BIG PICTURE THINKING: ENHANCE MULTI-AGENT IMITATION LEARNING THROUGH GLOBAL DEPENDENCIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-agent reinforcement learning (MARL) has emerged as a promising approach for solving complex problems involving multi-agent collaboration or competition. Recently, researchers have turned to imitation learning to avoid the explicit design of intricate reward functions in MARL. By formulating the problem as a distribution-matching task based on expert trajectories, imitation learning enables agents to continually approximate expert policies without requiring manual reward engineering. However, classical multi-agent imitation learning frameworks, such as MAGAIL, often treat individual agent's distribution matching independently, disregarding the intricate dependencies that arise from agent cooperation. This neglect results in inaccurate estimations of action-value functions, weak feedback from the discriminator, and a significant vanishing gradient problem. This paper proposed a novel multi-agent joint distribution matching framework based on the Transformer architecture. It explicitly models global dependencies among agents within the generator and discriminator components sequentially and autoregressively. We also theoretically prove the effectiveness of this framework in enhancing reward variance and advantage gradient. Extensive experiments demonstrated the remarkable performance improvements achieved by our proposed method on various benchmarks.

1 INTRODUCTION

In recent years, Reinforcement Learning (RL) has significantly advanced in continuous decision-making tasks involving a single agent (Mnih et al., 2013; Schulman et al., 2017; Chen et al., 2021). However, real-life applications necessitate collaboration or competition among multiple agents, such as machine dexterity control (Chen et al., 2022), multi-UAV control (Yun et al., 2022), and multiplayer games (Samvelyan et al., 2019). Consequently, Multi-Agent Reinforcement Learning (MARL) has garnered substantial attention. Nonetheless, a primary limitation of RL and MARL lies in the intricate process of designing explicit reward functions for complex tasks (Russell, 1998; Ng & Russell, 2000; Fu et al., 2017; Hadfield-Menell et al., 2017), which is essential for facilitating robust online learning. Specifically, in multi-agent systems, establishing proper reward functions for individual agents to induce desired behaviors presents a formidable challenge due to the interdependency of each agent's objectives, mediated through unstructured implicit correlations (Song et al., 2018; Wang et al., 2021).

Imitation learning (IL) (Hussein et al., 2017) offers a viable alternative for guiding agents without explicit rewards by leveraging expert demonstrations capable of accomplishing the given task. This approach treats the agent's learning process as a distribution matching problem, intending to approximate the expert policy continuously (Hadfield-Menell et al., 2016). A prevalent method in IL involves the utilization of a generative adversarial framework, which has demonstrated significant advancements in most single-agent tasks (Ho & Ermon, 2016). Nonetheless, in multi-agent IL, a substantial and fundamental hurdle arises in effectively modeling the dependencies among multiple agents while providing individual reward functions that enable the decoupling of the global perspective (Wang et al., 2021). Acquiring proficient individual reward functions through the agent's independent state-action learning is feasible in single-agent settings. Conversely, in multi-agent settings, it becomes imperative to establish a joint reward function distribution that accounts for the

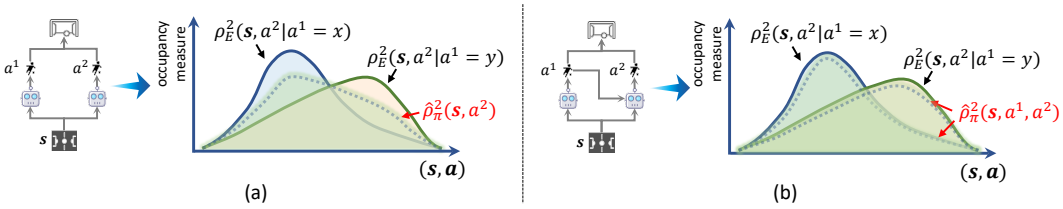

Figure 1: In a collaborative task involving two agents, labeled as i^1 and i^2, with action space $\{x, y\}$, i^2's policy is strongly influenced by i^1's actions, leading to varying occupancy measures for i^2. (a) Policy-Independent Distribution Matching: When the agents act independently, imitation learning approximates their policy occupancy measures separately. When i^2's policy varies with i^1's actions, distribution matching attempts to align a new distribution with both individual distributions, potentially causing more significant errors and uniform distributions. (b) Policy-Dependent (Joint) Distribution Matching: When agents are interdependent, imitation learning considers i^1's actions when approximating i^2's distribution. It results in joint distributions linked to other agents' actions, reducing errors and identifying advantageous actions.

interdependence among agents, thereby facilitating the modeling of collaborative behaviors and the provision of precise individual reward functions (i.e., credit assignment).

However, previous studies in multi-agent imitation learning have predominantly formalized it as a separate distribution matching problem concerning independent individual policies among the agents (Song et al., 2018), as shown in Fig. (1). This formalization assumes that the agents make independent decisions based on independent observations (i.e., mean-field factorization (Yang et al., 2018) of the joint policy) (Zhan et al., 2019; Le et al., 2017; Yu et al., 2019). However, this perspective overlooks the agents' intricate interdependencies and cooperative relationships. The absence of these underlying correlations among the agents gives rise to specific challenges during training within the generative adversarial framework (Wang et al., 2021). These challenges manifest as pronounced imbalances in training speed between the generator and discriminator, leading to minimal variations in rewards and the subsequent vanishing of policy gradients for the generator (Zhang et al., 2022). The root cause of this issue stems from the fact that the discriminator independently generates rewards for each agent. This independent reward generation introduces biases and delays in the reward signal, thereby impeding satisfactory exploration efficiency in reinforcement learning and impeding the training speed of the generator. Consequently, such circumstances often result in the development of a well-trained discriminator alongside a poorly performing generator suffering from the problem of gradient vanishing.

In order to tackle the challenges above, the key is to model the dependence structure among multiple agents. Recent studies conducted by Tian et al. (2019) and Liu et al. (2020) propose using opponent modeling to achieve pertinent policies. Nonetheless, this approach incurs unnecessary modeling expenses and redundancy, while still needing coordination during execution. Motivated by the successful application of the Transformer (Vaswani et al., 2017) architecture in reinforcement learning (Chen et al., 2021; Wen et al., 2022), we advocate for the comprehensive utilization of the powerful sequence modeling capabilities offered by the Transformer in the context of multi-agent imitation learning and propose $\mathbf{MILD}^2$, intending to capture the interdependencies among multiple agents. By employing a Transformer-based framework for multi-agent imitation learning, we can accurately approximate the global distribution of rewards among the agents, amplify the reward variance and advantage during model training, and mitigate convergence issues arising from notable disparities in learning speeds between the generator and discriminator within multi-agent tasks. The contributions of this paper can be summarized as follows: First, we reveal the need for more complex dependency modeling in traditional independent multi-agent imitation learning frameworks, leading to insignificant reward and advantage variances and hindering model convergence. Secondly, from the perspective of enhancing advantage variances, we provide theoretical evidence that the joint reward and policy distribution matching architecture with complex dependency modeling is generally superior to the traditional individual discriminator and generator of independent agents. Thirdly, based on the joint distribution matching framework proposed above, we innovatively construct a sequential modeling generator and discriminator based on the Transformer architecture. Finally, experiments on three cooperative benchmarks demonstrated that our method outperformed baselines regarding convergence and accumulated rewards, while exhibiting good stability.

2 BACKGROUND & PRELIMINARIES

This section describes the problem setup and notations for MARL, imitation learning, and Transformer models.

2.1 MARKOV GAMES

Cooperative multi-agent reinforcement learning (MARL) problems are commonly formulated as Markov games, which extend the framework of Markov decision processes (MDPs) (Littman, 1994). A Markov game for N agents is defined by a tuple $(N, \mathcal{S}, \{\mathcal{A}_i\}_{i=1}^N, \mathcal{T}, \{\mathcal{R}_i\}_{i=1}^N, \gamma)$, where $\mathcal{S}$ denotes the set of states and $\{\mathcal{A}_i\}_{i=1}^N$ represents N sets of actions. The transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A}_1 \times \ldots \times \mathcal{A}_N \rightarrow P(\mathcal{S})$ describes the state transition process, where $P(\mathcal{S})$ is the set of probability distributions over $\mathcal{S}$. Given that the system is in state s_t at timestep t, the agents jointly take actions $(a_1, \ldots, a_N)$, and the state transitions to s_{t+1} with probability $T(s_{t+1}|s_t, a_1, \ldots, a_N)$. The joint policy $\pi_\theta = [\pi_{\theta_1}, \ldots, \pi_{\theta_N}]$ represents the vector of individual agent policies. Occasionally, we may omit the policy parameters θ for convenience. It is worth noting that each agent has access to the complete state information. Let $i_{1:N}$ be a permutation of N agents. To refer to a subset of agents from i_k to i_j $(1 \leq k \leq j \leq N)$, we employ the subscript notation $k : j$, such that $\pi_{k:j}$ denotes the agent policies $\{\pi_k, \pi_{k+1}, \ldots, \pi_j\}$. Each agent i is associated with an individual reward function $\mathcal{R}_i : \mathcal{S} \times \mathcal{A}_i \times \ldots \times \mathcal{A}_N \rightarrow \mathbb{R}$. The objective of each agent is to maximize its expected return, given by $\mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_i^t\right]$. Here, r_i^t denotes the reward received by agent i at timestep t, and the discount factor $\gamma \in [0, 1)$ determines how much future rewards are discounted. The task rewards given by environments are identical across all agents in the cooperative tasks examined in this study.

2.2 DISTRIBUTION MATCHING FOR IMITATION LEARNING

Imitation learning is a problem scenario in which an agent aims to replicate trajectories $\{\tau_1, \tau_2, \ldots\}$ that are demonstrated by an expert policy π_E (Schaal, 1996; Hussein et al., 2017; Ho & Ermon, 2016). Each trajectory τ consists of state-action pairs $\{(s_0, a_0), (s_1, a_1), \ldots\}$. Several approaches have been proposed to tackle the imitation learning problem. Behavioral cloning employs supervised learning to imitate expert demonstrations and learn the policy that maximizes the likelihood (Bain & Sammut, 1995; Torabi et al., 2018; Fujimoto & Gu, 2021). Inverse reinforcement learning (IRL) involves recovering a reward function, which can then be used to train an expert policy using reinforcement learning (Ng & Russell, 2000; Hadfield-Menell et al., 2016). In the context of $\mathrm{IRL}(\pi_E)$, the objective is to retrieve a reward function that optimizes the demonstrated trajectories by π_E. GAIL (Ho & Ermon, 2016) interprets the imitation learning problem as matching two occupancy measures, i.e., the distribution over states and actions encountered when exploring the environment with a policy. Formally, for a policy π, it is defined as $\rho_\pi(s, a) = \pi(a|s) \sum_{t=0}^{\infty} \gamma^t P(s_t = s|\pi)$. GAIL draws a connection between IRL and occupancy measure matching, showing that the former is a dual of the latter:

$$IRL_\psi(\pi_E) = \mathrm{argmin}_{\pi \in \Pi} -H(\pi) + \psi^*(\rho_\pi - \rho_E), \tag{1}$$

where $\psi^*(x) = \sup_y x^T y - \psi(y)$ is convex conjugate of ψ, which could be interpreted as a measure of similarity between the occupancy measures of expert policy and agent's policy. Wang et al. (2023) view multi-agent imitation learning as a distribution matching problem. They define the state-action visitation distribution of a joint policy $\pi = [\pi^1, \ldots, \pi^N]$ as $\rho_\pi(s, \mathbf{a}) := (1 - \gamma) \prod_{i=1}^N \pi^i(a^i|s) \sum_{t=0}^{\infty} \gamma^t P(s_t = s|(\pi))$. Therefore, distribution matching provides a solution to the imitation learning problem. Guan et al. (2021) demonstrate that the GAIL algorithm converges to the expert policy in the single-agent case using various policy gradient techniques (Guan et al., 2021), including TRPO (Schulman et al., 2015). They introduce the GAIL problem as the following min-max problem:

$$\begin{aligned} &\min_\theta \max_\phi \mathcal{L}(\theta, \phi) \\ &\quad \text{s.t. } \mathcal{L}(\theta, \phi) := V(\pi_E, r_\phi) - V(\pi_\theta, r_\phi) - \psi(\phi). \end{aligned} \tag{2}$$

Here, $V(\pi, r) = \mathbb{E}_{s_0 \sim \rho_0} \mathbb{E}_\pi [\sum_{t=0}^{\infty} \gamma^t r_t]$ represents the expected return starting from an initial state according to policy π and using reward function $r(s, a)$. In the multi-agent scenario, imitation learning becomes more complex due to the involvement of multiple expert policies $\pi_{E_0}, \ldots, \pi_{E_N}$ in generating the expert trajectories (Song et al., 2018; Wang et al., 2023). Successful imitation in this setting necessitates coordinating the policies of all N agents.

This paper mainly focuses on the MARL problem and multi-agent adversarial generative-based imitation learning frameworks, such as MAGAIL. In this paper, we write $-i^{k:j}$ to denote the set of all agents excluding $i^k, \ldots, i^j$. We define the multi-agent state-action value function for agents $i^{1:k}$ as $\mathbf{Q}_\theta^{1:k}(\mathbf{s}, \mathbf{a}^{1:k})$, which is the expected total reward once agents $i^{1:k}$ have taken their actions. Note that for $k = 0$, this becomes the state value function; for $k = N$, this is the usual state-action value function. As such, we can define the multi-agent advantage function as $\mathbf{A}_\theta^{1:k}(\mathbf{s}, \mathbf{a}^{1:m}, \mathbf{a}^{1:k}), (m \leq k)$, which is the advantage of agents $i^{1:k}$, playing $\mathbf{a}^{1:k}$, given $\mathbf{a}^{1:m}$.

2.3 The Transformer model

The Transformer model (Vaswani et al., 2017) is a neural network architecture widely used in various domains such as natural language processing, image processing, time-series forecasting, and sequential decision-making (Chen et al., 2021; Janner et al., 2021; Wen et al., 2022), with impressive performance. It excels in capturing long-term dependencies, parallel processing, and extracting patterns from large datasets. Unlike RNNs and CNNs, it overcomes limitations in modeling long-term dependencies and incorporating global contextual information. The self-attention mechanism, a critical component, enables focusing on different positions in the input sequence and establishing meaningful relationships between tokens or token pairs. The attention function, represented as $\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V}$, utilizes trainable vectors of queries, keys, and values ($\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$) to compute attention weights. The dimensionality of $\mathbf{Q}$ and $\mathbf{K}$ is represented by d_k, and these vectors share the same self-attention parameters. The Transformer model follows an encoder-decoder architecture. The encoder consists of stacked layers of self-attention networks and feed-forward neural networks to enhance the representation of the input sequence, considering both local and global dependencies. The decoder operates similarly, autoregressively generating output tokens conditioned on the input sequence.

3 Theoretical Analysis

This section will discuss the main issues of imitation learning methods based on distributed matching in multi-agent scenarios when using the independent framework. From the perspective of advantage variance, we attempt to explain the possible causes of gradient vanishing in multi-agent generative adversarial algorithms represented by MAGAIL and propose the theoretical guarantee of significantly improving performance through global dependency-enhanced discriminators.

3.1 Problem Statement

In the conventional single-agent GAIL framework, the generator competes with the discriminator, with the discriminator assigning rewards through a classification task, and the generator using reinforcement learning, specifically policy gradients, to maximize these rewards. GAIL differs significantly from GAN in generator optimization (Zhang et al., 2022). GAN relies on supervised learning, while GAIL uses policy gradients. However, using policy gradients in GAIL leads to high gradient estimation variance, slowing learning and creating an imbalance between the generator and discriminator, affecting GAIL's convergence (Baram et al., 2017).

MAGAIL (Song et al., 2018) extends GAIL to multi-agent cooperative tasks by matching individual policy and reward distributions, but it needs help with complex agent dependencies. This lack of global dependencies results in low rewards and affects adversarial generation convergence. In multi-agent settings, global dependency modeling for credit assignment and action generation faces challenges due to collaborative interactions among multiple agents (Chang et al., 2003). This difficulty leads to two main training issues: (1) Due to difficulty in assessing agents' contributions to team success or failure, it is easy to introduce unwanted bias in reward acquisition. (2) Independent policy/reward modeling significantly decelerates the training speed of the generator compared to the discriminator in the estimation of the Multi-Agent Policy Gradient (MAPG) (Kuba et al., 2021).

3.2 Learning speed imbalance problem

In multi-agent generative adversarial imitation learning, the issue of inadequate rewards for the generator hinders its training. This problem occurs because the discriminator learns faster than

the generator, leading to consistently low and similar rewards for the generator's actions. To illustrate this problem, consider the loss function of Multi-Agent Trust Region Policy Optimization (MATRPO) (Li & He, 2020; Kuba et al., 2022). The loss function of MATRPO is $J(\theta) = \mathbb{E}_{\mathbf{s}\sim\rho_{\pi_{old}},\mathbf{a}\sim\pi_{old}}[\frac{\pi_\theta(\mathbf{a}|\mathbf{s})}{\pi_{old}(\mathbf{a}|\mathbf{s}))}\mathbf{A}_{\pi_{old}}(\mathbf{s},\mathbf{a}))]$, where $\pi_{old}(\mathbf{a}|\mathbf{s})$ denotes the sampling (behavior) joint policy, $\rho_{\pi_{old}}$ denotes the distribution generated by π_{old}. The gradient of $J(\theta)$ w.r.t θ is proportional to $\mathbf{A}_{\pi_{old}}(\mathbf{s},\mathbf{a})$, i.e., the joint advantage function. In some cases, when an independent discriminator assigns low rewards to the generator, the differences in rewards for joint actions by multiple agents become insignificant. The discriminator categorizes all agent behaviors as non-expert, resulting in low rewards for all agents, even if some perform well individually. Consequently, the advantage function, which measures the advantages of joint agent actions, remains minimal, causing the joint policy's training to fall into the bottleneck.

3.3 Reward Variance Analysis of the Discriminator

The optimization process of RL algorithms typically depends on assessing the disparity between the rewards the environment offers and the present value function. This disparity is commonly captured by the advantage function in Actor-Critic algorithms (Konda & Tsitsiklis, 1999). In the majority of multi-agent actor-critic algorithms (Li & He, 2020; Kuba et al., 2022), the joint advantage function can be computed as $\mathbf{A}_{\pi_\theta}(\mathbf{s},\mathbf{a}) = r(\mathbf{s},\mathbf{a}) + \gamma\mathbf{V}_\phi(\mathbf{s}') - \mathbf{V}_\phi(\mathbf{s})$, where s' is the next state, $\mathbf{V}_\phi$ is a neural network parameterized by ϕ to approximate the individual value function of joint policy π_θ. The optimization objective of $\mathbf{V}_\phi$ is $\min_\phi [\mathbf{R}^{cdr}(\mathbf{s},\mathbf{a}) - \mathbf{V}_\phi(\mathbf{s})]^2$, where $\mathbf{R}^{cdr}(\mathbf{s},\mathbf{a})$ is the cumulative discounted rewards provided by the discriminator in the multi-agent generative adversarial imitation learning framework.

Theorem 1. *Let R^{cdr}_ω be the cumulative discounted individual reward given by the discriminator parameterized by w, and let V_ϕ be the joint value function parameterized by ϕ. Let $\mathcal{D}^\pi$ denote the total data collected by π. Then increasing the joint advantage variance $\sum_{k=1}^N \mathrm{Var}[A^k_{\pi_\theta}(\mathbf{s},a_k)]$ of multi-agent policy gradients is equivalent to solving a bi-level joint value loss optimization problem that is related to the joint reward function and the joint value function:*

$$\max_w \sum_{k=1}^N \mathbb{E}_{\mathbf{s},a^k\in\mathcal{D}^{\pi^k}}[R^{cdr}_\omega(\mathbf{s},a^k) - V_{\phi^*}(\mathbf{s})]^2, \tag{3}$$

$$\text{s.t. } \phi^*(\mathbf{s}) = \min_\phi \sum_{k=1}^N \mathbb{E}_{\mathbf{s},a^k\in\mathcal{D}^{\pi^k}}[R^{cdr}_\omega(\mathbf{s},a^k) - V_\phi(\mathbf{s})]^2,$$

where $R^{cdr}_\omega(\mathbf{s},a^k) = -\sum_{t=l}^T \gamma^{t-l}(\log\sigma(1 - D_\omega(\mathbf{s}_t,a^k_t)))$ denotes the individual reward function for the agent i^k, and σ denotes activation function (generally Sigmoid*).*

For proof see Appendix (B.2). According to Thm. (1), to enhance the variance of joint advantage, the objective is to maximize the expected L2 norm of the discrepancy between the cumulative discounted rewards the discriminator offers and the joint value function. The natural idea is to decouple the joint reward function and train a discriminator with credit assignment capability, thus transforming an independent reward function into a global joint one. Below, we present a theorem proving that a global dependency-enhanced discriminator framework that offers a joint reward distribution has a more significant advantage variance than the independent discriminator.

Lemma 2. *(Multi-agent Advantage Decomposition (Kuba et al., 2021)). Let $i^{1:N}$ be a permutation of N agents. For any state $\mathbf{s}\in\mathcal{S}$ and joint actions $\mathbf{a} = \mathbf{a}^{1:N}\in\mathcal{A}$, the following equation holds for any subset of N agents and any permutation of their labels: $\mathbf{A}^{k+1,\dots,N}_{\pi_\theta}(\mathbf{s},\mathbf{a}^{1:k},\mathbf{a}^{k+1:N}) = \sum_{j=k+1}^N A^j_{\pi_\theta}(\mathbf{s},\mathbf{a}^{1:j-1},a^j)$, where $k = 0,\dots,N-1$.*

Theorem 3. *Let $A^k_{\pi_\theta}(\mathbf{s},\mathbf{a}^{-k},a^k)$ be the advantage function of agent i^k with global dependency-enhanced discriminator, and let $\mathbf{A}_{\pi_\theta}(\mathbf{s},\mathbf{a})$ be the joint advantage function of all agents in independent framework, global dependency-enhanced discriminator framework has a more significant advantage variance compared to the independent framework:*

$$\sum_{k=1}^N \underset{\substack{\mathbf{s},\mathbf{a}^{-k}\in D^{\pi^{-k}}_t\\ \mathbf{s},a^k\in D^{\pi^k}_t}}{\mathrm{Var}}[A^k_{\pi_\theta}(\mathbf{s},\mathbf{a}^{-k},a^k)] \geq \sum_{k=1}^N \underset{\mathbf{s},a^k\in D^\pi_t}{\mathrm{Var}}[A^k_{\pi_\theta}(\mathbf{s},a^k)] \tag{4}$$

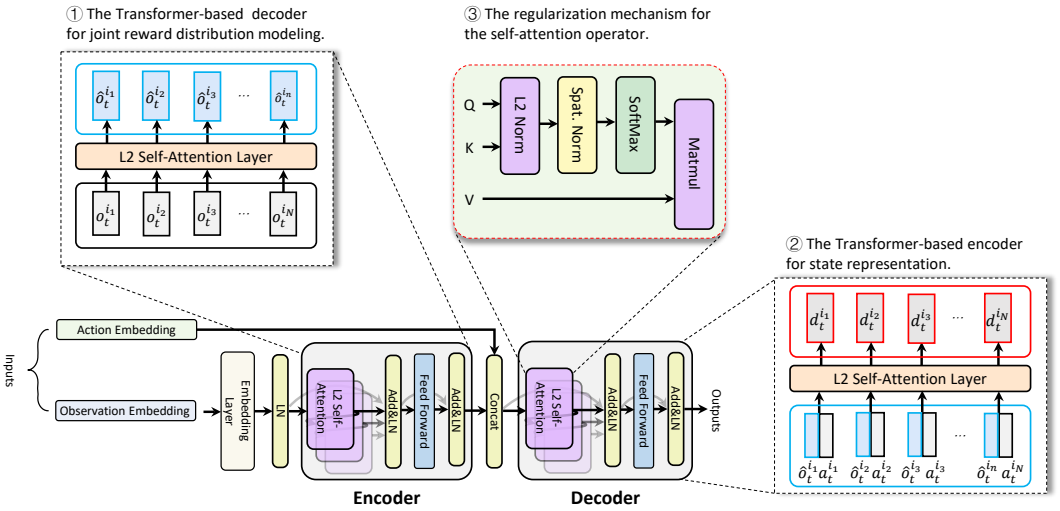

Figure 2: The Transformer-based architecture for our global dependency-enhanced discriminator. The encoder processes all agents' observations into a latent representation, which the decoder uses along with agent actions to produce expert degree scores. L2 self-attention and spectral normalization ensure Lipschitz continuity.

For proof see Appendix (B.2). Let $R_w^{cdr}(\mathbf{s}, \mathbf{a}^{-k}, a^k)$ be the cumulative discounted reward of i^k given by the global dependency-enhanced discriminator parameterized by w. When optimal ϕ^* help keep the difference between $V_\pi(\mathbf{s})$ and $V_{\phi^*}(\mathbf{s})$ at a minimal level, similar to Eq. (12), we also have

$$\max_w \sum_{t=0}^{\infty} \gamma^{2t} \frac{|\mathcal{D}_t^\pi|}{|\mathcal{D}^\pi|} \sum_{k=1}^{N} \operatorname*{Var}_{\substack{\mathbf{s},\mathbf{a}^{-k} \in D_t^{\pi^{-k}} \\ \mathbf{s},a^k \in D_t^{\pi^k}}} [A_{\pi_\theta}^k(\mathbf{s}, \mathbf{a}^{-k}, a^k)] \approx \max_w \operatorname*{\mathbb{E}}_{\substack{\mathbf{s},\mathbf{a}^{-k} \in D_t^{\pi^{-k}} \\ \mathbf{s},a^k \in D_t^{\pi^k}}} \sum_{k=1}^{N} [R_w^{cdr}(\mathbf{s}, \mathbf{a}^{-k}, a^k) - V_{\phi_k^*}(\mathbf{s})]^2. \tag{5}$$

The theorems above showcase that employing a global dependency-enhanced discriminator and generator to model joint reward/policy distribution matching, as opposed to the conventional independent architecture, facilitates enhanced reward variance and advantage variance during the model's training. This approach effectively mitigates the concern of disparate training speeds between the generator and discriminator to achieve better.

4 METHODOLOGY

This section introduces a **M**ulti-agent generative adversarial **I**mitation **L**earning framework via global **D**ependency-enhanced **D**istribution matching (**MILD**^{2}), which focuses on modeling complex agent dependency structure and collaborative behaviors sequentially autoregressively.

4.1 GLOBAL DEPENDENCY-ENHANCED DISCRIMINATOR

The reward function formula given in Eq. (5) indicates that the design of the discriminator should consider global state and action information to establish a joint reward distribution with a global perspective. It is designed to avoid misallocating credit for individual discriminatory rewards, which can mislead and delay the learning speed of the policy generator. Moreover, it addresses the challenges associated with low reward variance and policy gradient vanishing. The construction of the joint reward distribution is precisely accomplished through the discriminator, denoted as D_w. In evaluating the proficiency of agent i^k's actions, the discriminator not only takes into account the global state $\mathbf{s}$ and the agent's action a^k, but also considers the actions $\mathbf{a}^{-k}$ of other agents i^{-k} at the current time step. It enables the discriminator to capture the dependencies and collaborative relationships among multiple intelligent agents. The marginal distribution of the joint reward distribution corresponds to the individual reward function for each agent. A straightforward approach to meet

the requirements above for discriminator design involves employing a multilayer perceptron (MLP) that takes the joint actions of the global state and agents as input. However, this design encounters two challenges: (1) It struggles to effectively capture the dynamic correlations among the actions of intelligent agents, which in turn hampers credit assignment; (2) The high dimensionality of the state and joint action inputs poses difficulties for discriminator learning. Considering the potential power of sequence modeling in the multi-agent domain (Meng et al., 2023; Wen et al., 2022), we model the global dependency structure among agents in a sequential autoregressive paradigm, and utilize the Transformer model to construct the discriminator and achieve the objectives above.

Overall architecture of discriminator. The proposed transformer-based discriminator architecture is illustrated in Fig. (2). It comprises an encoder and a decoder module. The encoder module embeds the agents' observations, constructing a global state representation. It processes a sequence of observations $\mathbf{s} = (o^1, \ldots, o^N)$ in arbitrary order through multiple computational blocks. Each block consists of a self-attention mechanism, an MLP layer, and residual connections to mitigate the issues of gradient vanishing and network degradation as the depth increases. The output encoding of the observations, denoted as $(\hat{o}^1, \ldots, \hat{o}^N)$, captures the information on the agents $(i^1, \ldots, i^N)$ and the higher-level interrelationships that depict the agents' interactions. On the other hand, the decoder module evaluates the expertise level of all agents' actions and generates individual rewards accordingly. It focuses on the embedded joint action $\mathbf{a}^{1:N} = (a^1, \ldots, a^N)$ and the embedded latent state representation $(\hat{o}^1, \ldots, \hat{o}^N)$ within a sequence of decoder blocks. Crucially, each decoder block incorporates a self-attention mechanism for capturing global dependencies instead of using masked self-attention. The decoding process concludes with an MLP layer and skipping connections. The output of the final decoder block consists of a sequence of logits $(d^1, \ldots, d^N)$ representing the joint reward distribution. Moreover, the individual reward function can be calculated by $r_t^k(\mathbf{s}_t, \mathbf{a}_t^{-k}, a_t^k) = -\log \sigma(d_t^k)$, where σ is the activation function. This architecture draws inspiration from the Multi-Agent Transformer (MAT) model (Wen et al., 2022), which first approached MARL as a sequence modeling problem.

Optimization objective of discriminator. We train the discriminator according to Eq. (1), where the measure of similarity $\psi = W_1^d$ for occupancy measure (distribution) matching adopts Wasserstein distance (Xiao et al., 2019). So the loss function of discriminator $\mathcal{L}_d = W_d^1(\rho_\pi, \rho_E)$ is defined as:

$$\mathcal{L}_d = \sup_{r:(\mathcal{S},\mathcal{A})\to\mathbb{R}} \sum_{k=1}^{N} \Big(\mathop{\mathbb{E}}_{y\sim\rho_E^{1:k}}[r(y)] - \mathop{\mathbb{E}}_{x\sim\rho_\pi^{1:k}}[r(x)] + \mathop{\mathbb{E}}_{(x,y)\sim\rho_\pi^{1:k}\times\rho_E^{1:k}}[\Omega_{d,\varepsilon}(r,x,y)] \Big), \tag{6}$$

where $\Omega_{d,\varepsilon}(r,x,y) = -\frac{1}{4\varepsilon}(r(y) - r(x) - d(x,y))^2$ regularizes the reward function in such a way that it decreases the objective if $r_\omega(\cdot)$ is not a Lipschitz (1) function.

As shown in previous works, Lipschitz continuity is crucial for Wasserstein loss in GANs (Arjovsky et al., 2017; Xiao et al., 2019). Recent research found that transformer-based discriminators' standard dot product self-attention layer may lack Lipschitz continuity, especially in discrete action spaces. To address this, we use two regularization techniques in our discriminator (Lee et al., 2022). For more information, refer to Appendix C.1.

4.2 Sequential autoregressive modeling generator

Similar to the designed discriminator above, we must also address the challenge of modeling complex dependencies among multiple agents for the generator (policy model). Agents often collaborate in various tasks, affecting each other's actions and behavior. Ignoring this can lead to suboptimal assessment and hinder collaboration. Unlike existing methods that match individual policies, we aim to create a generator that considers these interactions. We design a sequential autoregressive model inspired by MAT (Wen et al., 2022) to generate a joint policy distribution for multiple agents. The proposed framework's training pipeline follows classical GAIL (Song et al., 2018; Wang et al., 2023). For more details, refer to Appendix C.2.

5 Experiments

The fundamental insight of $\mathbf{MILD}^2$ revolves around a global dependency-enhanced framework for multi-agent imitation learning inspired by Thm. (3), as well as an encoder-decoder architecture

Table 1: Mean accumulated trajectory rewards with standard deviation across baselines on four benchmarks.

Benchmarks	Tasks	MAGAIL	CQL-MA	ICQ-MA	TD3-BC	OMAR	MILD^2(Ours.)
SMAC	3m	19.97±1.56	15.20±1.93	12.87±1.56	12.64±1.40	19.82±0.02	**20.00±0.00**
	3s5z	19.92±0.02	10.66±0.93	18.77±0.50	14.40±1.09	19.87±0.11	**20.00±0.02**
	6h vs 8z	16.90±0.12	7.91±0.14	10.13±0.35	8.56±0.18	16.33±0.15	**19.78±0.07**
	MMM2	5.01±0.03	5.01±0.03	11.32±0.87	3.09±0.24	10.99±0.22	**20.52±0.09**
Football	3 vs 1	4.88±0.04	4.18±0.47	2.85±0.58	1.42±0.33	4.55±0.34	**4.89±0.02**
	counterattack	4.62±0.08	0.36±0.02	2.21±0.44	0.27±0.06	1.14±0.18	**4.77±0.16**
	pass and shoot	3.92±0.39	1.39±0.71	3.11±0.62	1.68±0.09	2.72±0.58	**4.83±0.11**
Bi-DexHands	CatchOver2Underarm	6.70±0.10	15.65±1.01	3.53±0.17	10.13±1.15	16.85±1.21	**24.16±0.62**
	DoorOpenInward	12.47±23.34	189.74±41.71	-7.20±36.08	217.68±45.01	114.47±34.31	**395.98±0.39**
	DoorCloseOutward	503.32±0.12	839.48±11.29	215.365±0.08	41.84±5.16	818.76±2.43	**1016.89±0.13**
Multi-agent Mujoco	HalfCheetah 6×1	435.46±9.00	2189.50±959.35	3977.20±127.14	4123.60±146.41	4088.93±165.67	**4475.95±74.75**

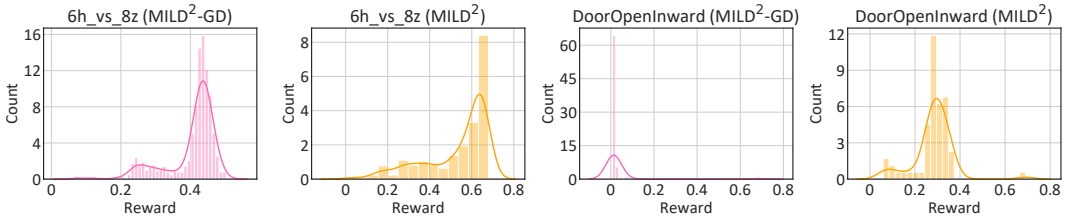

Figure 3: Distribution of rewards given by the discriminator during training. The data is collected on 6h_vs_8z (discrete actions) and DoorOpenInward (continuous actions) environments.

that provides an efficient implementation for modeling and matching the joint reward and policy distribution. In this section, we evaluate the performance of **MILD**^{2} on four benchmarks and compare them with state-of-the-art methods. Our experiments aim to answer two questions: (1) Does considering intricate dependencies among agents and modeling joint distributions contribute to multi-agent imitation learning models' rapid convergence and superior performance? (2) Does the discriminator architecture, fortified with global dependency, effectively augment reward and advantage variance, and alleviate the imbalance of training speed issue?

5.1 Experimental Setup

Benchmark datasets. We evaluated **MILD**^{2} using four benchmarks: StarCraftII Multi-Agent Challenge (SMAC) benchmark (Samvelyan et al., 2019), Google Research Football benchmark (Football) (Kurach et al., 2020), Bimanual Dexterous Hands Manipulation benchmark (Bi-DexHands) (Chen et al., 2022), and Multi-agent MuJoCo benchmark (Ma-Mujoco) (de Witt et al., 2020). We constructed several multi-agent offline datasets on these benchmarks by collecting 10,000 (for tasks in Bi-DexHands) and 100,000 (for tasks in others) transitions of expert policy from HAPPO (Kuba et al., 2022).

Baselines. We compare our method against five classical multi-agent offline RL methods, including MAGAIL (Song et al., 2018), the multi-agent version of CQL (CQL-MA) (Kumar et al., 2020), ICQ-MA (Yang et al., 2021), TD3-BC (Fujimoto & Gu, 2021), and OMAR (Pan et al., 2022). Following most baseline methods, each algorithm runs with five seeds, where the performance is evaluated 20 times every 50 episodes. We show experimental details in Appendix (D.1).

5.2 Main Results

Tab. (1) shows that **MILD**^{2} outperforms MAGAIL (baseline for independent distribution matching) and other advanced multi-agent imitation learning methods across four benchmarks. In discrete action space benchmarks (SMAC and Football), **MILD**^{2} consistently improves accumulated trajectory rewards by 0.15% to 81.27% across various difficulty settings. It also achieves a significant 33.33% to 300% win rate improvement on challenging SMAC tasks involving complex cooperation. In continuous action space benchmarks (Bi-DexHands and Ma-Mujoco), **MILD**^{2} exhibits an 8.54% to 81.91% performance enhancement. Fig. (8) visually demonstrates that our proposed framework achieves faster joint distribution matching of multi-agent policies while maintaining stability, even outperforming expert policies. Introducing global dependencies through the discriminator and gen-

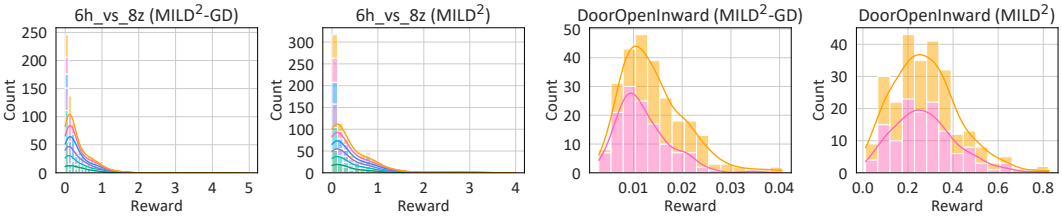

Figure 4: Statistical frequency histogram of individual rewards given by the discriminator. The data is collected on 6h_vs_8z (6 agents) and DoorOpenInward (2 agents) tasks at environmental step 1,900,000.

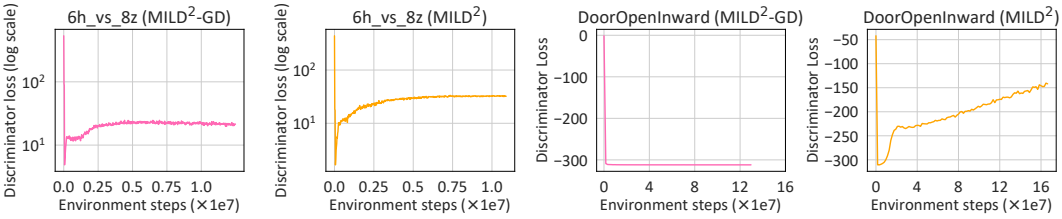

Figure 5: Discriminator loss curve comparison during training on 6h_vs_8z and DoorOpenInward environments.

erator, modeling joint reward and policy distribution, enables each agent to receive more plausible individual rewards and advantages. It leads to discovering higher-rewarding behaviors beyond those demonstrated in expert demonstrations. Overall, **MILD**^{2} significantly improves performance, especially in complex and challenging environments.

5.3 Ablation Studies

To demonstrate the effectiveness of the proposed algorithm, we conducted some ablation experiments to showcase the enhanced effects of capturing global dependencies among agents in the distribution matching-based multi-agent imitation learning framework.

Analysis on reward distribution. We compared the reward distributions of **MILD**^{2} and a variant called **MILD**^{2}-GD, which uses independent individual distribution matching and lacks global dependency modeling. Fig. (3) shows the discriminator's reward distributions during training for both algorithms. Fig. (3) and Fig. (4) illustrate that introducing global dependency modeling to **MILD**^{2} increases reward variance and raises the median reward. It supports the effectiveness of **MILD**^{2} in addressing the low reward variance issue, as hypothesized in our study.

Analysis on training loss. Moreover, **MILD**^{2} effectively counteracted the pattern of discriminator loss (D-loss), initially decreasing and subsequently reaching a plateau. Fig. (5) presents a representative dataset depicting the training progression of D-loss. All parameters except the algorithms employed remained consistent between the two training. Significantly, **MILD**^{2} successfully elevated the D-loss, which indicates that incorporating global dependencies through modeling facilitated a reduction in the problem of gradient vanishing after amplifying reward variance. See Appendix (D.2) for more ablation and robustness experiments.

6 Conclusion

In conclusion, this paper addresses the limitations of traditional independent multi-agent imitation learning frameworks in capturing complex dependencies among multiple agents. We highlight the importance of sequential modeling interdependencies and cooperative relationships among agents to enhance reward and advantage variances, and to facilitate model convergence. The paper proposes a Transformer-based framework for multi-agent imitation learning, leveraging the sequence modeling capabilities of the Transformer model to approximate the global distribution of reward function and policy accurately. Experimental results on cooperative benchmarks demonstrate the effectiveness of the proposed method, outperforming baseline algorithms in terms of convergence, stability, and robust adaptability. More other explicit agent dependency structures could explored in future work.

REFERENCES

Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *ICML*, volume 69, 2004.

Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, volume 70, pp. 214–223, 2017.

Michael Bain and Claude Sammut. A framework for behavioural cloning. In *Machine Intelligence*, pp. 103–129, 1995.

Nir Baram, Oron Anschel, Itai Caspi, and Shie Mannor. End-to-end differentiable adversarial imitation learning. In *ICML*, volume 70, pp. 390–399, 2017.

Peter W. Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, and Koray Kavukcuoglu. Interaction networks for learning about objects, relations and physics. In *NeurIPS*, pp. 4502–4510, 2016.

Raunak P. Bhattacharyya, Derek J. Phillips, Blake Wulfe, Jeremy Morton, Alex Kuefler, and Mykel J. Kochenderfer. Multi-agent imitation learning for driving simulation. In *IROS*, pp. 1534–1539, 2018.

Yu-Han Chang, Tracey Ho, and Leslie Pack Kaelbling. All learning is local: Multi-agent learning in global reward games. In *NeurIPS*, pp. 807–814, 2003.

Lili Chen, Kevin Lu, Aravind Rajeswaran, et al. Decision transformer: Reinforcement learning via sequence modeling. In *NeurIPS*, pp. 15084–15097, 2021.

Yuanpei Chen, Yaodong Yang, Tianhao Wu, et al. Towards human-level bimanual dexterous manipulation with reinforcement learning. In *NeurIPS*, 2022.

Christian Schröder de Witt, Bei Peng, Pierre-Alexandre Kamienny, et al. Deep multi-agent reinforcement learning for decentralized continuous cooperative control. *arXiv preprint arXiv:2003.06709*, 2020.

Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: pure attention loses rank doubly exponentially with depth. In *ICML*, volume 139, pp. 2793–2803, 2021.

Ishan Durugkar, Elad Liebman, and Peter Stone. Balancing individual preferences and shared objectives in multiagent reinforcement learning. In *IJCAI*, pp. 2505–2511, 2020.

Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *ICML*, volume 48, pp. 49–58, 2016.

Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248*, 2017.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In *NeurIPS*, pp. 20132–20145, 2021.

Ziwei Guan, Tengyu Xu, and Yingbin Liang. When will generative adversarial imitation learning algorithms attain global convergence. In *AISTATS*, volume 130, pp. 1117–1125, 2021.

Dylan Hadfield-Menell, Stuart Russell, Pieter Abbeel, and Anca D. Dragan. Cooperative inverse reinforcement learning. In *NeurIPS*, pp. 3909–3917, 2016.

Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart J. Russell, and Anca D. Dragan. Inverse reward design. In *NeurIPS*, pp. 6765–6774, 2017.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *NeurIPS*, pp. 4565–4573, 2016.

Yedid Hoshen. VAIN: attentional multi-agent predictive modeling. In *NeurIPS*, pp. 2701–2711, 2017.

Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Comput. Surv.*, 50(2):21:1–21:35, 2017.

Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. In *NeurIPS*, pp. 1273–1286, 2021.

Hyunjik Kim, George Papamakarios, and Andriy Mnih. The lipschitz constant of self-attention. In *ICML*, volume 139, pp. 5562–5571, 2021.

Thomas N. Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard S. Zemel. Neural relational inference for interacting systems. In *ICML*, volume 80, pp. 2693–2702, 2018.

Vijay R. Konda and John N. Tsitsiklis. Actor-critic algorithms. In *NeurIPS*, pp. 1008–1014, 1999.

Jakub Grudzien Kuba, Muning Wen, Linghui Meng, Shangding Gu, Haifeng Zhang, David Mguni, Jun Wang, and Yaodong Yang. Settling the variance of multi-agent policy gradients. In *NeurIPS*, pp. 13458–13470, 2021.

Jakub Grudzien Kuba, Ruiqing Chen, Muning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. Trust region policy optimisation in multi-agent reinforcement learning. In *ICLR*, 2022.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *NeurIPS*, 2020.

Karol Kurach, Anton Raichuk, Piotr Stanczyk, et al. Google research football: A novel reinforcement learning environment. In *AAAI*, pp. 4501–4510, 2020.

Hoang Minh Le, Yisong Yue, Peter Carr, and Patrick Lucey. Coordinated multi-agent imitation learning. In *ICML*, volume 70, pp. 1995–2003, 2017.

Kwonjoon Lee, Huiwen Chang, Lu Jiang, Han Zhang, Zhuowen Tu, and Ce Liu. Vitgan: Training gans with vision transformers. In *ICLR*, 2022.

Hepeng Li and Haibo He. Multi-agent trust region policy optimization. *arXiv preprint arXiv:2010.07916*, 2020.

Max Guangyu Li, Bo Jiang, Hao Zhu, Zhengping Che, and Yan Liu. Generative attention networks for multi-agent behavioral modeling. In *AAAI*, pp. 7195–7202, 2020.

Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *ICML*, pp. 157–163, 1994.

Minghuan Liu, Ming Zhou, Weinan Zhang, Yuzheng Zhuang, Jun Wang, Wulong Liu, and Yong Yu. Multi-agent interactions modeling with correlated policies. In *ICLR*, 2020.

Linghui Meng, Muning Wen, Chenyang Le, Xiyun Li, Dengpeng Xing, Weinan Zhang, Ying Wen, Haifeng Zhang, Jun Wang, Yaodong Yang, and Bo Xu. Offline pre-trained multi-agent decision transformer. *Mach. Intell. Res.*, 20(2):233–248, 2023.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Andrew Y. Ng and Stuart Russell. Algorithms for inverse reinforcement learning. In *ICML*, pp. 663–670, 2000.

Ling Pan, Longbo Huang, Tengyu Ma, and Huazhe Xu. Plan better amid conservatism: Offline multi-agent reinforcement learning with actor rectification. In *ICML*, volume 162, pp. 17221–17237, 2022.

David Radke, Kate Larson, and Tim Brecht. Exploring the benefits of teams in multiagent learning. In *IJCAI*, pp. 454–460, 2022.

Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *AISTATS*, volume 9, pp. 661–668, 2010.

Stéphane Ross, Geoffrey J. Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, volume 15, pp. 627–635, 2011.

Stuart Russell. Learning agents for uncertain environments (extended abstract). In *COLT*, pp. 101–103, 1998.

Mikayel Samvelyan, Tabish Rashid, Christian Schröder de Witt, et al. The starcraft multi-agent challenge. In *AAMAS*, pp. 2186–2188, 2019.

Stefan Schaal. Learning from demonstration. In *NeurIPS*, pp. 1040–1046, 1996.

John Schulman, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz. Trust region policy optimization. In *ICML*, volume 37, pp. 1889–1897, 2015.

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

Jiaming Song, Hongyu Ren, Dorsa Sadigh, and Stefano Ermon. Multi-agent generative adversarial imitation learning. In *NeurIPS*, pp. 7472–7483, 2018.

Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning multiagent communication with backpropagation. In *NeurIPS*, pp. 2244–2252, 2016.

Zheng Tian, Ying Wen, Zhichen Gong, Faiz Punakkath, Shihao Zou, and Jun Wang. A regularized opponent model with maximum entropy objective. In *IJCAI*, pp. 602–608, 2019.

Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. In *IJCAI*, pp. 4950–4957, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pp. 5998–6008, 2017.

Caroline Wang, Ishan Durugkar, Elad Liebman, and Peter Stone. Dm^2: Decentralized multi-agent reinforcement learning for distribution matching, 2023.

Hongwei Wang, Lantao Yu, Zhangjie Cao, and Stefano Ermon. Multi-agent imitation learning with copulas. In *ECML PKDD*, volume 12975, pp. 139–156, 2021.

Muning Wen, Jakub Grudzien Kuba, Runji Lin, Weinan Zhang, Ying Wen, Jun Wang, and Yaodong Yang. Multi-agent reinforcement learning is a sequence modeling problem. In *NeurIPS*, 2022.

Huang Xiao, Michael Herman, Jörg Wagner, Sebastian Ziesche, Jalal Etesami, and Thai Hong Linh. Wasserstein adversarial imitation learning. *arXiv preprint arXiv:1906.08113*, 2019.

Haoran Xu, Li Jiang, Jianxiong Li, Zhuoran Yang, Zhaoran Wang, Wai Kin Victor Chan, and Xianyuan Zhan. Offline RL with no OOD actions: In-sample learning via implicit value regularization. In *ICLR*, 2023.

Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang. Mean field multi-agent reinforcement learning. volume 80, pp. 5567–5576, 2018.

Yiqin Yang, Xiaoteng Ma, Chenghao Li, et al. Believe what you see: Implicit constraint approach for offline multi-agent reinforcement learning. In *NeurIPS*, pp. 10299–10312, 2021.

Lantao Yu, Jiaming Song, and Stefano Ermon. Multi-agent adversarial inverse reinforcement learning. In *ICML*, volume 97, pp. 7194–7201, 2019.

Won Joon Yun, Soohyun Park, Joongheon Kim, et al. Cooperative multi-agent deep reinforcement learning for reliable surveillance via autonomous multi-uav control. *arXiv preprint arXiv:2201.05843*, 2022.

Eric Zhan, Stephan Zheng, Yisong Yue, Long Sha, and Patrick Lucey. Generating multi-agent trajectories using programmatic weak supervision. In *ICLR*, 2019.

Yi-Feng Zhang, Fan-Ming Luo, and Yang Yu. Improve generated adversarial imitation learning with reward variance regularization. *Mach. Learn.*, 111(3):977–995, 2022.