Multi-Agent Imitation by Learning and Sampling from Factorized Soft Q-Function

Yi-Chen Li[1,2], **Zhongxiang Ling**[1,2], **Tao Jiang**[1,2,4], **Fuxiang Zhang**[3], **Pengyuan Wang**[1,2], **Lei Yuan**[1,2,4], **Zongzhang Zhang**[1,2], **Yang Yu**[1,2,4*]

[1] National Key Laboratory for Novel Software Technology, Nanjing University, China,
[2] School of Artificial Intelligence, Nanjing University, Nanjing, China,
[3] Nanyang Technological University, Singapore,
[4] Polixir Technologies, Nanjing, China,
liyc@lamda.nju.edu.cn, lingzx@smail.nju.edu.cn, fuxiang001@e.ntu.edu.sg
{wangpy, yuanl}@lamda.nju.edu.cn, zzzhang@nju.edu.cn, yuy@nju.edu.cn

Abstract

Learning from multi-agent expert demonstrations, known as Multi-Agent Imitation Learning (MAIL), provides a promising approach to sequential decision-making. However, existing MAIL methods including Behavior Cloning (BC) and Adversarial Imitation Learning (AIL) face significant challenges: BC suffers from the compounding error issue, while the very nature of adversarial optimization makes AIL prone to instability. In this work, we propose **M**ulti-**A**gent imitation by learning and sampling from **F**actor**I**zed **S**oft Q-function (MAFIS), a novel method that addresses these limitations for both online and offline MAIL settings. Built upon the single-agent IQ-Learn framework, MAFIS introduces the value decomposition network to factorize the imitation objective at agent level, thus enabling scalable training for multi-agent systems. Moreover, we observe that the soft Q-function implicitly defines the optimal policy as an energy-based model, from which we can sample actions via stochastic gradient Langevin dynamics. This allows us to estimate the gradient of the factorized optimization objective for continuous control tasks, avoiding the adversarial optimization between the soft Q-function and the policy required by prior work. By doing so, we obtain a tractable and *non-adversarial* objective for both discrete and continuous multi-agent control. Experiments on common benchmarks including the discrete control tasks StarCraft Multi-Agent Challenge v2 (SMACv2), Gold Miner, and Multi Particle Environments (MPE), as well as the continuous control task Multi-Agent MuJoCo (MaMuJoCo), demonstrate that MAFIS achieves superior performance compared with baselines. Our code is available at https://github.com/LAMDA-RL/MAFIS.

1 Introduction

Multi-Agent Imitation Learning (MAIL) focuses on learning policies from expert demonstrations. Compared to Multi-Agent Reinforcement Learning (MARL) that requires repeated and tedious design of reward functions [49], MAIL provides an efficient solution for optimal sequential decision-making, especially considering that collecting expert demonstrations is often easier and faster than designing reward functions [1]. Applications of MAIL over diverse domains such as driving simulation [4], unmanned aerial vehicles deployment [44] and robotics control [42] has shown its great potential.

A large body of previous works on MAIL can be categorized into two approaches: Behavioral Cloning (BC) [22, 50, 42] and Adversarial Imitation Learning (GAIL) [48, 19, 36]. BC reduces MAIL to a

*Corresponding Author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

supervised learning problem, whereas AIL trains a pair of generator and discriminator such that the discriminator can not distinguish behaviors of the generator from demonstrations. However, both approaches have their own limitations. Behavioral cloning suffers from compounding error, where small prediction mistakes accumulate over time as the model encounters states it was not trained on [33, 47]. AIL, due to the inherent nature of adversarial optimization, often suffers from training instability and sensitivity to hyper-parameters [25].

Recently, Garg et al. [17] propose a single-agent Imitation Learning (IL) framework, named as IQ-Learn. Although consistent with AIL's original modeling of IL, IQ-Learn reformulates IL as an optimization problem over the Q-function and the policy. Furthermore, Garg et al. [17] discover that the optimal policy can be expressed in closed form through the given Q-function. This leads to a tractable objective in discrete control tasks that requires optimizing only the Q-function. However, we cannot directly apply IQ-Learn to MAIL due to two key obstacles. First, as IQ-Learn is designed for single-agent tasks, applying it to MAIL by treating multiple agents as a single entity will lead to inefficient training and make it unsuitable for decentralized execution. Second, the closed-form solution for the policy requires calculating the `logsumexp` of Q values, which is computationally intractable in continuous control tasks. IQ-Learn addresses this by adversarially optimizing the policy and Q-function, which however has shown to be empirically unstable and may even fail [3]. A new MAIL method with strong generalization ability and enhanced stability is in high demand.

To deal with the aforementioned challenges, this paper proposes **M**ulti-**A**gent imitation by learning and sampling from **F**actor**I**zed **S**oft Q-function, abbreviated as MAFIS. We start by adapting IQ-Learn to MAIL tasks via replacing the single-agent policy and soft Q-function with the joint policy and global soft Q-function, respectively. Inspired by the value decomposition network [38, 31], we then consider representing the global soft Q-function as a weighted sum of individual soft Q-functions. With careful derivation, we find that the objective can now be factorized at the agent level, thus enabling scalable training and decentralized execution in multi-agent systems. However, it requires calculating the `logsumexp` of Q values over each agent's action space, which is intractable for continuous control tasks. Our key insight is that we can still estimate the gradient of the objective as long as we can sample actions from the optimal policy. Additionally, we observe that when given a soft Q-function Q, the optimal policy has a closed-form expression as an Energy-Based Model (EBM) [37] with $-Q$ being the energy function. Actions can be sampled from the EBM using stochastic gradient Langevin dynamics [45]. Built upon the above improvements, we successfully obtain MAFIS, a unified online and offline MAIL framework for both discrete and continuous control. To validate the effectiveness of MAFIS, we conduct extensive experiments on common benchmarks including the discrete control tasks StarCraft Multi-Agent Challenge v2 (SMACv2) [13], Gold Miner [15], and Multi Particle Environments (MPE) [27], as well as the continuous control task Multi-Agent MuJoCo (MaMuJoCo) [11], the result of which shows that MAFIS achieves superior performance compared with baseline methods.

2 Related Work

Multi-Agent Imitation Learning Due to its promise to turn a small expert dataset into a powerful decision making engine, Multi-Agent Imitation Learning (MAIL) has attracted broad attention recently. Le et al. [22] proposes to learn a latent probabilistic coordination model from demonstrations and then use BC to output a group of imitation polices conditioned on the learned embedding of coordination and local observations. Zhan et al. [50] generalizes this idea, using a hierarchical framework to generate the high-level intentions of each agent. Apart from BC, some studies model MAIL using the inverse reinforcement learning framework [53]. Song et al. [36] extends GAIL [21] to multi-agent scenarios where it learns a group of discriminators for every agent. Yu et al. [48] borrows the idea of Fu et al. [16] to learn reward functions that are highly correlated with the ground truth rewards. Moreover, Gruver et al. [19] conditions the learned reward functions on a learned embedding representing agents' behaviors. Recently, Bui et al. [7] propose MIFQ, which is also based on the IQ-Learn framework. However, the optimal soft value function in MIFQ is represented differently than in our method. Moreover, it is mainly for discrete control tasks. A detailed discussion on the distinction between MIFQ and ours is presented in Appendix B.

Value Decomposition Network To efficiently learn an optimal Q function for decentralized execution with the temporal difference loss [39], researchers in MARL have proposed many kinds of

value decomposition methods. Sunehag et al. [38] decomposes the centralized value function Q^{tot} into a sum of local value functions $Q^i, i \in \{1, 2, \cdots, n\}$. Rashid et al. [31] offer QMIX to represent Q^{tot} as a weighted sum of Q^i, where the weights are generated by a mixing network satisfying the individual global-max principle. Son et al. [35] transform the original Q^{tot} into a new, easily factorizable one with the same optimal actions in both functions. Wang et al. [43] take a duplex dueling network architecture which encodes the IGM principle into the neural network architecture to factorize the joint value function. In this paper, we use a mixing network similar to that of QMIX, but other Q-function decomposition methods can also be compatible with MAFIS.

Energy-Based Model Energy-based models (EBMs) are a class of probabilistic models that define a probability distribution $p(x)$ over data as a Gibbs distribution, $p(x) \propto \exp\{-E(x)\}$, where $E(x)$ is the energy function [23]. Unlike explicit probabilistic models, EBMs do not require exact computation of normalizing constants, but sampling and training often rely on approximate methods like Markov chain Monte Carlo [32] or Langevin dynamics [45]. EBMs are particularly flexible and capable of modeling complex dependencies between variables, and have shown promise in various domains, including vision [12], language [46], and reinforcement learning [20, 8]. In the field of IL, there are also works that represent the policy as an EBM. Specifically, Florence et al. [14] propose implicit behavioral cloning, which represents the policy as an energy-based model. They find that compared with explicitly modeling the policies, EBMs is better at modeling complex distributions, such as discontinuous or multi-modal ones. However, their approach is limited to offline learning and they only consider single agent tasks. To the best of our knowledge, our work is the first MAIL method that samples from and learns policies from the perspective of EBMs.

3 Preliminaries

To support subsequent analysis, this section will present the necessary background including the problem formulation and key notation definitions. Unless otherwise specified, we will use bold symbols throughout to represent joint variables across all agents.

3.1 Cooperative MARL

We model a fully cooperative multi-agent task as a Markov game [24], which can be defined by a tuple $\mathcal{M} = \langle \mathcal{I}, \mathcal{S}, \boldsymbol{\mathcal{A}}, P, \Omega, r, \gamma \rangle$. Here, $\mathcal{I} = \{1, 2, \cdots, n\}$ is the set of agents, $\mathcal{S}$ is the state space, and $\boldsymbol{\mathcal{A}} = \prod_{i=1}^{n} \mathcal{A}^i$ is the joint action space where $\mathcal{A}^i$ is the action space for agent i. At time step $t \in \mathbb{N}$, each agent $i \in \mathcal{I}$ chooses an action $a_t^i \in \mathcal{A}^i$ at the global state $s_t \in \mathcal{S}$, together forming a joint action $\boldsymbol{a}_t = (a_t^1, a_t^2, \cdots, a_t^n)$. By executing action $\boldsymbol{a}_t$ in the environment, the agents receive a shared reward $r(s_t, \boldsymbol{a}_t)$ and the environment transitions to the next state $s_{t+1} \sim P(\cdot|s_t, \boldsymbol{a}_t)$. The goal of the agents is to find an optimal joint policy $\boldsymbol{\pi}$ that maximizes the expected discounted return $\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r(s_t, \boldsymbol{a}_t)]$, where $\gamma \in [0, 1)$ is a discount factor and $\boldsymbol{a}_t \sim \boldsymbol{\pi}(\cdot|s_t)$. Since independently updating each individual policy often results in poor convergence [10], the *Centralized Training with Decentralized Execution* (CTDE) paradigm has been predominantly adopted in current MARL research. In CTDE, we assumes that the global states and the actions as well as policies of teammates are accessible during the training phase.

3.2 Imitation Learning

In real-world tasks, defining the reward function r can be difficult and tedious. By contrast, obtaining expert demonstrations is relatively easier [1]. Imitation Learning (IL) focuses on learning an optimal policy from expert demonstrations (trajectories generated by policy π_E), without needing to know the underlying reward function r. Two main approaches are commonly used in IL: Behavioral Cloning (BC) [29] and Inverse Reinforcement Learning (IRL) [2]. BC casts IL as a supervised learning problem over state-action pairs [29]. While simple, it suffers from the compounding error issue [33]. On the other hand, IRL tries to first recover the reward function from the expert demonstrations and then extracts an optimal policy with that reward function via reinforcement learning [39]. It has been theoretically proven that IRL can mitigate compounding errors, thereby better imitating the expert [47]. Following existing IRL work [21], we consider the maximum causal entropy IRL framework [52], the objective of which is shown as below:

$$\max_{\pi} \min_{r} \left(H(\pi) + \mathbb{E}_{\pi}[r(s, a)]\right) - \mathbb{E}_{\pi_E}[r(s, a)], \tag{1}$$

where $H(\pi) \triangleq \mathbb{E}_\pi[-\log \pi(a|s)]$ is the γ-discounted causal entropy [5] of the policy π.

3.3 IQ-Learn Framework

For a fixed policy π, let Q^π be its *soft Q-function* defined as

$$Q^\pi(s,a) \triangleq \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right] + \alpha H(\pi), \tag{2}$$

Here, $\alpha > 0$ is called the *entropy weight*, which determines the relative weight of return and entropy. Garg et al. [17] find that there is an one-to-one correspondence between the sets of feasible r and Q^π, thus converting Equation (1) into the following new objective:

$$\max_Q \min_\pi \mathcal{J}(\pi,Q) \triangleq -(1-\gamma)\mathbb{E}_{s_0\sim\rho_0}[V^\pi(s_0)] + \mathbb{E}_{\mathcal{D}_E}\big[\phi\big(Q(s,a) - \gamma\mathbb{E}_{s'\sim P(\cdot|s,a)}V^\pi(s')\big)\big], \tag{3}$$

where $V^\pi(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}[Q(s,a) - \alpha\log\pi(a|s)]$ is called the *soft value function* [20], and ρ_0 is the initial state distribution. The concave function $\phi : \mathbb{R} \to \mathbb{R}$ serves as a regularizer for the soft Q-function. When ϕ is chosen as the identity function, i.e., $\phi(x) = x$, Garg et al. [17] illustrate that the objective shown as in Equation (3) is essentially attempting to minimize the total variation distance [18] between the state-action distributions of the expert and the imitator.

4 Our Method

We now describe our method, **M**ulti-**A**gent imitation by learning and sampling from **F**actor**I**zed **S**oft Q-function (MAFIS). We will demonstrate how the single-agent IQ-Learn framework can be extended to both continuous and discrete multi-agent control tasks, enabling stable and efficient training while supporting decentralized execution. Algorithm 1 summarizes the pseudo code of MAFIS.

4.1 Soft Q-Function Factorization

Given the expert demonstrations $\mathcal{D}_E$, a straightforward approach for MAIL is to cast it as a single-agent learning problem. That is, treating the multiple agents as a single entity and learning the joint policy $\boldsymbol{\pi}$ via Equation (3). However, this does not support decentralized execution. Drawing inspiration from *value decomposition networks* [38, 31], we instead consider representing the joint soft Q-function $Q^{tot}(\boldsymbol{\tau}_t, \boldsymbol{a}_t)$ as a weighted sum of individual Q-functions $\{Q^1(\tau_t^1, a_t^1), Q^2(\tau_t^2, a_t^2), \cdots, Q^n(\tau_t^n, a_t^n)\}$, i.e.,

$$Q^{tot}(\boldsymbol{\tau}_t, \boldsymbol{a}_t) = \sum_{i=1}^{n} k^i(s_t) Q^i(\tau_t^i, a_t^i), \tag{4}$$

where $k : \mathcal{S} \to \mathbb{R}^n_{\geq 0}$ is the *mixing network*. For agent $i \in \mathcal{I}$ at time step $t \in \mathbb{N}$, o_t^i is its local observation and $\tau_t^i = (o_0^i, a_0^i, o_1^i, a_1^i, \cdots, o_{t-1}^i, a_{t-1}^i, o_t^i) \in \mathcal{T}^i$ is its observation-action history. An illustration of the joint Q-function's network architecture is in Appendix C.1. With Q^{tot} being represented as in Equation (4), we derive the following result.

Proposition 4.1. *For a fixed joint Q-function $Q^{tot}(\boldsymbol{\tau}, \boldsymbol{a})$ and $\phi(x) = x$, the joint policy $\boldsymbol{\pi}_{Q^{tot}}(\boldsymbol{a}|\boldsymbol{\tau})$ that minimizes $\mathcal{J}(\boldsymbol{\pi}, Q^{tot})$ satisfies*

$$\boldsymbol{\pi}_{Q^{tot}}(\boldsymbol{a}|\boldsymbol{\tau}) = \prod_{i=1}^{n} \pi_{Q^i}(a^i|\tau^i), \forall \boldsymbol{\tau} \in \mathcal{T}, \boldsymbol{a} \in \boldsymbol{\mathcal{A}}, \tag{5}$$

where

$$\pi_{Q^i}(a^i|\tau^i) = \frac{1}{Z_{\tau^i}} \exp\left\{\frac{1}{\alpha} k^i(s) Q^i(\tau^i, a^i)\right\} \tag{6}$$

with $Z_{\tau^i} = \sum_{a\in\mathcal{A}^i} \exp\{\frac{1}{\alpha}k^i(s)Q^i(\tau^i,a)\}$. Thus, we have

$$\max_{Q^{tot}} \min_{\boldsymbol{\pi}} \mathcal{J}(\boldsymbol{\pi}, Q^{tot}) = \max_{Q^{tot}} \mathcal{J}(\boldsymbol{\pi}_{Q^{tot}}, Q^{tot}).$$

Moreover, $\mathcal{J}(\boldsymbol{\pi}_{Q^{tot}}, Q^{tot})$ can be further reduced as

$$\mathcal{J}(\boldsymbol{\pi}_{Q^{tot}}, Q^{tot}) = \sum_{i=1}^{n} -\alpha(1-\gamma)\mathbb{E}_{s_0\sim\rho_0}\left[\log Z_{\tau_0^i}\right] + \mathbb{E}_{(\boldsymbol{\tau},\boldsymbol{a},\boldsymbol{\tau}')\sim\mathcal{D}_E}\left[k^i(s)Q^i(\tau^i,a^i) - \gamma\alpha\log Z_{\tau^{i,\prime}}\right]. \tag{7}$$

Due to space limitation, we defer the proof to Appendix A.1. Proposition 4.1 tells us that the adversarial training of Q^{tot} and $\boldsymbol{\pi}$ can be converted into a non-adversarial one by learning Q^{tot} based on Equation (7), thereby avoiding the potential instability associated with adversarial training [25]. Another interesting observation is that although $\log Z_{\boldsymbol{\tau}}$ is originally defined in the joint action space whose computation grows exponentially with the number of agents n, it can be factorized thanks to the joint soft Q-function decomposition, thereby allowing scalable and efficient training.

A recent work, MIFQ [7], also considers extending IQ-Learn to multi-agent tasks. They propose to decompose the joint soft Q-function as in Equation (4). Proposition 4.4 of MIFQ appears similar to our above Proposition 4.1. However, they represent the individual optimal policy $\pi_{Q^i}(a^i|\tau^i)$ as $\pi_{Q^i}(a^i|\tau^i) = \frac{\exp\{Q^i(\tau^i,a^i)\}}{\sum_{a^i\in\mathcal{A}^i}\exp\{Q^i(\tau^i,a^i)\}}$, while we obtain that $\pi_{Q^i}(a^i|\tau^i) = \frac{\exp\{\frac{1}{\alpha}k^i(s)Q^i(\tau^i,a^i)\}}{\sum_{a\in\mathcal{A}^i}\exp\{\frac{1}{\alpha}k^i(s)Q^i(\tau^i,a)\}}$. Moreover, given the optimal policy, the soft value function $V^*(s)$ in Bui et al. [7] is $V^*(s) = \sum_{i=1}^{n} k^i(s)\log\sum_{a^i\in\mathcal{A}^i}\exp\{(Q^i(\tau^i,a^i)\}$, whereas we derive in Appendix A.1 that $V^*(s) = \sum_{i=1}^{n}\log\sum_{a^i\in\mathcal{A}^i}\exp\{k^i(s)Q^i(\tau^i,a^i)\}$. We additionally present a more detailed discussion on the distinctions between MIFQ [7] and ours in Appendix B.

Although Equation (7) converts the adversarial objective in Equation (3) into a non-adversarial one, two obstacles remain. First, computing $Z_{\tau^i} = \sum_{a\in\mathcal{A}^i}\exp\{\frac{1}{\alpha}k^i(s)Q^i(\tau^i,a)\}$ requires summing over the entire action space, which is intractable for continuous control tasks. Second, the individual optimal policy π_{Q^i} requires access to the global state to compute $k^i(s)$ as shown in Equation (6), which can be inaccessible during execution. We will delve into them in the next part.

4.2 Optimal Policy as Sampling from Soft Q-Function

Because $\log Z_{\tau^i}$ in Equation (7) is computationally intractable for continuous control tasks, a straightforward approach is to directly imitate the expert via Equation (3). That is, adversarially train the joint policy $\boldsymbol{\pi}$ and joint Q-function Q^{tot}, where $V^{\boldsymbol{\pi}}(\boldsymbol{\tau}) = \mathbb{E}_{\boldsymbol{a}\sim\boldsymbol{\pi}(\cdot|\boldsymbol{\tau})}[Q^{tot}(\boldsymbol{\tau},\boldsymbol{a}) - \alpha\log\boldsymbol{\pi}(\boldsymbol{a}|\boldsymbol{\tau})]$ is approximated by sampling actions from the current policy π. However, experiments on single-agent tasks have shown that doing so can lead to noticeable instabilities and even failures [3].

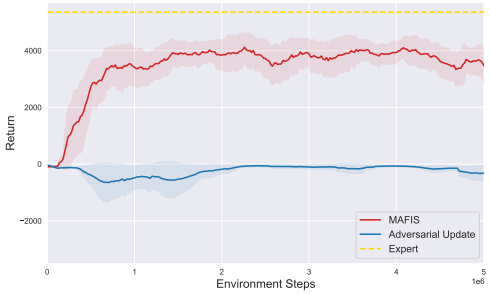


Figure 1: Performance comparison of MAFIS and Adversarial Update on the `Ant(2x4)` task from the MaMuJoCo benchmark.

We conducted experiments on the `Ant(2x4)` task from the MaMuJoCo benchmark [11] to validate this idea. Specifically, we maintain an individual policy π^i for each agent $i\in\mathcal{I}$, thus the joint policy $\boldsymbol{\pi}(\boldsymbol{a}|\boldsymbol{\tau}) = \prod_{i=1}^{n}\pi^i(a^i|\tau^i)$. The architecture of the Q-function remains similar to that of QMIX [31]. However, since the tasks of MaMuJoCo have continuous action spaces, we modify each individual Q-function to take the local history-action pair (instead of the local history only as in QMIX) as input and output the corresponding Q-value. We term this approach *Adversarial Update*. The empirical results presented in Figure 1 (averaged across five random seeds) reveal that the Adversarial Update fails to learn an effective imitation policy. New approaches to better handle imitation learning tasks with continuous action spaces are needed.

Our key insight is that computing $\log Z_{\tau^i}$ can be bypassed as long as we can sample actions from the optimal individual policy π_{Q^i}. Let θ denote the learnable parameters of the joint Q-function Q^{tot}, which includes the mixing network module k and the individual Q-function modules $\{Q^1, Q^2, \cdots, Q^n\}$. The following proposition holds.

Algorithm 1 MAFIS

Initialize the joint Q-function Q^{tot}, expert demonstrations $\mathcal{D}_E$, and total training steps T.
(Online only) Empty replay buffer $\mathcal{B}$.
(Continuous control only) Langevin steps K and step size ϵ, N samples for each estimation.
for step $t = 1$ to T **do**
  (Online only) Sample transitions in the environment and store them in the buffer $\mathcal{B}$.
  if Discrete control **then**
    Update Q^{tot} via Equation (7).
  else
    Update Q^{tot} via Equation (8).
  end if
end for

Proposition 4.2. *The gradient of $\mathcal{J}(\boldsymbol{\pi}_{Q^{tot}}, Q^{tot})$ with respect to θ is equal to*

$$\begin{aligned}\nabla_\theta \mathcal{J}\left(\boldsymbol{\pi}_{Q^{tot}}, Q^{tot}\right) = \sum_{i=1}^{n} -(1-\gamma)\mathbb{E}_{s_0\sim\rho_0, a_0^i\sim\pi_{Q^i}(\cdot|\tau_0^i)}\left[\nabla_\theta\left(k^i(s)Q^i(\tau_0^i, a_0^i)\right)\right] \\ + \mathbb{E}_{(\boldsymbol{\tau},\boldsymbol{a},\boldsymbol{\tau}')\sim\mathcal{D}_E}\Big[\nabla_\theta\left(k^i(s)Q^i(\tau^i, a^i)\right) - \gamma\mathbb{E}_{a^{i,\prime}\sim\pi_{Q^i}(\cdot|\tau^{i,\prime})}\left[\nabla_\theta\left(k^i(s')Q^i(\tau^{i,\prime}, a^{i,\prime})\right)\right]\Big].\end{aligned} \tag{8}$$

The proof is in Appendix A.2. Proposition 4.2 tells us that to update θ by performing gradient ascent [51] on $\mathcal{J}(\boldsymbol{\pi}_{Q^{tot}}, Q^{tot})$, all we need is to sample a batch of actions from $\boldsymbol{\pi}$. But how? Given a data point $\boldsymbol{x}$, an EBM defines its probability density $p(\boldsymbol{x})$ as $p(\boldsymbol{x}) = \frac{\exp\{-E(\boldsymbol{x})\}}{\sum_{\boldsymbol{x}}\exp\{-E(\boldsymbol{x})\}}$, where $E(\cdot)$ is called the *energy function* [37]. We observe that the policy $\pi_{Q^i}(\cdot|\tau)$ is inherently an Energy-Based Model (EBM) with $-\frac{1}{\alpha}k^i(s)Q^i(\tau,\cdot)$ being the energy function. Thanks to Stochastic Gradient Langevin dynamics (SGLD) [45], we can iteratively refine samples K times via the following rule:

$$a^{i,k+1} \leftarrow a^{i,k} + \frac{\epsilon^2}{2\alpha}\nabla_{a^{i,k}}k^i(s)Q^i(\tau, a^{i,k}) + \epsilon\omega,$$

which has been proven that $a^{i,K} \sim \pi_{Q^i}(\cdot|\tau)$ under certain conditions [54]. Here, $\epsilon > 0$ is the step size, $\omega \sim \mathcal{N}(0, \sigma^2)$ is the Gaussian noise with variance σ^2 and $a^{i,1} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$. We also conducted experiments on the same `Ant(2x4)` task to validate the effectiveness of this approach, where the expectation over π_{Q^i} in Equation (8) is estimated by parallelly sampling N actions[2]. As shown in Figure 1, our method MAFIS significantly outperforms Adversarial Update.

Decentralized Execution As demonstrated in Proposition 4.1, each agent $i \in \mathcal{I}$ must sample its own actions during the decentralized execution phase according to π_{Q^i}. However, π_{Q^i} explicitly depends on the global state s, which is inaccessible during decentralized execution in partially observable environments. To resolve this incompatibility, we observe that $k^i(s) > 0, \forall s \in \mathcal{S}$. This means that the original policy π_{Q^i} and its decentralized counterpart $\tilde{\pi}_{Q^i}(a|\tau)$ share identical maximizers. Here, $\tilde{\pi}_{Q^i}(a|\tau)$ is constructed using only local action-observation history τ and is defined as:

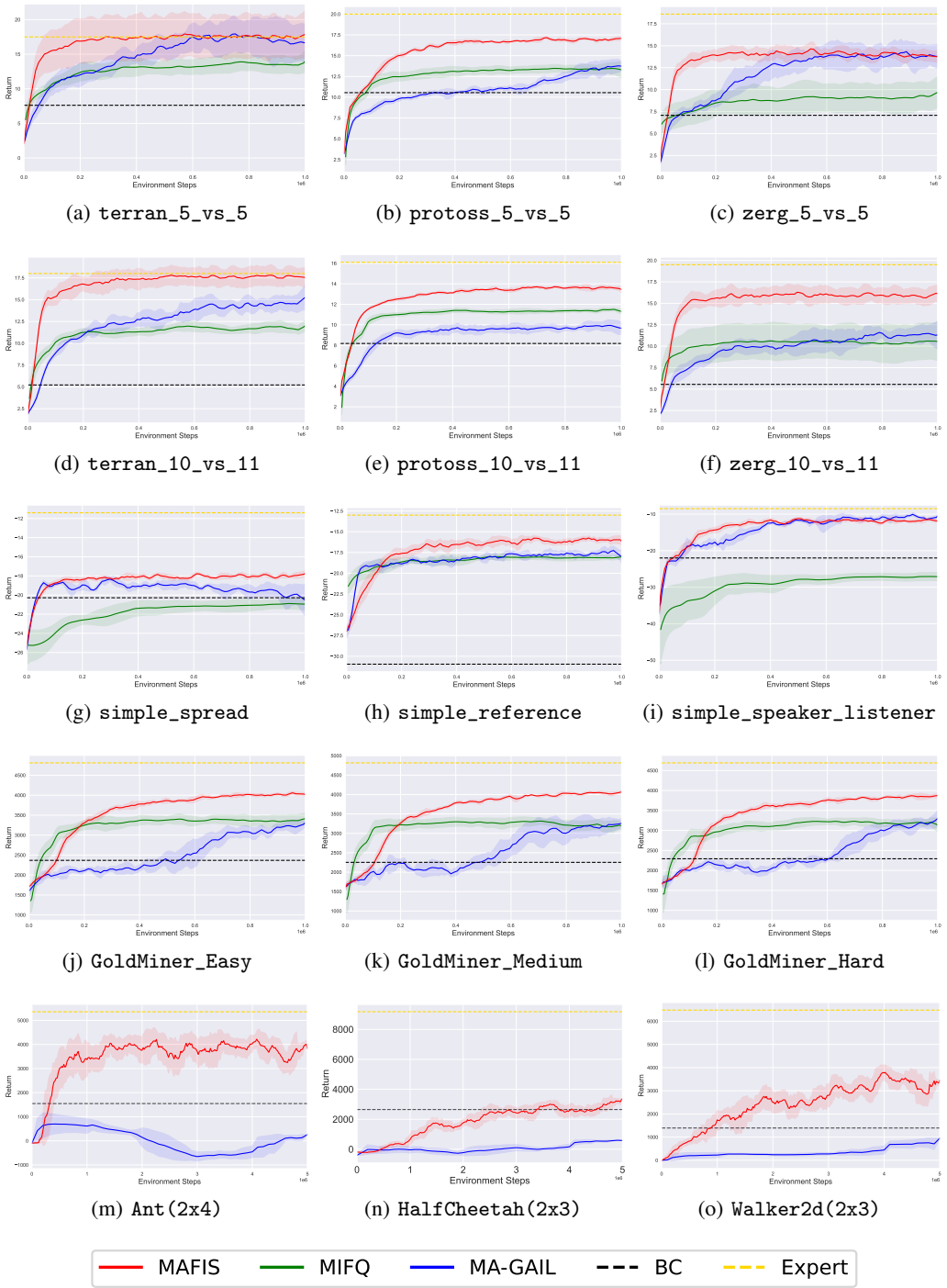
$$\tilde{\pi}_{Q^i}(a|\tau) \triangleq \frac{\exp\left\{\frac{1}{\alpha}Q^i(\tau, a)\right\}}{\sum_{\tilde{a}\in\mathcal{A}^i}\exp\left\{\frac{1}{\alpha}Q^i(\tau, \tilde{a})\right\}}. \tag{9}$$

Furthermore, to mitigate the stochasticity introduced by sampling, only the action with the highest Q-value among the N sampled actions from $\tilde{\pi}_{Q^i}(a|\tau)$ will be selected for execution. By adopting $\tilde{\pi}_{Q^i}(a|\tau)$ for action sampling, we successfully eliminate the need for $k^i(s)$ in the decentralized execution phase while preserving the optimality of the selected action.

5 Experiments

In this section, we will validate the effectiveness of MAFIS through extensive experiments. First, we will describe the experimental setup, including details of the baselines, benchmarks, and expert

[2]We set $N = 20$ in practice, and more hyper-parameter settings are presented in Appendix C.3.



(a) `terran_5_vs_5` (b) `protoss_5_vs_5` (c) `zerg_5_vs_5`

(d) `terran_10_vs_11` (e) `protoss_10_vs_11` (f) `zerg_10_vs_11`

(g) `simple_spread` (h) `simple_reference` (i) `simple_speaker_listener`

(j) `GoldMiner_Easy` (k) `GoldMiner_Medium` (l) `GoldMiner_Hard`

(m) `Ant(2x4)` (n) `HalfCheetah(2x3)` (o) `Walker2d(2x3)`

Figure 2: Online results in SMACv2, MPE, Gold Miner and MaMuJoCo.

demonstration datasets. Then, we will compare the performance of MAFIS against the baselines on several benchmarks. Finally, a sensitivity analysis is presented. Due to space limitations, more implementation details of MAFIS are deferred to Appendix C.

Baselines The following methods serve as baselines: (1) *Behavioral Clong (BC)* [29, 50], an offline method that casts MAIL as a supervised learning method to simply maximize the probability of the

Table 1: Offline results in SMACv2, MPE, Gold Miner, and MaMuJoCo. Due to space limitation, we abbreviate `simple_speaker_listener` as `simple_sl`. In addition, to better illustrate MAFIS's performance during online learning and to demonstrate the benefit of online samples for MAIL, we have also included the results of MAFIS's online imitation in the last column.

		Expert	BC	MIFQ	MAFIS	MAFIS (online)
SMACv2	`terran_5_vs_5`	17.55	7.64	15.57	**18.35**	18.63
	`protoss_5_vs_5`	20.08	10.51	14.74	**15.11**	16.98
	`zerg_5_vs_5`	18.62	7.10	11.70	**13.28**	13.70
	`terran_10_vs_11`	18.05	5.21	12.27	**16.27**	17.85
	`protoss_10_vs_11`	16.10	8.16	11.70	**13.21**	13.60
	`zerg_10_vs_11`	19.53	5.54	10.72	**16.08**	15.64
MPE	`simple_spread`	-11.44	-20.27	-39.95	**-18.77**	-17.84
	`simple_reference`	-12.98	-30.82	-18.74	**-17.80**	-15.87
	`simple_sl`	-8.50	-21.92	-29.40	**-13.49**	-11.60
Gold Miner	`GoldMiner_Easy`	4810.23	2395.90	3252.17	**4004.02**	4046.05
	`GoldMiner_Medium`	4815.32	2294.54	3193.83	**3978.82**	4038.11
	`GoldMiner_Hard`	4691.97	2309.02	3229.75	**3850.69**	3877.43
MaMuJoCo	`Ant(2x4)`	5687.83	1547.32	-	**3694.13**	3790.63
	`HalfCheetah(2x3)`	9187.68	2333.69	-	**2446.51**	3114.38
	`Walker2d(2x3)`	6481.51	1743.18	-	**3679.56**	3685.90

expert's action. (2) *Multi-Agent Generative Adversarial Imitation Learning (MA-GAIL)* [36], an online MAIL method that adversarially trains a discriminator and generator where the discriminator learns to recognize whether a state-action pair comes from the expert demonstrations, while the generator maximizes the reward given by the discriminator via reinforcement learning [39]. For easier debugging and compatibility with the latest Python packages, we re-implement MA-GAIL in PyTorch [28] based on the author's open-source code[3]. (3) *Multi-Agent Inverse Factorized soft Q-learning (MIFQ)* [7], which extends the single-agent IQ-Learn framework to multi-agent setting by learning a factorized soft Q-function and state value function. In our experiments, we use the official code provided by the authors[4]. Among all the baselines, MIFQ is applicable to discrete control tasks whereas BC and MA-GAIL can be applied to both discrete and continuous control tasks.

Benchmarks To validate the effectiveness of MAFIS, we follow Bui et al. [7] to select three discrete control benchmarks that includes: (1) *StarCraft Multi-Agent Challenge v2 (SMACv2)* [13], which focuses on decentralized micromanagement challenges in the game of StarCraft II. Based on the types of the races, SMCAv2 divides scenarios from StarCraft II into three groups including Protoss, Terran or Zerg. We choose `{terran, protoss, zerg}_5_vs_5` and `{terran, protoss, zerg}_10_vs_11` as test beds, where `5_vs_5` and `10_vs_11` denote the number of allies versus enemies. (2) *Multi Particle Environments (MPE)* [27] from the PettingZoo library [40], a set of communication oriented environment where particle agents can (sometimes) move, communicate, see each other, push each other around, and interact with fixed landmarks. We select three tasks from MPE including (i) `simple_spread` where three agents learn to cover all the landmarks while avoiding collisions and (ii) `simple_reference` where two agents learn to get closer to their target landmark, and (iii) `simple_speaker_listener` which is similar to `simple_reference`, except that one agent is the speaker and can speak but cannot move, while the other agent is the listener (cannot speak, but must navigate to correct landmark). (3) *Gold Miner* [15], adapted from the Reinforcement Learning Competition hosted by FPT-Software. In this game, two teams each with two members compete to mine the gold, where the one that mines more gold wins the game. We consider three tasks including `GoldMiner_{Easy, Medium, Hard}`. Apart from the aforementioned three benchmarks, we additionally choose a continuous control benchmark: (4) *Multi-agent MuJoCo (MaMuJoCo)* [11], a collection of multi agent factorizations of the Multi-Joint dynamics with Contact (MuJoCo) environments from Gym [6]. We choose `Ant(2x4)`, `HalfCheetah(2x3)`

[3] https://github.com/ermongroup/multiagent-gail/

[4] https://openreview.net/attachment?id=xrbgXJomJp&name=supplementary_material

Figure 3: Sensitivity to N in `Ant(2x4)`.

Figure 4: Sensitivity to α in `Walker2d(2x3)`.

and `Walker2d(2x3)` to compare the performance of all the algorithms. Here, `2x4` means that the task involves two agents, each with an action space of dimension 4.

Expert Demonstrations For discrete control tasks, the experts are trained via QMIX [31] until convergence using the ground truth reward functions, whereas for continuous control tasks, we learn the experts using HASAC [26]. Then, we utilize these experts to collect demonstrations in their corresponding environments. Specifically, for each discrete control task, we collect 100 expert trajectories, while for each continuous control task, we collect 20 expert trajectories. For fair comparison, MAFIS and all baselines utilize the same expert demonstrations for imitation.

5.1 Results

Online Learning The online learning results of different MAIL algorithms (averaged across 5 random seeds) are shown in Figure 2. As BC is an offline algorithm, we plot its results using a dashed line after training it to convergence. From Figure 2, we can see that our method MAFIS consistently outperforms all the baselines. On the other hand, although the baseline MA-GAIL achieves results similar to MAFIS on some tasks, such as `zerg_5_vs_5` and `simple_speaker_listener`, it performs significantly worse than MAFIS on more challenging tasks. In continuous control tasks, it even underperforms BC. We suspect that this may be due to the instability caused by the adversarial optimization in MA-GAIL. MAFIS achieves consistently better performance than MIFQ, which we hypothesize stems from its appropriate representation of the optimal soft value function.

Offline Learning For each task, under each seed, we train each algorithm using batch stochastic gradient ascent [51] for 3×10^6 steps. After training, we use the final checkpoint to roll out 10 trajectories in the environment and take the average return of these trajectories as the evaluation result for that seed. To mitigate the impact of randomness, we randomly select 5 seeds and compute the mean and variance of the evaluation results across them as the final reported outcome. The offline learning performance of different algorithms is shown in Table 1, where we bold the results with the highest mean. From Table 1 we can see that our method consistently outperforms the baselines.

5.2 Sensitivity Analysis

In this section, we will perform a sensitivity analysis of key hyper-parameters in our method. By comparing MAFIS's performance under different hyper-parameter configurations, we aim to understand their impacts on MAFIS and provide insights for practical hyper-parameter tuning. Full sensitivity analysis results are presented in Appendix D.

Sensitivity to the Hyper-parameter N The hyper-parameter N represents the number of samples drawn from π_{Q^i} to estimate $\mathbb{E}_{a^i \sim \pi_{Q^i}(\cdot|\tau^i)}\left[\nabla_\theta\left(k^i(s)Q^i(\tau^i, a^i)\right)\right]$ when computing Equation (8). Theoretically, a larger N leads to a more accurate estimation of the term; however, it also incurs a higher computational cost. We set $N = 1, 5, 10, 20, 30$ and evaluate the performance of MAFIS on `Ant(2x4)`, as shown in Figure 3. From Figure 3, we observe that as N increases, MAFIS's performance improves accordingly. To balance performance and computation cost, we set $N = 20$ for all benchmark experiments.

Sensitivity to the Hyper-parameter α The hyper-parameter α controls the sharpness of the policy distribution. Specifically, the larger the value of α, the closer the resulting policy distribution approaches a random distribution. Since $-\frac{1}{\alpha}k^i(s)Q^i(\tau^i, a^i)$ serves as the energy function for SGLD, a larger α will make the sampling process more exploratory. We set $\alpha = 0.2, 0.5, 1$, and 1.5 and evaluate MAFIS's performance on the `Walker2d(2x3)` task. The result is shown in Figure 4, from which we can see that MAFIS performs similarly when $\alpha = 0.2, 0.5, 1$, but experiences a significant performance drop when $\alpha = 1.5$. We believe this is reasonable because, as mentioned earlier, a larger α makes the sampling process more random, which may lead to a greater approximation error for π_{Q^i} given a limited number of Langevin steps and samples.

6 Conclusion

We introduce **MAFIS**, a novel Multi-Agent Imitation Learning (MAIL) framework that enhances scalability, stability, and efficiency. By factorizing the soft Q-function, MAFIS enables decentralized execution and effective training in multi-agent settings. Additionally, it bypasses policy learning in continuous control tasks through energy-based sampling. Extensive experiments on SMACv2, Gold Miner, MPE, and MaMuJoCo demonstrate that MAFIS achieves state-of-the-art performance, making it a promising approach for multi-agent imitation learning.

Limitation and Future Work Our algorithm achieves state-of-the-art performance but still lags behind expert demonstrations, especially in continuous control tasks. Future work will focus on improving imitation efficiency and handling complex action spaces to bridge this gap.

Acknowledgments and Disclosure of Funding

This work is supported by NSFC (62495093) and Jiangsu Science Foundation (BK20243039), the National Science Foundation of China (62495093,62506159, U24A20324), the Natural Science Foundation of Jiangsu (BK20241199, BK20243039), and the AI & AI for Science Project of Nanjing University. The authors would like to extend their appreciation to Yuhang Ran, Chenghe Wang and Feng Chen for their detailed discussions on the implementation details, and anonymous reviewers for providing valuable comments during the reviewing process.

References

[1] Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2004.

[2] Stephen C. Adams, Tyler Cody, and Peter A. Beling. A survey of inverse reinforcement learning. *Artificial Intelligence Review*, 55(6):4307–4346, 2022.

[3] Firas Al-Hafez, Davide Tateo, Oleg Arenz, Guoping Zhao, and Jan Peters. LS-IQ: Implicit reward regularization for inverse reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2023.

[4] Raunak P Bhattacharyya, Derek J Phillips, Blake Wulfe, Jeremy Morton, Alex Kuefler, and Mykel J Kochenderfer. Multi-agent imitation learning for driving simulation. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 1534–1539, 2018.

[5] Michael Bloem and Nicholas Bambos. Infinite time horizon maximum causal entropy inverse reinforcement learning. In *IEEE Conference on Decision and Control (CDC)*, pages 4911–4916. IEEE, 2014.

[6] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. *arXiv preprint arXiv:1606.01540*, 2016.

[7] The Viet Bui, Tien Mai, and Thanh Hong Nguyen. Inverse factorized soft Q-learning for cooperative multi-agent imitation learning. In *Advances in Neural Information Processing System (NeurIPS)*, 2024.

[8] Ruifeng Chen, Chengxing Jia, Zefang Huang, Tian-Shuo Liu, Xu-Hui Liu, and Yang Yu. Offline transition modeling via contrastive energy learning. In *International Conference on Machine Learning (ICML)*, 2024.

[9] KyungHyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[10] Caroline Claus and Craig Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 746–752, 1998.

[11] Rodrigo de Lazcano, Kallinteris Andreas, Jun Jet Tai, Seungjae Ryan Lee, and Jordan Terry. Gymnasium robotics, 2024. URL http://github.com/Farama-Foundation/Gymnasium-Robotics.

[12] Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3603–3613, 2019.

[13] Benjamin Ellis, Jonathan Cook, Skander Moalla, Mikayel Samvelyan, Mingfei Sun, Anuj Mahajan, Jakob N. Foerster, and Shimon Whiteson. SMACv2: An improved benchmark for cooperative multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[14] Pete Florence, Corey Lynch, Andy Zeng, Oscar A. Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Conference on Robot Learning (CoRL)*, pages 158–168, 2021.

[15] FPT-Software. FPT reinforcement learning competition, 2020. URL https://github.com/xphongvn/rlcomp2020.

[16] Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2018.

[17] Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, and Stefano Ermon. IQ-Learn: Inverse soft-q learning for imitation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4028–4039, 2021.

[18] Geoffrey Grimmett and David Stirzaker. *Probability and Random Processes*. Oxford University Press, 2020.

[19] Nate Gruver, Jiaming Song, Mykel J. Kochenderfer, and Stefano Ermon. Multi-agent adversarial inverse reinforcement learning with latent variables. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1855–1857, 2020.

[20] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning (ICML)*, pages 1352–1361, 2017.

[21] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 4565–4573, 2016.

[22] Hoang Minh Le, Yisong Yue, Peter Carr, and Patrick Lucey. Coordinated multi-agent imitation learning. In *International Conference on Machine Learning (ICML)*, pages 1995–2003, 2017.

[23] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

[24] Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 157–163, 1994.

[25] Chen Liu, Mathieu Salzmann, Tao Lin, Ryota Tomioka, and Sabine Süsstrunk. On the loss landscape of adversarial training: Identifying challenges and how to overcome them. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[26] Jiarong Liu, Yifan Zhong, Siyi Hu, Haobo Fu, Qiang Fu, Xiaojun Chang, and Yaodong Yang. Maximum entropy heterogeneous-agent reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2024.

[27] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8024–8035, 2019.

[29] Dean Pomerleau. Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, 3(1):88–97, 1991.

[30] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-Baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

[31] Tabish Rashid, Mikayel Samvelyan, Christian Schröder de Witt, Gregory Farquhar, Jakob N. Foerster, and Shimon Whiteson. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 4292–4301, 2018.

[32] Christian P Robert, George Casella, and George Casella. *Monte Carlo statistical methods*, volume 2. Springer, 1999.

[33] Stéphane Ross, Geoffrey J. Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 627–635, 2011.

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

[35] Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Hostallero, and Yung Yi. QTRAN: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 5887–5896, 2019.

[36] Jiaming Song, Hongyu Ren, Dorsa Sadigh, and Stefano Ermon. Multi-agent generative adversarial imitation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 7472–7483, 2018.

[37] Yang Song and Diederik P. Kingma. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*, 2021.

[38] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinícius Flores Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 2085–2087, 2018.

[39] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction (Second Edition)*. MIT Press, 2018.

[40] J Terry, Benjamin Black, Nathaniel Grammel, Mario Jayakumar, Ananth Hari, Ryan Sullivan, Luis S Santos, Clemens Dieffendahl, Caroline Horsch, Rodrigo Perez-Vicente, et al. Pettingzoo: Gym for multi-agent reinforcement learning. *Advances in Neural Information Processing Systems (NeurIPS)*, pages 15032–15043, 2021.

[41] MTCAJ Thomas and A Thomas Joy. *Elements of information theory*. Wiley-Interscience, 2006.

[42] Hongwei Wang, Lantao Yu, Zhangjie Cao, and Stefano Ermon. Multi-agent imitation learning with copulas. In *Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*, pages 139–156, 2021.

[43] Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. QPLEX: Duplex dueling multi-agent q-learning. In *International Conference on Learning Representations (ICLR)*, 2021.

[44] Xiaojie Wang, Zhaolong Ning, Song Guo, Miaowen Wen, Lei Guo, and H Vincent Poor. Dynamic uav deployment for differentiated services: A multi-agent imitation learning based approach. *IEEE Transactions on Mobile Computing*, 22(4):2131–2146, 2021.

[45] Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *International Conference on Machine Learning (ICML)*, pages 681–688, 2011.

[46] Minkai Xu, Tomas Geffner, Karsten Kreis, Weili Nie, Yilun Xu, Jure Leskovec, Stefano Ermon, and Arash Vahdat. Energy-based diffusion language models for text generation. *arXiv preprint arXiv:2410.21357*, 2024.

[47] Tian Xu, Ziniu Li, and Yang Yu. Error bounds of imitating policies and environments. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[48] Lantao Yu, Jiaming Song, and Stefano Ermon. Multi-agent adversarial inverse reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 7194–7201, 2019.

[49] Lei Yuan, Ziqian Zhang, Lihe Li, Cong Guan, and Yang Yu. A survey of progress on cooperative multi-agent reinforcement learning in open environment. *arXiv preprint arXiv:2312.01058*, 2023.

[50] Eric Zhan, Stephan Zheng, Yisong Yue, Long Sha, and Patrick Lucey. Generating multi-agent trajectories using programmatic weak supervision. In *International Conference on Learning Representations (ICLR)*, 2019.

[51] Zhi-Hua Zhou. *Machine Learning*. Springer Nature, 2021.

[52] Brian D. Ziebart, Andrew L. Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 1433–1438, 2008.

[53] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 1433–1438, 2008.

[54] Difan Zou, Pan Xu, and Quanquan Gu. Faster convergence of stochastic gradient langevin dynamics for non-log-concave sampling. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 1152–1162, 2021.

A Omitted Proofs

In this section, we will provide the omitted proofs of the propositions mentioned in the main text. To facilitate checking the propositions' content while reading the proofs, we will restate them before providing the corresponding proofs.

A.1 Proof of Proposition 4.1

Proposition 4.1. *For a fixed joint Q-function $Q^{tot}(\boldsymbol{\tau}, \boldsymbol{a})$ and $\phi(x) = x$, the joint policy $\boldsymbol{\pi}_{Q^{tot}}(\boldsymbol{a}|\boldsymbol{\tau})$ that minimizes $\mathcal{J}(\boldsymbol{\pi}, Q^{tot})$ satisfies*

$$\boldsymbol{\pi}_{Q^{tot}}(\boldsymbol{a}|\boldsymbol{\tau}) = \prod_{i=1}^{n} \pi_{Q^i}(a^i|\tau^i), \forall \boldsymbol{\tau} \in \boldsymbol{\mathcal{T}}, \boldsymbol{a} \in \boldsymbol{\mathcal{A}}, \tag{5}$$

where

$$\pi_{Q^i}(a^i|\tau^i) = \frac{1}{Z_{\tau^i}} \exp\left\{\frac{1}{\alpha} k^i(s) Q^i(\tau^i, a^i)\right\} \tag{6}$$

with $Z_{\tau^i} = \sum_{a \in \mathcal{A}^i} \exp\left\{\frac{1}{\alpha} k^i(s) Q^i(\tau^i, a)\right\}$. Thus, we have

$$\max_{Q^{tot}} \min_{\boldsymbol{\pi}} \mathcal{J}(\boldsymbol{\pi}, Q^{tot}) = \max_{Q^{tot}} \mathcal{J}(\boldsymbol{\pi}_{Q^{tot}}, Q^{tot}).$$

Moreover, $\mathcal{J}(\boldsymbol{\pi}_{Q^{tot}}, Q^{tot})$ can be further reduced as

$$\mathcal{J}(\boldsymbol{\pi}_{Q^{tot}}, Q^{tot}) = \sum_{i=1}^{n} -\alpha(1-\gamma)\mathbb{E}_{s_0 \sim \rho_0}\left[\log Z_{\tau_0^i}\right] + \mathbb{E}_{(\boldsymbol{\tau}, \boldsymbol{a}, \boldsymbol{\tau}') \sim \mathcal{D}_E}\left[k^i(s) Q^i(\tau^i, a^i) - \gamma\alpha \log Z_{\tau^{i,\prime}}\right]. \tag{7}$$

Proof. For a fixed joint Q-function Q^{tot} and $\phi(x) = x$, the joint policy's objective is to

$$\min_{\boldsymbol{\pi}} \mathbb{E}_{(\boldsymbol{\tau}, \boldsymbol{a}, \boldsymbol{\tau}') \sim \mathcal{D}_E}\left[Q^{tot}(\boldsymbol{\tau}, \boldsymbol{a}) - \gamma V^{\boldsymbol{\pi}}(\boldsymbol{\tau}')\right] - (1-\gamma)\mathbb{E}_{s_0 \sim \rho_0}[V^{\boldsymbol{\pi}}(\boldsymbol{\tau}_0)].$$

Thus, it is sufficient to maximize $V^{\boldsymbol{\pi}}(\boldsymbol{\tau}), \forall \boldsymbol{\tau} \in \mathcal{T}^n$, i.e.,

$$\begin{aligned}
\boldsymbol{\pi}_{Q^{tot}}(\cdot|\boldsymbol{\tau}) \in & \arg\max_{\boldsymbol{\pi}} V^{\boldsymbol{\pi}}(\boldsymbol{\tau}) \\
:= & \mathbb{E}_{\boldsymbol{a} \sim \boldsymbol{\pi}(\cdot|\boldsymbol{\tau})}\left[Q^{tot}(\boldsymbol{\tau}, \boldsymbol{a}) - \alpha \log \boldsymbol{\pi}(\boldsymbol{a}|\boldsymbol{\tau})\right] \\
= & -\alpha D_{\mathrm{KL}}\left(\boldsymbol{\pi}(\cdot|\boldsymbol{\tau}) \| \frac{1}{Z_{\boldsymbol{\tau}}} \exp\left\{\frac{1}{\alpha} Q^{tot}(\boldsymbol{\tau}, \cdot)\right\}\right) + \alpha \log(Z_{\boldsymbol{\tau}}),
\end{aligned}$$

where $Z_{\boldsymbol{\tau}} = \sum_{\boldsymbol{a} \in \boldsymbol{\mathcal{A}}} \exp\left\{\frac{1}{\alpha} Q^{tot}(\boldsymbol{\tau}, \boldsymbol{a})\right\}$ and D_{KL} denotes the Kullback-Leibler divergence [41]. As $\log(Z_{\boldsymbol{\tau}})$ is independent of $\boldsymbol{\pi}$ and $D_{\mathrm{KL}}\left(\boldsymbol{\pi}(\cdot|\boldsymbol{\tau}) \| \frac{1}{Z_{\boldsymbol{\tau}}} \exp\left\{\frac{1}{\alpha} Q^{tot}(\boldsymbol{\tau}, \cdot)\right\}\right)$ achieves its minimum 0 only when $\boldsymbol{\pi}(\boldsymbol{a}|\boldsymbol{\tau}) = \frac{1}{Z_{\boldsymbol{\tau}}} \exp\left\{\frac{1}{\alpha} Q^{tot}(\boldsymbol{\tau}, \boldsymbol{a})\right\}, \forall \boldsymbol{a} \in \boldsymbol{\mathcal{A}}$ and, we obtain

$$\boldsymbol{\pi}_{Q^{tot}}(\boldsymbol{a}|\boldsymbol{\tau}) = \frac{1}{Z_{\boldsymbol{\tau}}} \exp\left\{\frac{1}{\alpha} Q^{tot}(\boldsymbol{\tau}, \boldsymbol{a})\right\}, \forall \boldsymbol{\tau} \in \boldsymbol{\mathcal{T}}, \boldsymbol{a} \in \boldsymbol{\mathcal{A}}. \tag{10}$$

By taking Equation (4) into Equation (10), we get

$$\begin{aligned}
\boldsymbol{\pi}_{Q^{tot}}(\boldsymbol{a}|\boldsymbol{\tau}) = & \frac{\exp\left\{\frac{1}{\alpha}\left(\sum_{i=1}^{n} k^i(s) Q^i(\tau^i, a^i)\right)\right\}}{\sum_{\tilde{\boldsymbol{a}} \in \boldsymbol{\mathcal{A}}} \exp\left\{\frac{1}{\alpha}\left(\sum_{i=1}^{n} k^i(s) Q^i(\tau^i, \tilde{a}^i)\right)\right\}} \\
= & \frac{\prod_{i=1}^{n} \exp\left\{\frac{1}{\alpha} k^i(s) Q^i(\tau^i, a^i)\right\}}{\prod_{i=1}^{n} \sum_{\tilde{a}^i \in \mathcal{A}^i} \exp\left\{\frac{1}{\alpha} k^i(s) Q^i(\tau^i, \tilde{a}^i)\right\}} \\
= & \prod_{i=1}^{n} \frac{\exp\left\{\frac{1}{\alpha} k^i(s) Q^i(\tau^i, a^i)\right\}}{\sum_{\tilde{a}^i \in \mathcal{A}^i} \exp\left\{\frac{1}{\alpha} k^i(s) Q^i(\tau^i, \tilde{a}^i)\right\}} \\
:= & \prod_{i=1}^{n} \pi_{Q^i}(a^i|\tau^i),
\end{aligned}$$

which verifies Equation (5) and Equation (6). Therefore, $\mathcal{J}(\boldsymbol{\pi}_{Q^{tot}}, Q^{tot})$ can be reduced as

$$
\begin{aligned}
\mathcal{J}(\boldsymbol{\pi}_{Q^{tot}}, Q^{tot}) =& \mathbb{E}_{(\boldsymbol{\tau},\boldsymbol{a},\boldsymbol{\tau}')\sim\mathcal{D}_E}\left[Q^{tot}(\boldsymbol{\tau},\boldsymbol{a}) - \gamma V^{\boldsymbol{\pi}_{Q^{tot}}}(\boldsymbol{\tau}')\right] - (1-\gamma)\mathbb{E}_{s_0\sim\rho_0}[V^{\boldsymbol{\pi}_{Q^{tot}}}(\boldsymbol{\tau}_0)] \\
=& \mathbb{E}_{(\boldsymbol{\tau},\boldsymbol{a},\boldsymbol{\tau}')\sim\mathcal{D}_E}\left[Q^{tot}(\boldsymbol{\tau},\boldsymbol{a}) - \gamma\alpha\log Z_{\boldsymbol{\tau}'}\right] - \alpha(1-\gamma)\mathbb{E}_{s_0\sim\rho_0}[\log Z_{\boldsymbol{\tau}_0}] \\
=& \mathbb{E}_{(\boldsymbol{\tau},\boldsymbol{a},\boldsymbol{\tau}')\sim\mathcal{D}_E}\left[\sum_{i=1}^{n} k^i(s)Q^i(\tau^i,a^i) - \gamma\alpha\log\sum_{\boldsymbol{a}'\in\boldsymbol{\mathcal{A}}}\exp\left\{\frac{1}{\alpha}\sum_{i=1}^{n}k^i(s')Q^i(\tau^{i,\prime},a^{i,\prime})\right\}\right] \\
& - \alpha(1-\gamma)\mathbb{E}_{s_0\sim\rho_0}\left[\log\sum_{\boldsymbol{a}_0\in\boldsymbol{\mathcal{A}}}\exp\left\{\frac{1}{\alpha}\sum_{i=1}^{n}k^i(s_0)Q^i(\tau_0^i,a_0^i)\right\}\right] \\
=& \mathbb{E}_{(\boldsymbol{\tau},\boldsymbol{a},\boldsymbol{\tau}')\sim\mathcal{D}_E}\left[\sum_{i=1}^{n} k^i(s)Q^i(\tau^i,a^i) - \gamma\alpha\log\prod_{i=1}^{n}\sum_{a^{i,\prime}\in\mathcal{A}^i}\exp\left\{\frac{1}{\alpha}k^i(s')Q^i(\tau^{i,\prime},a^{i,\prime})\right\}\right] \\
& - \alpha(1-\gamma)\mathbb{E}_{s_0\sim\rho_0}\left[\log\prod_{i=1}^{n}\sum_{a_0^i\in\mathcal{A}^i}\exp\left\{\frac{1}{\alpha}k^i(s_0)Q^i(\tau_0^i,a_0^i)\right\}\right] \\
=& \mathbb{E}_{(\boldsymbol{\tau},\boldsymbol{a},\boldsymbol{\tau}')\sim\mathcal{D}_E}\left[\sum_{i=1}^{n} k^i(s)Q^i(\tau^i,a^i) - \gamma\alpha\sum_{i=1}^{n}\log\sum_{a^{i,\prime}\in\mathcal{A}^i}\exp\left\{\frac{1}{\alpha}k^i(s')Q^i(\tau^{i,\prime},a^{i,\prime})\right\}\right] \\
& - \alpha(1-\gamma)\mathbb{E}_{s_0\sim\rho_0}\left[\sum_{i=1}^{n}\log\sum_{a_0^i\in\mathcal{A}^i}\exp\left\{\frac{1}{\alpha}k^i(s_0)Q^i(\tau_0^i,a_0^i)\right\}\right] \\
=& \sum_{i=1}^{n}\mathbb{E}_{(\boldsymbol{\tau},\boldsymbol{a},\boldsymbol{\tau}')\sim\mathcal{D}_E}\left[k^i(s)Q^i(\tau^i,a^i) - \gamma\alpha\log\sum_{a^{i,\prime}\in\mathcal{A}^i}\exp\left\{\frac{1}{\alpha}k^i(s')Q^i(\tau^{i,\prime},a^{i,\prime})\right\}\right] \\
& - \alpha(1-\gamma)\mathbb{E}_{s_0\sim\rho_0}\left[\log\sum_{a_0^i\in\mathcal{A}^i}\exp\left\{\frac{1}{\alpha}k^i(s_0)Q^i(\tau_0^i,a_0^i)\right\}\right] \\
:=& \sum_{i=1}^{n}\mathbb{E}_{(\boldsymbol{\tau},\boldsymbol{a},\boldsymbol{\tau}')\sim\mathcal{D}_E}\left[k^i(s)Q^i(\tau^i,a^i) - \gamma\alpha\log Z_{\tau^{i,\prime}}\right] - \alpha(1-\gamma)\mathbb{E}_{s_0\sim\rho_0}\left[\log Z_{\tau_0^i}\right],
\end{aligned}
$$

with $Z_\tau = \sum_{a\in\mathcal{A}}\exp\{\frac{1}{\alpha}k^i(s)Q^i(\tau,a)\}$, which concludes the proof. □

A.2 Proof of Proposition 4.2

Proposition 4.2. *The gradient of $\mathcal{J}(\boldsymbol{\pi}_{Q^{tot}}, Q^{tot})$ with respect to θ is equal to*

$$
\begin{aligned}
\nabla_\theta\mathcal{J}\left(\boldsymbol{\pi}_{Q^{tot}}, Q^{tot}\right) =& \sum_{i=1}^{n} -(1-\gamma)\mathbb{E}_{s_0\sim\rho_0, a_0^i\sim\pi_{Q^i}(\cdot|\tau_0^i)}\left[\nabla_\theta\left(k^i(s)Q^i(\tau_0^i,a_0^i)\right)\right] \\
& + \mathbb{E}_{(\boldsymbol{\tau},\boldsymbol{a},\boldsymbol{\tau}')\sim\mathcal{D}_E}\left[\nabla_\theta\left(k^i(s)Q^i(\tau^i,a^i)\right) - \gamma\mathbb{E}_{a^{i,\prime}\sim\pi_{Q^i}(\cdot|\tau^{i,\prime})}\left[\nabla_\theta\left(k^i(s')Q^i(\tau^{i,\prime},a^{i,\prime})\right)\right]\right].
\end{aligned}
\tag{8}
$$

Proof. All we need to prove is

$$
\alpha\nabla_\theta\log Z_{\tau^i} = \mathbb{E}_{a\sim\pi_{Q^i}}\left[\nabla_\theta\left(k^i(s)Q^i(\tau,a)\right)\right], \forall\tau\in\mathcal{T}^i, a\in\mathcal{A}^i, i\in\mathcal{I}.
$$

By definition, we have that

$$
\begin{aligned}
\alpha\nabla_\theta\log Z_{\tau^i} &= \alpha\frac{\nabla_\theta\sum_{a\in\mathcal{A}^i}\exp\{\frac{1}{\alpha}k^i(s)Q^i(\tau,a)\}}{Z_\tau} \\
&= \alpha\sum_{a\in\mathcal{A}^i}\underbrace{\frac{\exp\{\frac{1}{\alpha}k^i(s)Q^i(\tau,a)\}}{Z_\tau}}_{\pi_{Q^i}}\nabla_\theta\left(\frac{1}{\alpha}k^i(s)Q^i(\tau,a)\right) \\
&:= \mathbb{E}_{a\sim\pi_{Q^i}}\left[\nabla_\theta\left(k^i(s)Q^i(\tau,a)\right)\right], \forall\tau\in\mathcal{T}^i, a\in\mathcal{A}^i, i\in\mathcal{I},
\end{aligned}
$$

which concludes the proof. □

B Distinction with MIFQ

Closely related to our work, Bui et al. [7] also considers extending IQ-Learn to multi-agent tasks by incorporating value decomposition network. However, their work differs significantly from ours. We highlight the key differences to help readers better understand the distinctions between our work and theirs:

- Given a joint Q-function, the corresponding optimal soft value function $V^*(s)$ in Bui et al. [7] is $V^*(s) = \sum_{i=1}^n k^i(s) \log \sum_{a^i \in \mathcal{A}^i} \exp\{(Q^i(\tau^i, a^i)\}$, whereas we derive in Appendix A.1 that $V^*(s) = \sum_{i=1}^n \log \sum_{a^i \in \mathcal{A}^i} \exp\{k^i(s) Q^i(\tau^i, a^i)\}$. Apparently, $\sum_{i=1}^n k^i(s) \log \sum_{a^i \in \mathcal{A}^i} \exp\{(Q^i(\tau^i, a^i)\} \neq \sum_{i=1}^n \log \sum_{a^i \in \mathcal{A}^i} \exp\{k^i(s) Q^i(\tau^i, a^i)\}$.
- MIFQ does not consider continuous control tasks. As shown in Figure 1 of Bui et al. [7], computing the soft value function requires to get $\log \sum_{a^i \in \mathcal{A}^i} \exp\{(Q^i(\tau^i, a^i)\}$, which is intractable for continuous action spaces. We bypass computing the `logsumexp` term by observing that the gradient of our objective can be estimated by sampling actions from the energy-based policy.

C Implementation

In this section, we will present omitted details on the implementation of MAFIS.

C.1 Network Architecture

We adopt the same network architecture of the joint soft Q-function as QMIX [31], which is shown in Figure 5. Agents $1, 2, \cdots, n$ represent the individual Q-function. To make sure that $k^i(s_t) > 0, \forall i \in \mathcal{I}$, the output layer of the mixing network will be passed through the sigmoid function.

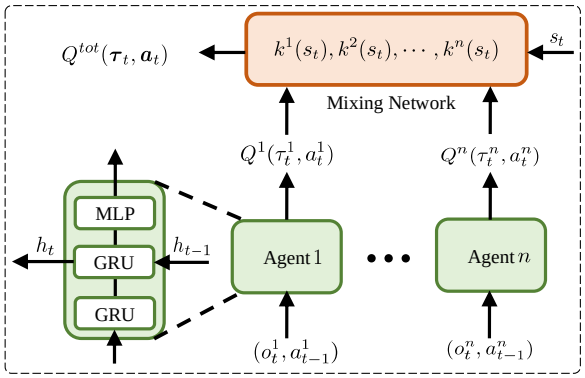


Figure 5: Network architecture of the joint soft Q-function. MLP ann GRU represent multi-layer perceptron [51] and gated recurrent unit [9], respectively. The mixing network is also an MLP.

C.2 Hardware and Software

We use the following software versions:

- Python 3.7
- Gym 0.21.0 [6]
- MuJoCo-py 2.1.2.14
- PyTorch 1.12.1 [28]

We use the following hardware:

- NVIDIA RTX 4090 x 8
- 12th Gen Intel(R) Core(TM) i9-12900K

C.3 Hyper-Parameter Settings

The hyper-parameter settings used for benchmarks results are presented in Table 2 and Table 3.

Table 2: Hyper-parameter settings for discrete control tasks.

Hyper-parameter	Value
batch size	32
α	0.5 for `zerg_{10_vs_10, 5_vs_5}` and `protoss_5_vs_5` 0.2 for others
(online) update frequency	5 for MPE and SMACv2 2 for Gold Miner

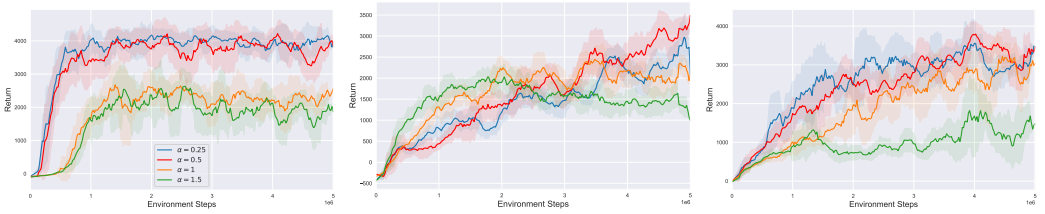
Table 3: Hyper-parameter settings for continuous control tasks.

Hyper-parameter	Value
batch size	1000
Langevin steps K	25
Langevin nose variance σ^2	0.25
Sample number N	20
Entropy weight α	0.5
(online) update frequency	5

C.4 Technical Details of MAFIS

We implement our method upon the pyMARL code library [34]. The design of the joint soft Q-function's network (which we can Q-network) is inspired by the network architecture of QMIX [31]. Additionally, for discrete control tasks, we introduce dropout with a rate of 0.5 in the mixing network to mitigate the risk of over-fitting. For continuous control tasks, we incorporate a target Q-network, which is updated using the Polyak average update mechanism [30] with an update ratio of 0.005. To ensure stable training, we use the target Q-network to sample actions to estimate Equation (8). Furthermore, we apply a gradient penalty to the Q-network with a coefficient of 0.25 and a gradient margin of 1. We also found that constraining the output of the Q-network can further improve performance. Therefore, we apply L2 regularization to its output with a coefficient of 0.01. Garg et al. [17] prove that $(1-\gamma)\mathbb{E}_{s_0\sim\rho_0}[V^\pi(s_0)] = \mathbb{E}_\mu[V^\pi(s) - \gamma V^\pi(s')]$ for any feasible policy μ. Practically, the authors utilize the expert demonstration dataset as $\mathcal{D}_\mu$ for offline imitation learning, while employing a balanced mixture ($1:1$ ratio) of expert demonstrations and policy-generated rollouts for online imitation[5]. We follow their configurations for online and offline learning.

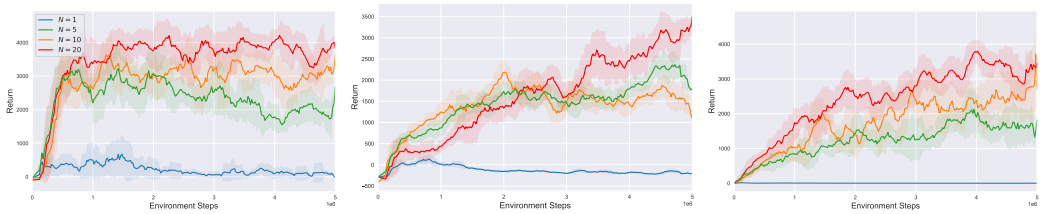
D More Sensitivity Analysis Results



(a) `Ant(2x4)` (b) `HalfCheetah(2x3)` (c) `Walker2d(2x3)`

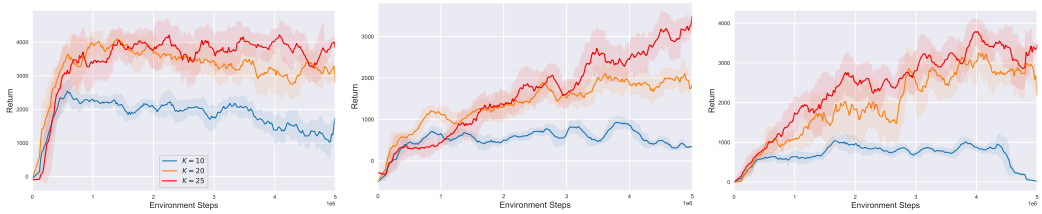
Figure 6: Sensitivity of MAFIS to the entropy weight α. Excessively large values of alpha can cause the model to over-prioritize exploration, thereby compromising its convergence performance.

[5] https://github.com/Div99/IQ-Learn



(a) `Ant(2x4)` (b) `HalfCheetah(2x3)` (c) `Walker2d(2x3)`

Figure 7: Sentivity of MAFIS to the number of parallel samples N. As N increases, more samples are obtained, leading to more accurate gradient estimation in Equation (8) during training and higher probability of locating the global maximum during evaluation. However, this comes at the cost of increased computational overhead.



(a) `Ant(2x4)` (b) `HalfCheetah(2x3)` (c) `Walker2d(2x3)`

Figure 8: Sentivity of MAFIS to the number of Langevin dynamics steps K. Careful enlargement of K promotes convergence in Langevin dynamics.

(a) `Ant(2x4)` (b) `HalfCheetah(2x3)` (c) `Walker2d(2x3)`

Figure 9: Sentivity of MAFIS to the standard deviation σ of the noise. A properly sized σ is required. When σ is too small, sampling may become trapped in local optima; conversely, an excessively large σ may cause unstable deviation from the optimal solution.

(a) `Ant(2x4)` (b) `HalfCheetah(2x3)` (c) `Walker2d(2x3)`

Figure 10: Sentivity of MAFIS to the Langevin dynamics step size ϵ. When ϵ is too small, performing $K = 25$ steps of Langevin dynamics may remain far from the convergence point; whereas an excessively large ϵ leads to instability in the later sampling stages.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We clearly present our contributions and scope in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discussion the limitation of our work in Section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We present the detailed proof to each proposition in Appendix A.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We present the implementation details in both Section 5 and Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We upload the code and data in supplementary material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All needed information are specified in Section 5 and Appendix C

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Main results shown in Figure 2 and Table 1 report error bars.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide information on computer resources in Appendix C.2

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We did follow the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: This work proposes MAFIS, a new multi-agent imitation learning algorithm. MAFIS has potential societal benefits in areas such as autonomous systems and robotics, contributing to safer and more reliable multi-agent systems. Ethically, this work aligns with standard machine learning advancements, but care must be taken to avoid misuse or emergent adversarial behaviors in sensitive applications. Researchers and practitioners should ensure rigorous testing, transparency, and ethical compliance when deploying MAFIS in real-world scenarios.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the original paper that produced the used code package.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.