expert policy:

$$\begin{aligned} \max_\pi V(\pi) &= V(\pi_E) \\ &= a(\pi_E) + b(\pi_E) = \max_\pi a(\pi) + \max_\pi b(\pi). \end{aligned}$$

Assume that there is a policy $\pi \neq \pi_E$ such that $b(\pi) < b(\pi_E)$, and that $b(\pi)$ cannot be improved within finite updates. This contradicts Condition 2, which establishes that the single-agent imitation learning algorithm should be able to improve the agent policy until its value converges to the value of the expert policy. □

Corollary 2 implies that within a finite number of policy updates by agent i, the quantity $L_\epsilon(\boldsymbol{\pi})$ increases, because the other terms corresponding to agents $j \neq i$ are unchanged. By Theorem 1 and Lemma 6, $L_\epsilon(\boldsymbol{\pi})$ is upper bounded by the constant $\epsilon|S|K$. Thus, by the monotone convergence theorem, the objective $L_\epsilon(\boldsymbol{\pi})$ converges. Further, the expert policy $\boldsymbol{\pi_E}$ is a maximizer of $L(\boldsymbol{\pi})$, $L_\epsilon(\boldsymbol{\pi})$, and $J(\boldsymbol{\pi})$ (Lemma 4 and Lemma 6).

A.2 Mixed Task and Imitation Reward

Theorem 2. *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). Then $\boldsymbol{\pi}_E$ are a Nash equilibrium for reward functions of the form, $\alpha R_T + \beta R_{I,i}$, for any $\alpha, \beta > 0$.*

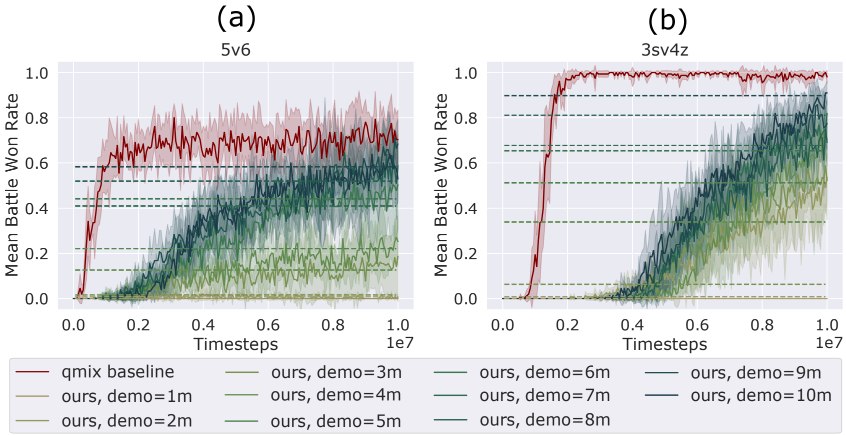

Figure 4: Learning curves of DM^2 (our method) trained with demonstrations sampled every million steps in the learning of the original demonstrator policy. Horizontal dotted lines indicate the demonstration qualities, colored to match corresponding learning curves. QMIX (centralized baseline) is included for reference.

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. □

A.3 Experimental Details

Sensitivity to Demonstration Quality. One finding in the experimental section of the paper is that DM^2 is relatively insensitive to demonstration quality beyond a certain baseline level of competence. To further support this claim, we train DM^2 with more demonstration qualities. Figure 4 shows that the algorithm improves monotonically as the demonstration quality improves, but quickly saturates. This result indicates it is important to supply good demonstrations, but not necessary to supply optimal demonstrations.

Demonstration Styles of Ablation Study. The ablation study examines expert demonstrations that vary in two dimensions: co-trained versus concurrently sampled. 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. 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.

Implementation Details. The algorithm implementations are based on the multi-agent PPO implementation provided by Yu et al. [43] (MIT license) and the PyMARL code base [30] (Apache license). The StarCraft environment is also provided by Samvelyan et al. [30] (MIT license).

All MARL implementations in this paper have fully separate policy/critic networks and optimizers per agent. For all IPPO agent s, 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 [5]. We attempted running QMIX with the the IPPO agent architecture, but found that the performance of QMIX significantly suffered (Figure 5 on 5v6). Thus, for the QMIX experiments in the main body of the paper, the better-performing policy architecture was applied.

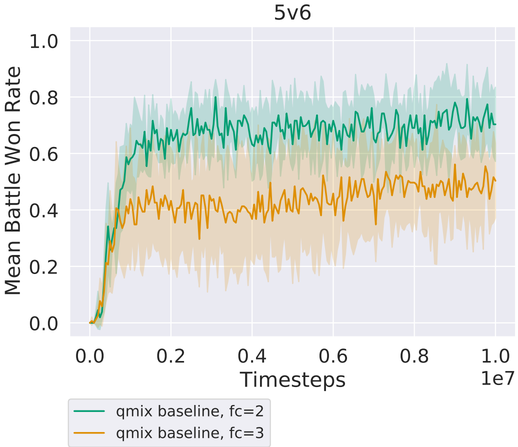

Figure 5: QMIX is sensitive to the agent policy architecture. Performance on the 5v6 task suffers significantly when an extra fully connected layer is added.

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

Hyperparameters. For QMIX, the default parameters specified in Rashid et al. [27] are used for both tasks. For IPPO, and the IPPO component of DM^2, mostly default parameters (as specified in [27, 43]) were used. The hyperparameters that varied between tasks or were tuned are provided in Table 1. The remaining hyperparameters may be viewed with the code 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.

Computing Architecture. All experiments were performed without parallelized training, on machines running Ubuntu 18.04 with the following configurations:

- Intel Xeon CPU E5-2698 v4; Nvidia Tesla V100-SXM2 GPU.
- Intel Xeon CPU E5-2630 v4; Nvidia Titan V GPU.

	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.

[5] This is the architecture used in Rashid et al. [27]